# EMC® SnapView™ Command Line Interfaces (CLI)

## REFERENCE

**P/N 069001181**
**REV A15**

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

Published July, 2008

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.

# Contents

## Chapter 2 SnapView Clone CLI Commands



## Chapter 3 Admsnap Commands for Clones

## Chapter 4 SnapView Snapshot CLI Commands



## Chapter 5 Admsnap Commands for Snapshots

# Preface

*As part of an effort to improve and enhance the performance and capabilities of its product line, EMC from time to time releases revisions of its hardware and software. Therefore, some functions described in this guide may not be supported by all revisions 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 manual, please contact your EMC representative.*

## About this manual

This manual describes the two command line interfaces to EMC® SnapView™ software: the admsnap utility and the CLI (Command Line Interface). You should read this manual if you will use typed or scripted commands (instead of or in addition to EMC Navisphere® Manager software) to configure and run the SnapView software. This manual assumes that you are familiar with the operating system running on the servers you will manage and with storage-system components and configurations.

**Note:** This document uses the term "CX4 series" to refer to CX4-120, CX4-240, CX4-480, and CX4-960 storage systems, the term "CX3 series" to refer to CX3 model 10 systems, CX3 model 20 systems, CX3 model 40 systems, and CX3 model 80 systems, and the term "CX series" to refer to CX300-series, CX500-series, and CX700 storage systems.
This document also uses the term "AX4-5 series" to refer to AX4-5SC, AX4-5SCi, AX4-5, AX4-5i storage systems and the term "AX series" to refer to AX150 series and AX100 series storage systems. AX150 series systems include the AX150SC, AX150SCi, AX150, and AX150i. AX100 series systems include the AX100SC, AX100SCi, AX100, and AX100i.

## Audience

This guide is intended for use by customers and service providers who use admsnap and the EMC Navisphere Command Line Interface (CLI).

Readers of this guide are expected to be familiar with EMC Navisphere CLI, which is introduced in the *EMC Navisphere Command Line Interface (CLI) Reference*.

**Note:** If you have an IRIX or Tru64 server, refer to revision A11 of this guide.

### Storage systems no longer covered in this document

The table below lists the storage systems that are not longer covered in this document and the last revision of this document that included the storage systems.

| Storage system removed | Last revision including the storage system |
|---|---|
| CX200, CX400, and CX600 | Revision 14 |
| FC4700 series | Revision 08 |

## Organization

This manual is organized as follows:

Chapter 1 Introduces the EMC SnapView software application, including clones and snapshots. This chapter also lists the configuration guidelines and describes the SnapView terminology differences between Navisphere Express and Navisphere Manager.

Chapter 2 Describes the SnapView clone CLI commands, which you use to manage clones.

Chapter 3 Explains using admsnap with clones and describes the commands available.

Chapter 4 Describes the SnapView snapshot CLI commands, which you use to manage snapshots.

Chapter 5 Explains using admsnap with snapshots and describes the commands available.

Chapter 6 Contains examples, from setting up clones and snapshots (with Navisphere CLI) and snapshots of clones to using them (with admsnap and Navisphere CLI).

| | |
|---|---|
| Appendix A | Describes numeric errors that can return from CLI commands to SnapView software. |
| Appendix B | Describes what bad blocks are, how SnapView handles them, and what you can do to correct them. |
| Glossary | Defines SnapView and other terms used in this guide. |

## Related documentation

Related documents include:

- *EMC CLARiiON Server Support Products for AIX Installation Guide* (P/N 300-002-044)
- *EMC CLARiiON Server Support Products for HP-UX Installation Guide* (P/N 300-002-043)
- *EMC CLARiiON Server Support Products for Linux Server Installation Guide* (P/N 300-002-041)
- *EMC CLARiiON Server Support Products for VMware ESX Server Installation Guide* (P/N 300-004-281)
- *EMC CLARiiON Server Support Products for NetWare Installation Guide* (P/N 300-002-040)
- *EMC CLARiiON Server Support Products for Solaris Installation Guide* (P/N 300-002-039)
- *EMC CLARiiON Server Support Products for Windows Installation Guide* (P/N 300-002-038)
- For the most current management and security content for CX4 series, CX3 series, or CX series and supported AX4-5 series or AX series storage systems, refer to EMC Navisphere Manager online help. For FC-series and C-series management and security content, refer to the most recent versions of *EMC Navisphere Manager Administrator's Guide* (P/N 069001125) and *EMC Navisphere Security Domains, Multi-Domains and User Accounts Administrator's Guide* (P/N 069001124).
- For the most current information on CX4 series, CX3 series, or CX series and AX4-5 series or AX series storage systems, refer to the latest version of the *EMC Navisphere Command Line Interface (CLI) Reference* (P/N 300-003-628). For information on FC-series and C-series storage systems, refer to the *EMC Navisphere Command Line Interface (CLI) Reference*, P/N 069001038 Rev A14.

## Conventions used in this guide

EMC uses the following conventions for notes and cautions.

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

**CAUTION**

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

### Typographical conventions

This manual uses the following format conventions:

| | |
|---|---|
| **this typeface** | Text (including punctuation) that you type verbatim, all commands, pathnames, and filenames, and directory names. It indicates the name of a dialog box, field in a dialog box, menu, menu option, or button. |
| `this typeface` | Represents a system response (such as a message or prompt), a file or program listing. |
| *this typeface* | Represents variables for which you supply the values; for example, the name of a directory or file, your username or password, and explicit arguments to commands. |
| **x > y** | Represents a menu path. For example, **Operations > Poll All Storage Systems** tells you to select **Poll All Storage Systems** on the **Operations** menu. |
| `[ ]` | Encloses optional entries. |
| \| | Separates alternative parameter values; for example:<br>*LUN-name* \| *LUN-number* means you can use either the LUN-name or the LUN-number. |

### Finding current information

The most up-to-date information about the SnapView software is posted on the EMC Powerlink® website. We recommend that you download the latest information before you start the SnapView software.

To access EMC Powerlink, use the following link:

`http://Powerlink.EMC.com`

After you log in, select **Support > Documentation/White Paper Library** and find the following:

- Release notes for SnapView and admsnap
- The latest version of this manual that is applicable to your software revision. For information on FC4700-series storage systems, refer to revision A08 of this guide.
- *EMC Installation Roadmap for CLARiiON Storage Systems*, which provides a checklist of the tasks that you must complete to install your storage system in a storage area network (SAN) or direct attach configuration.

### Where to get help

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

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

`http://Powerlink.EMC.com`

**Technical support —** For technical support, go to EMC WebSupport on Powerlink. To open a case on EMC WebSupport, you must be a WebSupport customer. Information about your site configuration and the circumstances under which the problem occurred is required.

### 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 guide to:

`techpub_comments@EMC.com`

# 1

# About EMC SnapView Software

This chapter introduces the EMC® SnapView™ software and its user interface, as well as the two command line interfaces for it. The command line interfaces include the server-based admsnap utility and the EMC Navisphere® CLI interface.

**Note:** CX4 series storage systems do not support Windows 2000 servers as an attached host.

Major topics are:

# Introduction to SnapView

SnapView is a storage-system-based software application that allows you to create a copy of a LUN by using either clones or snapshots.

A clone is an actual copy of a LUN and takes time to create, depending on the size of the source LUN. A snapshot is a virtual point-in-time copy of a LUN which tracks differences to your original data, and takes only seconds to create.

SnapView has the following important benefits:

- It allows full access to a point-in-time copy of your production data with modest impact on performance and without modifying the actual production data.
- For decision support or revision testing, it provides a coherent, readable and writable copy of real production data.
- For backup, it practically eliminates the time that production data spends offline or in hot backup mode. And it offloads the backup overhead from the production server to another server.
- It provides a consistent replica across a set of LUNs. You can do this by performing a consistent fracture, which is a fracture of more than one clone at the same time, or a consistent session that you create when starting a session in consistent mode.
- It provides instantaneous data recovery if the source LUN becomes corrupt. You can perform a recovery operation on a clone by initiating a reverse synchronization and on a snapshot session by initiating a rollback operation.

Depending on your application needs, you can create clones, snapshots, or snapshots of clones. For a detailed overview on clones, refer to the "Clones overview" (see page 15). For a detailed overview on snapshots, refer to the "Snapshots overview" (see page 16). For a comparison of using clones, snapshots, and snapshots of clones, refer to Table 1, "A comparison of clones and snapshots," on page 18.

## Clones overview

A clone is a complete copy of a source LUN. You specify a source LUN when you create a clone group. The copy of the source LUN begins when you add a clone LUN to the clone group. The software assigns each clone a clone ID. This ID remains with the clone until you remove the clone from its group.

While the clone is part of the clone group and unfractured, any production write requests made to the source LUN are simultaneously copied to the clone. Once the clone contains the desired data, you can fracture the clone. Fracturing the clone separates it from its source LUN, after which you can make it available to a secondary server.

Clone private LUNs track areas on the source and clone that have changed since the clone was fractured. A log records information that identifies data chunks on the source LUN and clone LUN that have been modified after you fractured the clone. A modified (changed) data chunk is a chunk of data that a production or secondary server changes by writing to the source LUN or clone. A log in the clone private LUN records this information, but no actual data is written to the clone private LUN. This log reduces the time it takes to synchronize or reverse synchronize a clone and its source LUN since the software copies only modified chunks.

Figure 1 shows an example of how a fractured clone works. Note, as the production server writes to the source LUN, and the secondary server writes to the clone, the clone private LUN tracks areas on the source and clone that have changed since the clone was fractured.

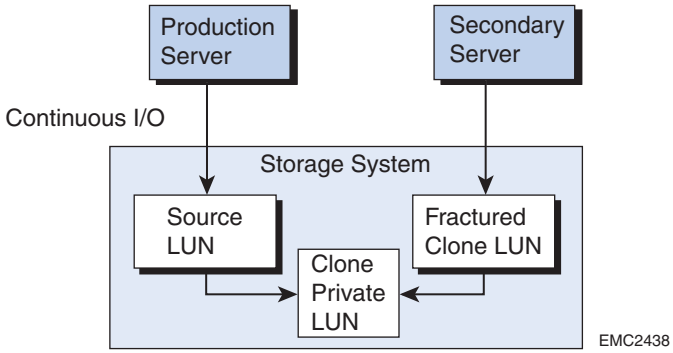


**Figure 1** **Clones example**

To create a consistent replica across a set of LUNs, you can perform a consistent fracture, which fractures one clone at a time per source LUN across multiple source LUNs.

SnapView also allows you to instantly restore a clone back to the source LUN, if the source LUN were to become corrupt or if a clone's data is desired for the source. You can do this by initiating a reverse synchronization on the clone.

## Snapshots overview

A snapshot is a virtual LUN that allows a secondary server to view a point-in-time copy of a source LUN. You determine the point in time when you start a SnapView session. The session keeps track of the source LUN's data at a particular point in time. Though a snapshot appears as a conventional LUN to other servers, its data does not reside on a disk like a conventional LUN. A snapshot is a composite of the unchanged data chunks on the source LUN and data chunks on the reserved LUN. The data chunks on the source LUN are those that have not been modified since you started the session. The data chunks in the reserved LUN pool are copies of the original source LUN data chunks that have been modified since you started the session.

During a session, the production server can still write to the source LUN and modify data. When this happens, the software stores a copy of the original point-in-time data on a reserved LUN in the reserved LUN pool. This operation is referred to as copy-on-first-write because it occurs only when a data chunk is first modified on the source LUN.

As the session continues and additional I/O modifies other data chunks on the source LUN, the amount of data stored in the reserved LUN pool grows. If needed, you can increase the size of the reserved LUN pool by adding more LUNs to the LUN pool.

**Note:** An adequate number of reserved LUNs is essential since SnapView terminates sessions if the reserved LUN runs out of space and no additional LUNs are in the reserved LUN pool.

From a secondary server, you can view a session's point-in-time data by activating (mapping) a snapshot to the session. You can activate only one snapshot at a time to a session. If another point-in-time view is desired, you can deactivate (unmap) a snapshot from a session and activate it to another session of the same source LUN or you can create another snapshot and activate it to the second session.

Figure 2 shows an example of how snapshots work. The snapshot is a composite of the source LUN and the reserved LUN in the global reserved LUN pool.

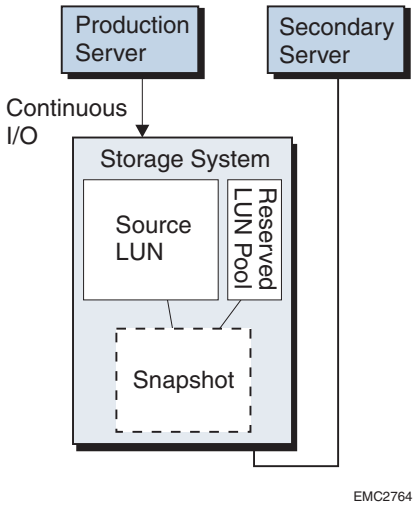


**Figure 2 Snapshot example**

To create a consistent replica across a set of LUNs, you can start a session in consistent mode.

SnapView also allows you to instantly restore a session's point-in-time data back to the source LUN, if the source LUN were to

become corrupt or if a session's point-in-time data is desired for the source. You can do this by using SnapView's rollback feature.

## Clone and snapshot comparison

The following table compares clones, snapshots, and snapshots of clones.

**Note:** The source LUN you specify when creating a clone group is available for server I/O during any clone state except for a reverse out-of-sync state. Any server writes made to the source LUN during a reverse synchronization are copied to the clone.

**Table 1 A comparison of clones and snapshots**

| Clones | Snapshots | Snapshots of clones |
|---|---|---|
| **Benefits** | | |
| • Provides immediacy in replacing the contents of the source LUN with the contents of the clone LUN and redirecting servers from the source to the clone, should the source become corrupted.<br>• Makes backup operation nondisruptive.<br>• Provides enhanced protection against critical data loss because it is an actual LUN. | • Provides immediacy in replacing the contents of the source LUN with the contents of the session, should the source LUN become corrupted.<br>• Makes backup operation nondisruptive.<br>• Provides a quick and instant copy because it is a virtual LUN. | • Provides immediacy in replacing the contents of the source LUN with the contents of the session, should the source LUN become corrupted.<br>• Makes backup operation nondisruptive.<br>• Provides an extra level of protection against critical data loss if both the source LUN and clone LUN become corrupted. |
| **Creation time** | | |
| Minutes to hours. The creation time depends on the size of the source LUN. Subsequent synchronizations are incremental. | Instantaneous | Instantaneous |
| **Disk space used** | | |
| Uses the same amount of disk space as the source LUN. | Uses reserved LUN pool space, which is usually 10% to 20% of the source LUN size per session, but will vary depending on how much data has changed on the source LUN. | Uses reserved LUN pool space (for the snapshot) and full disk space (for the clone), which usually totals 100% of the source LUN size for clones and 10% to 20% of the source LUN size per session, but will vary depending on how much data has changed on the source LUN. |

Table 1 A comparison of clones and snapshots (continued)

| Clones | Snapshots | Snapshots of clones |
|---|---|---|
| **Data recovery time after source LUN failure/corruption** | | |
| Instantaneous after initializing a reverse synchronization. | Instantaneous after initializing a rollback operation. | Combination of rollback from session and reverse-synchronization of clone. |
| **Performance impact on the storage system** | | |
| • There is no performance impact when a clone LUN is in a fractured state.<br>• For the initial synchronization of the clone LUN, there is a performance impact for the duration of the synchronization. Subsequent synchronizations or reverse synchronizations have comparable impact but the duration of the synchronization will be shorter since it is incremental. Impact is also determined by the synchronization rate, which is set when the clone LUN is added to the clone group and can be changed during a synchronization or reverse synchronization. | A performance decrease due to the copy-on-first-write. | Combination of both clone LUNs and snapshot LUNs. |

## SnapView components

SnapView consists of the following software components:

- A set of drivers that provides the SnapView functionality, and resides on the storage system with the LUNs you want to copy.

  **Note:** CX4 series, CX3 series, or CX series storage systems ship from the factory with SnapView software installed, but *not* enabled. To use the SnapView software functionality, the SnapView enabler must be installed on the storage system.

- The admsnap utility that provides a command line executable to let you manage clone and snapshot devices on the server. The admsnap utility ships with the SnapView enabler and resides on any servers connected to storage systems that have the SnapView software installed and enabled.

You must use SnapView with one of the following Navisphere components:

- Navisphere Manager user interface (UI), which must be installed on at least one storage system on the same network as the SnapView storage system.

  **Note:** CX4 series, CX3 series, CX300 series, CX500 series, and CX700 storage systems ship from the factory with Navisphere Manager installed and enabled.

- Navisphere Command Line Interface (CLI), which ships as part of the Navisphere Host Agent packages.

You must use Navisphere Manager or Navisphere CLI (not admsnap) to set up SnapView; then you can use admsnap and Navisphere Manager and/or Navisphere CLI to manage ongoing SnapView operations.

## About Navisphere Manager

Navisphere Manager is a centralized storage-system management tool for configuring and managing CLARiiON® storage systems. It provides the following basic functionality:

- Discovery of CLARiiON storage systems
- Storage configuration and allocation
- Status and configuration information display
- Event management

Navisphere Manager is a web-based user interface that lets you securely manage CLARiiON storage systems locally on the same LAN or remotely over the Internet, using a common browser. Navisphere Manager resides on a CX4 series, CX3 series, or CX series storage system or a Windows Server 2008, Windows Server 2003, or Windows 2000 server that is running the Storage Management Server software, and is downloaded to the browser when the Storage Management Server software is accessed.

For additional information on how to use SnapView with Navisphere Manager, refer to the EMC Navisphere Manager online help.

## About Navisphere Express

Navisphere Express software is a storage-system-based management tool used to manage AX4-5 series or AX series storage systems. It consists of the storage-system integrated management software and a web-based user interface (UI). Both the management software and the UI are installed at the factory on the storage controllers in the storage system. The easy-to-use Navisphere Express UI is displayed in a common browser and provides the following:

- Security
- Storage configuration and allocation
- Datan redundancy
- Status and configuration information display
- Event notification
- Data replication
- Data movement
- Data migration (AX4-5 series storage systems only)

For more information on these features, refer to the Navisphere Express online help.

## About Navisphere CLI

The Navisphere CLI provides another management interface (along with Navisphere Manager and admsnap) to clones and snapshots. You can use Navisphere CLI commands and admsnap commands together to manage clones and snapshots. You need both admsnap and Navisphere CLI because admsnap interacts with the server operating system and CLI interacts with the storage system.

## About Admsnap Utility

The admsnap utility is an executable program that you can run interactively or with a script to manage clones and snapshots. The admsnap utility resides on the servers connected to the storage system with the SnapView driver.

The admsnap utility runs on the following server platforms:

- Hewlett Packard HP-UX
- IBM AIX (RS/6000 and RS/6000 SP servers)
- Linux (32-bit Intel platform, 64-bit AMD processor Linux, 64-bit Intel Xeon processor, and 64-bit Intel Itanium processor)

  **Note:** Separate admsnap installation packages are available for the 32-bit Intel platform, 64-bit AMD processor Linux/64-bit Intel Xeon processor, and the 64-bit Intel Itanium processor. The admsnap packages for the 64-bit AMD processor Linux and the 64-bit Intel Xeon processor are the same. For minimum supported Linux kernel revisions for each platform, refer to the Admsnap Release Notes.

- Microsoft Windows (Windows Server 2008, Windows Server 2003, or Windows 2000)

  **Note:** Separate admsnap installation packages are available for Windows Server 2008, Windows Server 2003, and Windows 2000. Unless indicated otherwise, all references to a Windows server in this document refer to Windows Server 2003, and Windows 2000 servers *only*.

- Novell NetWare
- Sun Solaris
- VMware ESX Server

For the supported versions of these servers/operating systems, refer to the most up-to-date release notes for SnapView and admsnap.

## Using SnapView with MirrorView

You can use SnapView with MirrorView software to create a clone or a snapshot of a remote mirror LUN (a LUN participating as either a primary or secondary image), which allows for protection at both the local and remote sites should either of these become corrupt. However, clones offer added disk protection and have less of a performance impact than snapshots.

If you have a database corruption at the local site, you can use the point-in-time clone or snapshot of the primary image to recover the data. To do this, you would use the instant restore capability of clones (reverse synchronization) or snapshots (rollback) so that the primary image can instantly restore the non-corrupted data.

At the remote site, creating a clone or a snapshot of a secondary image allows you to replace the mirror with the clone or snapshot in the event that a data corruption occurred across the mirrors.

Since clones offer less of a performance impact than snapshots, you could create a clone of a secondary mirror before resynchronizing the mirror to the primary. This allows you greater protection with minimal performance impact should anything occur during the resynchronization, such as a data corruption, or the primary fails during the resynchronization, leaving the secondary in an unusable state. You can also create a clone of a mirror and back up the clone, which again offers less of a performance impact than backing up snapshots.

We recommend that you take a clone or a snapshot of a remote image only if the image's state is either synchronized or consistent (meaning that the last update that was started has completed successfully). If the image is synchronizing or out-of-sync, the clone or snapshot data will not be useful because the image was fractured or the update is still in progress.

**Note:** You must have SnapView version 6.24 or later to create a clone of a remote mirror LUN (a LUN participating as either a primary or secondary image).

If the clone's source is a MirrorView secondary image, the MirrorView driver prevents host I/O to the secondary image and reverse synchronization of an unfractured secondary image.

## Using SnapView with SAN Copy

You can use SnapView with SAN Copy software to create a clone of the destinationLUN. You can add the clone to the secondary server storage group rather than the SAN Copy destination. This allows the SAN Copy destination to maintain consistency with its source, and be available on an ongoing basis for incremental updates.

**Note:** SAN Copy tracks server writes to the SAN Copy source LUN (from the production server). SAN Copy does not track server writes to the SAN Copy destination LUN (from the secondary server), so you must fracture all clones of a SAN Copy destination LUN before you start the SAN Copy session. You can then synchronize the clones once the SAN Copy session completes.

## SnapView servers

The procedures in this document use two servers: one server (called the production server) contains the LUN you want to copy, and another server (called the secondary server) lets you view the clone or snapshot.

**IMPORTANT**

**If you have a Windows server, or if you are managing a system with Navisphere Express, you must have at least two servers to use the SnapView.**

The production server:

- Runs the customer applications
- Owns the source LUN

The secondary server (or any other server):

- Owns the clone or snapshot
- Reads from and writes to the fractured clone or activated snapshot
- Performs secondary tasks using the clone or snapshot or an independent analysis (such as, backup, decision support, or revision testing)

**Note:** You can configure a clustered server to access a source LUN, but not both the source LUN and its clone or snapshot. Only a server outside the cluster can access the clone or snapshot.

Figure 3 shows a sample SnapView environment with two servers creating clones and snapshots of two database files and their logs.

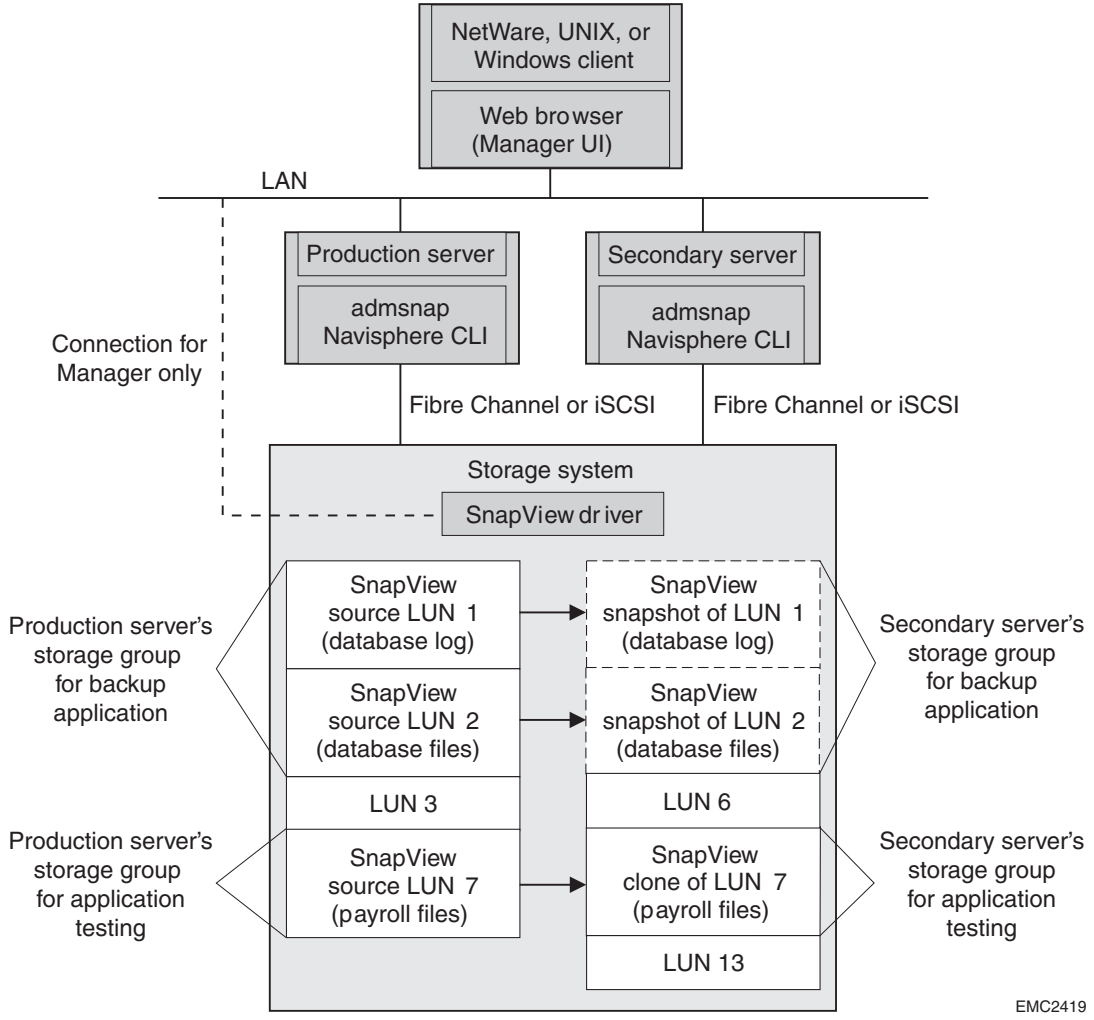


**Figure 3 Sample SnapView environment with clones and snapshots**

# Using SnapView with AX4-5 Series or AX Series storage systems

This section describes SnapView terminology differences between Navisphere Express and Navisphere Manager and the SnapView CLI commands supported for AX4-5 or AX series storage systems.

## Terminology differences between Navisphere Express and Navisphere Manager

As with Navisphere Express, Navisphere Manager on AX4-5 and AX series storage systems allows users to manage SnapView. Since Manager's user interface must include functionality for all storage system types it supports, some SnapView commands and features are limited or unavailable for AX4-5 and AX series storage systems (see Table 2 on page 28).

The following table lists and defines the SnapView terminology differences between Navisphere Express and Navisphere Manager. For terminology differences that are *not* specific to SnapView, refer to the EMC Navisphere Manager online help.

| Navisphere Express term | Navisphere Manager term | Navisphere Manager definition |
|---|---|---|
| snapshot | SnapView session and snapshot | A SnapView session is a point-in-time copy of a source LUN. The session keeps track of how the source LUN looks at a particular point in time.<br>A snapshot is a virtual LUN and when activated, it allows a secondary server to view a SnapView session.<br>You can create a snapshot before or after you start a SnapView session; however, the snapshot has no use until a secondary server activates it to a session. |
| disk resources | reserved LUN pool | The reserved LUN pool works with replication software, such as SnapView, SAN Copy, and MirrorView, to store data or information required to complete a replication task. For example, with SnapView, after you start a SnapView session and as the production server writes to the source LUN, the software stores a copy of the original data in chunks in the reserved LUN pool. When a secondary server activates the snapshot to the SnapView session, the snapshot views the original source LUN data chunks that have been modified since you started the session from the reserved LUN pool and unmodified data chunks from the source LUN. |

## SnapView CLI commands supported for AX4-5 Series or AX Series storage systems

Depending on your AX4-5 series or AX series storage system and management software, some SnapView commands and switches are limited or unavailable. Table 2 lists the commands that are supported for AX4-5 series and AX series storage systems.

**Note:** Navisphere CLI is not supported on AX series storage systems (AX150 and AX100 series).

**Table 2 SnapView CLI commands supported for AX4-5 and AX series systems**

| Command | Supported on | | | |
|---|---|---|---|---|
| | AX4-5 series with Navisphere Express | AX4-5 series with Navisphere Manager | | AX150 and AX100 series with Navisphere Manager enabled |
| | | Without the SnapView enabler installed | With the SnapView enabler installed | |
| **Snapshots** | | | | |
| -activatesnapshot | Yes | Yes | Yes | Yes |
| -chgrollback | No | No | Yes | No |
| -chgsnapshot | No | Yes | Yes | Yes |
| -createsnapshot | No | Yes | Yes | Yes |
| -deactivatesnapshot | Yes | Yes | Yes | Yes |
| -destroysnapshot | Yes | No | No | No |
| -listrollback | No | No | Yes | No |
| -listsnapshots | Yes | Yes | Yes | Yes |
| -listsessions | Yes[a] | Yes[a] | Yes[a] | Yes[a] |
| -listsnapableluns | Yes | Yes | Yes | Yes |
| -lunpool | Yes | Yes | Yes | Yes |
| -preparesnapshot | Yes | No | No | No |
| -rmsnapshot | No | Yes | Yes | Yes |
| -startrollback | No | No | Yes | No |
| -startsession | Yes[a] | Yes[a] | Yes | Yes[a] |
| -stopsession | Yes | Yes | Yes | Yes |

| Command | Supported on | | | |
|---|---|---|---|---|
| | AX4-5 series with Navisphere Express | AX4-5 series with Navisphere Manager | | AX150 and AX100 series with Navisphere Manager enabled |
| | | Without the SnapView enabler installed | With the SnapView enabler installed | |
| **Clones** | | | | |
| -addclone | No | No | Yes | No |
| -allocatecpl | No | No | Yes | No |
| -changeclone | No | No | Yes | No |
| -changeclonegroup | No | No | Yes | No |
| -changeclonefeature | No | No | Yes | No |
| -consistentfractureclones | No | No | Yes | No |
| -createclonegroup | No | No | Yes | No |
| -deallocatecpl | No | No | Yes | No |
| -destroyclonegroup | No | No | Yes | No |
| -fractureclone | No | No | Yes | No |
| -listclone | No | No | Yes | No |
| -listclonefeature | No | No | Yes | No |
| -listclonegroup | No | No | Yes | No |
| -removeclone | No | No | Yes | No |
| -reversesyncclone | No | No | Yes | No |
| -syncclone | No | No | Yes | No |

a. Some switches are not supported on this command.

## SnapView limits

This section lists the maximum SnapView limits for your storage system. For AX4-5 series or AX series storage systems refer to Table 3 on page 31, for CX4 series storage systems refer to Table 4 on page 32, for CX3 series storage systems refer to Table 5 on page 33, and for CX series storage systems refer to Table 6 on page 34.

**Note:** A metaLUN is a single entity, and therefore it counts as one of your clone or snapshot limits. For example, if a metaLUN is composed of five LUNs and you create a clone of this LUN, it counts as one of your clone limits, not five. SnapView also supports the new, larger LUNs that FLARE® software supports (refer to the FLARE release notes).

Table 3 Maximum SnapView limits for AX4-5 series or AX series storage systems

| Parameter | AX4-5 series with Navisphere Express | AX4-5 series with Navisphere Manager | | AX150 series with Navisphere Manager[a] | AX100 series with Navisphere Manager[a] |
|---|---|---|---|---|---|
| | | Without the SnapView enabler installed | With the SnapView enabler installed | | |
| **SnapView clones** | | | | | |
| **Clones** | | | | | |
| Per storage system | Not supported | Not supported | 128 | Not supported | Not supported |
| Per source LUN | | | 8 | | |
| **Clone groups** | | | | | |
| Per storage system | Not supported | Not supported | 64 | Not supported | Not supported |
| **Clone private LUNs** | | | | | |
| Per storage system (required) | Not supported | Not supported | 2 | Not supported | Not supported |
| **Source LUNs** | | | | | |
| Per storage system | Not supported | Not supported | 64 | Not supported | Not supported |
| **SnapView snapshots** | | | | | |
| **Snapshots[b]** | | | | | |
| Per storage system | 16 | 16 | 256 | 8 | 4 |
| Per source LUN | 1 | 1 | 8 | 1 | 1 |
| **SnapView sessions[a]** | | | | | |
| Per source LUN | 1 | 1 | 8 | 1 | 1 |
| **Reserved LUNs** | | | | | |
| Per storage system | 64 | 64 | 64 | 40 | 20 |
| **Source LUNs** | | | | | |
| Per storage system | 16 | 16 | 64 | 40 | 20 |

a. For limits with Navisphere Express, refer to the Navisphere Express online help.

b. The limits for snapshots and sessions include SnapView snapshots or SnapView sessions as well as reserved snapshots or reserved sessions used in other applications, such as SAN Copy (incremental sessions) and MirrorView/Asynchronous applications.

Table 4 Maximum SnapView limits for CX4 series storage systems

| Parameter | CX4-120 | CX4-240 | CX4-480 | CX4-960 |
|---|---|---|---|---|
| **SnapView clones** | | | | |
| **Clones** | | | | |
| Per storage system | 256[a] | 512[a] | 1024[a] | 2048[a] |
| Per source LUN | 8 | 8 | 8 | 8 |
| **Clone groups** | | | | |
| Per storage system | 128 | 256 | 512 | 1024 |
| **Clone private LUNs** | | | | |
| Per storage system (required) | 2 | 2 | 2 | 2 |
| **Source LUNs** | | | | |
| Per storage system | 128 | 256 | 512 | 1024 |
| **SnapView snapshots** | | | | |
| **Snapshots**[b] | | | | |
| Per storage system | 512 | 512 | 1024 | 2048 |
| Per source LUN | 8 | 8 | 8 | 8 |
| **SnapView sessions**[b] | | | | |
| Per source LUN | 8 | 8 | 8 | 8 |
| **Reserved LUNs** | | | | |
| Per storage system | 128 | 128 | 256 | 512 |
| **Source LUNs** | | | | |
| Per storage system | 128 | 128 | 256 | 512 |

a. The source LUN and MirrorView/Synchronous primary and secondary images no longer count towards the clone image limit.

b. The limits for snapshots and sessions include SnapView snapshots or SnapView sessions as well as reserved snapshots or reserved sessions used in other applications, such as SAN Copy (incremental sessions) and MirrorView/Asynchronous applications.

Table 5 Maximum SnapView limits for CX3 series storage systems

| Parameter | CX3 series model 10 | CX3 series model 20 | CX3 series model 40 | CX3 series model 80 |
|---|---|---|---|---|
| **SnapView clones** | | | | |
| **Clones** | | | | |
| Per storage system | 128 | 256[a] | 512[a] | 1024[a] |
| Per source LUN | 8 | 8 | 8 | 8 |
| **Clone groups** | | | | |
| Per storage system | 64 | 128 | 256 | 512 |
| **Clone private LUNs** | | | | |
| Per storage system (required) | 2 | 2 | 2 | 2 |
| **Source LUNs** | | | | |
| Per storage system | 64 | 128 | 256 | 512 |
| **SnapView snapshots** | | | | |
| **Snapshots**[b] | | | | |
| Per storage system | 256 | 512 | 1024 | 2048 |
| Per source LUN | 8 | 8 | 8 | 8 |
| **SnapView sessions**[b] | | | | |
| Per source LUN | 8 | 8 | 8 | 8 |
| **Reserved LUNs** | | | | |
| Per storage system | 64 | 128 | 256 | 512 |
| **Source LUNs** | | | | |
| Per storage system | 64 | 128 | 256 | 512 |

a. The source LUN and MirrorView/Synchronous primary and secondary images no longer count towards the clone image limit.

b. The limits for snapshots and sessions include SnapView snapshots or SnapView sessions as well as reserved snapshots or reserved sessions used in other applications, such as SAN Copy (incremental sessions) and MirrorView/Asynchronous applications.

Table 6 Maximum SnapView limits for CX series storage systems.

| Parameter | CX300-series | CX500-series | CX700 |
|---|---|---|---|
| **SnapView clones** | | | |
| **Clones** | | | |
| Per storage system | 100[a] | 100[a] | 200[a] |
| Per source LUN | 8 | 8 | 8 |
| **Clone groups** | | | |
| Per storage system | 50 | 50 | 100 |
| **Clone private LUNs** | | | |
| Per storage system (required) | 2 | 2 | 2 |
| **Source LUNs** | | | |
| Per storage system | 50 | 50 | 100 |
| **SnapView snapshots** | | | |
| **Snapshots[b]** | | | |
| Per storage system | 100 | 150 | 300 |
| Per source LUN | 8 | 8 | 8 |
| **SnapView sessions[c]** | | | |
| Per source LUN | 8 | 8 | 8 |
| **Reserved LUNs** | | | |
| Per storage system | 25 | 50 | 100 |
| **Source LUNs** | | | |
| Per storage system | 25 | 50 | 100 |

a. The source LUN and MirrorView/Synchronous primary and secondary images no longer count towards the clone image limit.

b. The limits for snapshots and sessions include SnapView snapshots or SnapView sessions as well as reserved snapshots or reserved sessions used in other applications, such as SAN Copy (incremental sessions) and MirrorView/Asynchronous applications.

c. The limit includes MirrorView/Synchronous primary and secondary images. The source is no longer counted towards the image limit.

**What next?** Depending on the type of replica you will create, go to one of the following chapters:

**Setting up and using clones**

- Chapter 2, "SnapView Clone CLI Commands."
- Chapter 3, "Admsnap Commands for Clones."

**Setting up and using snapshots**

- Chapter 4, "SnapView Snapshot CLI Commands."
- Chapter 5, "Admsnap Commands for Snapshots."

# 2

# SnapView Clone CLI Commands

This chapter explains the Navisphere SnapView clone CLI commands that manage SnapView clones. The clone feature is only supported on CX4 series, CX3 series, CX series, and AX4-5 series with Navisphere Manager and the SnapView enabler installed.

**Note:** If you currently use Java CLI to issue CLI commands, please note that Secure CLI is replacing Java CLI; therefore, only Secure CLI will support new features. We recommend that you use Secure CLI. Secure CLI is also faster and more secure than Java CLI. Please refer to the *EMC SnapView Command Line Interfaces (CLI)* (P/N 069001181 Revision A09) for Java CLI support.

Major sections in the chapter are:

# About Secure CLI

Secure CLI is a comprehensive Navisphere CLI solution that provides one application and one security model for all CLI commands. Secure CLI provides role-based authentication, audit trails of CLI events, and SSL-based data encryption. You do not need to install a JRE to run Secure CLI.

**Note:** Refer to the Host Agent/CLI and Utilities Release Notes, available on Powerlink, for a list of supported operating systems. You must be running FLARE Operating Environment version 02.19.xxx.5.yyy or higher.

Secure CLI commands run in a command window. Each command consists of the **naviseccli** command (and switches) together with another subcommand (and its switches).

**Note:** For commands that originated in Classic CLI, some command output may be enhanced; for example, Secure CLI can retrieve and display information from peer SPs. For Classic CLI commands that produce multiple warnings and require multiple confirmations, Secure CLI provides a single summary of warnings and a single confirmation. In general, Secure CLI preserves original command syntax and output for script compatibility.

Secure CLI does not support commands targeted to host agents, such as **-lunmapinfo**. For commands that you can issue to either an SP or host agent, such as **–getagent**, if you issue the command to a host agent, Secure CLI displays an error message. You must use Classic CLI to issue commands to host agents. For a list of commands that are not supported, refer to the *EMC Navisphere Command Line Interface (CLI) Reference*.

**Note:** Secure CLI does not distinguish case of characters, so, regardless of the host operating system, you can use either uppercase, lowercase, or any combination of characters as you type commands.

If a Secure CLI command fails and the CLI does not generate its own error message, it displays an error message from the SP agent. Secure CLI generates errors about command line syntax for commands and options and their values.

Secure CLI commands return 0 if the command is successful, or greater than zero if the command is unsuccessful.

# Getting started with Secure CLI

Before you begin to issue Secure CLI commands, you must create a Navisphere user account on the storage system. To create the required user accounts using Navisphere CLI, refer to the *EMC Navisphere Command Line Interface (CLI) Reference*. For details on using Navisphere 6.X security, refer to the Navisphere Manager online help.

You can also choose to configure a Navisphere 6.X security file to issue Secure CLI commands (see the next section, "Overview of using Navisphere 6.X security with Secure CLI") on the host. If you establish a security file, you do not need to include the switches **-user**, **-scope,** and -**password** (or the password prompt), in each command you issue.

**Note:** Establishing a security file to use commands that originated in Classic CLI ensures that other than the update from navicli to the naviseccli command, you do not need to modify any established scripts you may have. Secure CLI, unlike Classic CLI, requires the switches **-user**, **-scope**, and **-password** (or the password prompt) in each command line; you do not need to provide these switches in the command line if you establish a security file.

## Overview of using Navisphere 6.X security with Secure CLI

A storage system will not accept a command from Secure CLI unless the user who issues the command has a valid user account on the storage system. You can specify a valid account username, password, and scope (global, local, or lightweight directory access protocol (LDAP)) for each command you issue, or, more conveniently, you can create a Navisphere security file.

The Navisphere security file is an encrypted file stored for each user on each host. You can add or delete a user security file using the **-AddUserSecurity** or **-RemoveUserSecurity** functions as arguments to the **naviseccli** command. You cannot copy a security file to another host. You must issue the **-AddUserSecurity** function on the host for which you want to create the security file.

When you create a security file, the username you use to log in to the current host is automatically stored in the security file, or you can specify an alternative username for the security file in the **-AddUserSecurity** request using the optional **-user** switch. If you omit the -**user** switch, the security file uses your current username.

For example, to add yourself to the security file on the current host, given the alternative username **altusername**, the password **mypass** and the scope **0** (global scope), type:

**naviseccli -address SP-IP -AddUserSecurity -password mypass -scope 0 -user altusername**

Then, on this host, you can enter CLI commands to any storage system on which you have an account that matches the username **altusername**, with password **mypass** and global scope (**scope 0**).

---

**Note:** Username and password are case sensitive.

---

The security file is stored in your default home directory. With Secure CLI, you can specify an alternative file path using the optional **-secfilepath** switch.

---

**Note:** If you specify an alternative location for the security file, you must specify the file path in every subsequent CLI command you issue to ensure the CLI locates the security file.

---

To save the example used above to the alternative location "c:\altlocation\" type:

**naviseccli -AddUserSecurity -password mypass -scope 0 -user altusername - secfilepath c:\altlocation**

Then, for each subsequent command you issue, you must specify the **-secfilepath** switch with the security file path location "c:\altlocation\" in the command line.

# naviseccli

**Sends status or configuration requests to a storage system via command line**

## Description

The **naviseccli** command sends storage-system management and configuration requests to a storage system via the Internet.

## Syntax

The **naviseccli** command is used as follows:

**naviseccli -help**
or
**naviseccli**

[**-address** *IPaddress* | *NetworkName* | **-h** *IPaddress* | *NetworkName*]
[**-AddUserSecurity**]
[**-f** *filename]*
[**-m**]
[-**nopoll** | **-np**]
[-**parse** | **-p**]
[-**password** *password*]
[-**port** *port*]
[**-q**]
[-**RemoveUserSecurity**]
[-**scope** 0 | 1 | 2]
[-**timeout** | **-t** *timeout*]
[-**user** *username*]
[**-v**]
[**-xml**]
*CMD* [*optional_command_switches*]

## Prerequisites

To use the **naviseccli** command, you must be running Navisphere CLI 6.X or later.

## Options

-**help**

Displays the help screen and does not start the **naviseccli** process. To start the **naviseccli** process, use one or more of the switches that follow instead.

**-address** *IPaddress* | *NetworkName* | **-h** *IPaddress* | *NetworkName*

Specifies the IP address or network name of the targeted SP on the desired storage system. The default, if you omit this switch, is **localhost**.

**-AddUserSecurity**

Directs the CLI to add user security information to the security file on this server. You must use the **-scope** switch to add scope information to the security file. You can use the -**password** switch or enter your password into the password prompt (see -**password**), to supply the required password information to the security file. The **-user** and **-secfilepath** switches are optional with this command.

---

**Note:** If you specify the -**user** switch, you can create an alternative username to your server login name in the security file you create on this server. If you use the -**secfilepath** switch, you can specify an alternative location to your default home directory for the security file on this server. You must then use the -**secfilepath** switch in each subsequent command you issue.

---

**-f** *filename*

Specifies that the data be stored in a file.

**-m**

Suppresses output except for values. This option is most useful when used as part of a script.

---

**Note:** Only supported for commands that originated in Classic CLI.

---

**-nopoll | -np**

Directs the feature provider not to issue a poll request. This switch significantly increases performance when dealing with large or multiple storage systems. The feature provider automatically polls unless this switch is specified.

---

**Note:** When the **-nopoll** switch is set, **get** commands may return stale data and **set** commands may erase previously changed settings. Use caution when the **-nopoll** switch is set.

---

**-parse | -p**

Directs the CLI to validate the command. The CLI verifies the command syntax and displays a message stating whether the command was valid. The CLI takes no other action.

**-password** *password*

Specifies the password on the storage system you want to log in to. The password is visible in the command line. Passwords are case sensitive.

If you want to mask the password, and you are not using a security file, you can omit this switch from the command line. The CLI then prompts you to enter a password. The information you enter into the password prompt is concealed.

---

**Note:** You can omit this switch if you are using a security file. See -**AddUserSecurity**.

---

**-port** *portnumber*

Sets the port number (type) of the storage system. The default is 443. If you choose to change the default port number, management port 2163 will be supported; however, you will need to specify the **-port** switch and number 2163 in every subsequent command you issue.

**-q**

Suppresses error messages. This switch is useful when included as part of a script.

---

**Note:** Only supported for commands that originated in Classic CLI.

---

**-RemoveUserSecurity**

Directs the CLI to remove user security information about the current user from the security file on this server.

**-scope 0 | 1 | 2**

Specifies whether the user account on the storage system you want to log in to is local, global, or lightweight directory access protocol (LDAP). **0** (default) indicates global, **1** indicates local, and 2 indicates LDAP.

A global account is effective throughout the domain. When the administrator creates a global account, the software copies the definition of this account to the domain directory, which makes it accessible on all storage systems in the domain.

A local account is effective on only the storage systems for which the administrator creates the account. The user can log in to only those storage systems on which he has a local account.

LDAP maps the username/password entries to an external LDAP or active directory server for authentication.
Username/password pairs whose roles are not mapped to the external directory will be denied access.

**Note:** For more information on LDAP refer to the *EMC Navisphere Command Line Interface (CLI) Reference* or the Navisphere Manager online help.

**-secfilepath** *filepath*

Stores the security file in a file path location you specify. When you create a security file on a server using the **-addusersecurity** command, the security file is saved to your default home directory. If you want to store the security file in an alternative location, you can use the optional **-secfilepath** switch with the **-addusersecurity** command.

**Note:** If you use the **-secfilepath** switch to set up an alternative path for your security file, you must use this switch in every subsequent CLI command you issue, to ensure the CLI locates the security file.

**-timeout** | **-t** *timeout*

Sets the timeout value in seconds. The default is 600 seconds.

**-user** *username*

Specifies the username on the storage system you want to log in to. Usernames are case sensitive. You can omit this if your username has been added to the security file.

**Note:** You can use this switch when establishing a security file, to specify an alternative username. See -**AddUserSecurity.**

**-v**

Enables verbose error descriptions. This is the default unless **-q** is specified.

**Note:** Only supported for commands that originated in Classic CLI.

**-xml**

Specifies command output in XML format. Use the **-o** (override switch) when specifying **-xml** on commands that require confirmation. Otherwise, the XML output will contain your confirmation string.

*CMD*

One of a set of commands used with the **naviseccli** command.

### CMD switches

The *CMD* switches are described on the pages that follow.

### Spaces in arguments

Normally, each argument to a CLI command consists of numbers, letters, and other valid printable characters for the operating system on which the CLI is running.

If a filename or other argument includes a space, you must enclose the entire string that includes the name in quotation marks ("xx xx").

# Setting up SnapView to use clones

This section describes how to set up SnapView to use clones.

**Note:** You cannot expand the capacity of a LUN or metaLUN that is participating in a clone group until you remove the clone from the clone group and destroy the clone group. Neither the production nor secondary server can access this added capacity until the expansion is complete and you perform some additional operations. For detailed information on expanding a LUN or metaLUN, see the *EMC Navisphere Command Line Interface (CLI) Reference*.

## Prerequisites for setting up clones

Before you can set up and use clones, the following must be true:

- **Source LUNs must be bound**. For a client or production server to access a source LUN, you must assign the source LUN to a storage group and connect the storage group to the production server. To do this, you must enable data access control on the storage system.
- **LUNs that you plan to use as clone LUNs must be bound**. These LUNs must be the same size as the source LUNs that you will clone. EMC strongly recommends that you bind your clone LUNs on different RAID groups than their source LUNs. The clone's RAID group does *not* have to be the same RAID type as the source LUN.
- **For VMware ESX Servers, verify that the source LUN is presented to the virtual machine** (guest operating system running on the virtual machine). For information on how to present a LUN to the virtual machine, refer to the VMware documentation that shipped with your ESX Server.
- **For configuration restrictions when using VMFS volumes**, go to the E-Lab Interoperability Navigator on EMC Powerlink and under the **PDFs and Guides** tab, open the **VMware ESX server.pdf** file.
- **For a secondary server to access the clone LUN, the clone must be assigned to a storage group** (but you cannot read the clone until you fracture it). The storage group must be connected to the secondary server that will access the clone. You must assign the clone LUN to a storage group other than the storage group that

holds the source LUN. EMC supports placing a clone in the same storage group as its source LUN *only if* you use Replication Manager or Replication Manager/SE to put the clone in the storage group. This software provides s*ame host access* to the clone and the source LUN. For information on using these software products, refer to the documentation for the product.

If you have a VMware ESX Server, the clone and source LUNs must be accessed by different virtual machines, unless the virtual machine is running one of the software programs that support same host access.

## Overview of setting up SnapView to use clones

The following is a checklist for setting up SnapView clones:

- ❑ Allocate clone private LUNs; **clone | snapview -allocatecpl** (page 61).

  **Note:** This step is required only in the initial setup of clones.

- ❑ Create a clone group; **clone | snapview -createclonegroup** (page 72).
- ❑ Add a clone to a clone group; **clone | snapview -addclone** (page 57).

**What next?** To fracture the clone, refer to page 76.

For a complete step-by-step overview, refer to "Step-by-step clone overview - all platforms" on page 198.

## Clone states

Each clone in a clone group has its own state that indicates if it contains usable data. The possible clone states are: consistent, out-of-sync, remote mirror synchronizing, reverse out-of-sync, reverse synchronizing, synchronized, or synchronizing. Depending on the state of the clone, some operations may be unavailable (refer to Table 7 on page 50).

When you remove a clone from the clone group, it is no longer associated with its source LUN or clone group. It retains the copied data and becomes a conventional (regular) LUN.

---

**Note:** Table 7 on page 50 lists when the clone is available for server I/O. The source LUN specified when creating a clone group is available for server I/O during any clone state *except* for a reverse out-of-sync state. If the clone's source is a MirrorView secondary image, the MirrorView driver prevents host I/O to the source.

Any server writes made to the source LUN during a reverse synchronization are copied to the clone.If you do not want incoming source writes copied to the clone during a reverse synchronization, you must issue the **-UseProtectedRestore 1** switch with the **clone | snapview -addclone** command before issuing a reverse synchronization. However, before you can initiate the **Protected Restore** feature, you must globally enable it by issuing the **clone | snapview -changeclonefeature [-AllowProtectedRestore 1]** command.
You *cannot* perform a reverse synchronization if the clone's source is a MirrorView secondary image.

---

Table 7 Clone states

| Clone state | Description | Cause of state | Clone available for I/O | What next? | |
|---|---|---|---|---|---|
| | | | | Permitted operations | New clone state |
| **Consistent** | If the clone is unfractured, its data reflects some point-in-time copy of its source.<br>If the clone is fractured, its data reflects some point-in- time copy of its source, modified by any host writes to the clone. | • A clone is fractured while in a consistent or Synchronized state.<br>• A clone finishes synchronizing or reverse-synchronizing.<br>• An unfractured clone is in the Synchronized state, and its source receives a host write.<br>• A clone is in the remote mirror synchronizing state, and its source (a MirrorView secondary image) transitions to the consistent state. | Yes, if clone is fractured | Fracture (only if clone is not already fractured) | Consistent; Administratively fractured |
| | | | | Remove (only if the clone is fractured) | N/A |
| | | | | Synchronize (only if clone is fractured) | Synchronizing |
| | | | | Reverse synchronize (only if clone is fractured and clone's source is not a MirrorView secondary image) | Reverse synchronizing |
| **Out-of-sync** | A clone is not a point in time copy of its source and is unusable. | • A clone was fractured while synchronizing.<br>• A clone was fractured while in the remote mirror synchronizing state. | No | Synchronize | Synchronizing |
| | | | | Remove | N/A |
| | | | | Fracture (only if the clone was fractured by the system due to an error in the software or storage system; refer to the event log for the cause of the system fracture) | Out-of-sync; Administratively fractured |

Table 7 Clone states (continued)

| Clone state | Description | Cause of state | Clone available for I/O | What next? | |
|---|---|---|---|---|---|
| | | | | Permitted operations | New clone state |
| **Remote mirror synchronizing** | A clone is not a usable point-in-time copy because its source (a MirrorView secondary image) is in the synchronizing or out-of-sync state or may be in the process of a rollback. | An unfractured clone's source is a remote mirror secondary that begins synchronizing with the MirrorView primary image. | No | Fracture | Out-of-sync |
| **Reverse out-of-sync** | A clone's source is unusable and another reverse sync operation is recommended. | A clone was fractured while reverse-synchronizing. | Yes | Reverse synchronize | Reverse synchronizing |
| | | | | Remove | N/A |
| | | | | Fracture (only if the clone was fractured by the system due to an error in the software or storage system; refer to the event log for the cause of the system fracture) | reverse out-of-sync; administratively fractured |

Table 7 Clone states (continued)

| Clone state | Description | Cause of state | Clone available for I/O | What next? | |
|---|---|---|---|---|---|
| | | | | Permitted operations | New clone state |
| **Reverse synchronizing** | A clone is in the process of copying its data to its source LUN. | • Manually administered with Navisphere or the CLI from the production server.<br>• Automatically restarted following an SP failure only if a reverse synchronization was in progress during the SP failure and the recovery policy was set to **Auto**.<br>**Note** Before you can reverse synchronize a clone, you must fracture it. Also, the clone cannot be in an out-of-sync state, and no other clone in the clone group can be synchronizing or reverse synchronizing. | No | Fracture | Reverse out-of-sync |

Table 7 Clone states (continued)

| Clone state | Description | Cause of state | Clone available for I/O | What next? | |
|---|---|---|---|---|---|
| | | | | Permitted operations | New clone state |
| **Synchronized** | A clone is a byte-for-byte copy of its source.<br>**Note** The clone will immediately transition to the consistent or remote mirror synchronizing state if the source receives a write. | • After a synchronization, reverse synchronization, or a server write request, an unfractured clone automatically transitions to this state when the source LUN does not receive any server write requests during a specified period of time, which is defined by the Clone Groups Quiesce Threshold.<br>• A clone is added to a clone group without setting "Initial Sync Required". | No | Fracture | Consistent |
| | | | | Remove | N/A |

Table 7 Clone states (continued)

| Clone state | Description | Cause of state | Clone available for I/O | What next? | |
|---|---|---|---|---|---|
| | | | | Permitted operations | New clone state |
| **Synchronizing** | A clone is in the process of copying data from its source LUN. | • Manually administered through Navisphere by the production server.<br>• Automatically started when you add a clone to a clone group with the **Initial Sync Required** property selected.<br>• Automatically restarted following an SP failure (if a synchronization was in progress during an SP failure and the recovery policy was set to **Auto**).<br>**Note** Before you can manually synchronize a clone, you must fracture the clone, and the clone cannot be in a reverse out-of-sync state. | No | Fracture | Out-of-sync |

## CLI clone command functions

The **naviseccli** clone functions, listed alphabetically, are:

**Note:** As of version 6.26, you can issue clone commands using the **clone** or **snapview** prefix.

- **clone | snapview -addclone**
- **clone | snapview -allocatecpl**
- **clone | snapview -changeclone**
- **clone | snapview -changeclonegroup**
- **clone | snapview -changeclonefeature**
- **clone | snapview -consistentfractureclones**
- **clone | snapview -createclonegroup**
- **clone | snapview -deallocatecpl**
- **clone | snapview -destroyclonegroup**
- **clone | snapview -fractureclone**
- **clone | snapview -listclone**
- **clone | snapview -listclonefeature**
- **clone | snapview -listclonegroup**
- **clone | snapview -removeclone**
- **clone | snapview -reversesyncclone**
- **clone | snapview -syncclone**

The CLI clone functions are listed functionally, in the order you would perform them, in the following table.

Table 8 naviseccli SnapView clone command functions, listed in order of use

| Essential functions (in order performed) | |
|---|---|
| **clone | snapview -allocatecpl** | Designates clone private LUNs. |
| **clone | snapview -createclonegroup** | Creates a clone group. |
| **clone | snapview -addclone** | Adds a clone LUN to the clone group. |
| **clone | snapview -syncclone** | Starts synchronizing the clone with the source LUN (not needed if **-addclone** does not specify that an initial sync is not required). |
| **clone | snapview -consistentfractureclones** or **clone | snapview -fractureclone** | Frees the clone LUN(s) from its source, allowing you to use the clone independently. |
| **clone | snapview -removeclone** | Deletes a clone from a clone group. |
| **clone | snapview -destroyclonegroup** | Deletes a clone group. |
| **Optional and status functions (alphabetically)** | |
| **clone | snapview -listclone** | Displays clone information. |
| **clone | snapview -listclonefeature** | Displays clone features such as whether creating a clone group on this storage system is supported and other useful information. |
| **clone | snapview -listclonegroup** | Displays clone group information. |
| **Optional reconfiguration functions (alphabetically)** | |
| **clone | snapview -changeclone** | Changes one or more properties of a clone. |
| **clone | snapview -changeclonefeature** | For a storage system, enables or disables clone features such as protected restore. |
| **clone | snapview -changeclonegroup** | Changes clone group settings. |
| **clone | snapview -deallocatecpl** | Frees LUNs that were allocated as clone private LUNs from the storage-system clone structure. |
| **clone | snapview -reversesyncclone** | Starts a reverse sync process, in which the software copies the clone data to the source. |

# clone | snapview -addclone

**Adds a clone to a clone group.**

## Description

The **naviseccli clone** or the **naviseccli snapview** command with **-addclone** adds a clone to a clone group. Each clone LUN you add must be the same size as the source LUN. When you add a clone to a clone group, you create an actual copy of the source LUN you specified when creating the group. The copy of the source LUN begins when you specify the **-issyncrequired 1** command when adding the clone LUN.

The software assigns each clone a clone ID. This ID remains with the clone until you remove the clone from its group.

**Note:** When you add a clone to a clone group, SnapView automatically allocates and retains additional memory. This additional memory counts against the total replication software memory budget.

**Source LUN writes and unfractured clone states**
While the clone is part of the clone group and unfractured (not accessible to a secondary server), server writes to the source LUN can continue, with the following results:

**Server writes to the source —** When a source LUN receives a server write request, the clone transitions into a consistent state because the clone is no longer a byte-for-byte copy of its source. Writes made to the source LUN are simultaneously copied to the clone. The clone remains in a consistent state if server writes are made to the source LUN during the duration of the quiesce threshold. The quiesce threshold is the amount of time the clone must wait before transitioning from a consistent state to a synchronized state. If the source LUN receives any server writes during this period of time, the time set for the quiesce threshold restarts. You specify the quiesce threshold when you create a clone group.

**Note:** When you add a clone to the clone group, with the **Initial Sync Required** property selected, the clone state is synchronizing. The software transitions the clone to synchronized or consistent state only after the initial synchronization is complete.

**No server writes to the source —** If a source LUN does not receive any server writes for the duration of the quiesce threshold, any unfractured clone in a consistent state transitions into a synchronized state.

**Note:** For information about the state of a clone, see "Clone states" on page 49.

## Syntax

**clone -addclone** or **snapview -addclone** is used with **naviseccli** (described on page 42) as follows.

**clone | snapview -addclone -name** *name* | **-CloneGroupUid** *uid*
**-luns** *lun_numbers* **[-issyncrequired 0 | 1]**
**[-RecoveryPolicy manual | auto]**
**[-SyncRate high | medium | low | *value*]**
**[-UseProtectedRestore 0 | 1]**

## Prerequisites

You must have a user account on the storage system on which you want the command executed.

## Options

**-name** *name*

Specifies the name that you gave to the clone group with the **-createclonegroup** function. Use either this name or the clone group unique ID (16-byte WWN).

**-CloneGroupUid** *uid*

Specifies the 16-byte unique ID the software assigned when the clone group was created. The clone group ID consists of the numbers 0 through 9 and the letters A through F (these letters are not case sensitive). You can use the **-listclonegroup** command function to obtain the clone group ID (see page 85).

**-luns** *lun_numbers*

Specifies the LUN numbers that can serve as clones. Currently only one LUN is supported per command.

**-issyncrequired 0 | 1**

Specifies whether the software must synchronize the newly added clone.

**0** Synchronization is not required. Using this value lets you skip the synchronization period if the source LUN contains null data (as with a newly bound source LUN).

**CAUTION: When you specify issyncrequired 0, the software marks the clone as synchronized regardless of its contents. Therefore, do not specify issyncrequired 0 if the source LUN contains any data, because the clone will appear as synchronized when it is not synchronized, and will probably not contain usable data.**

**1** Synchronization is required (default). The software starts synchronizing the clone immediately.

**-RecoveryPolicy manual | auto**

Specifies the synchronization action the storage system will take after a hardware component (such as an SP) fails. Values are:

- **manual** - Manual recovery. The administrator must act to start synchronization.
- **auto** - Automatic. Synchronization will start automatically using the other path (if available). This is the default.

The value you select also applies if reverse-synchronization was in progress before the failure.

**-SyncRate high | medium | low | *value***

Sets the interval between writes to a clone being synchronized to **high**, **medium**, or **low**, or a *value* in milliseconds. Default is **medium**. A high rate synchronizes the clone the fastest, but may affect performance on other system resources. A low value has minimal effect on performance, but prolongs the synchronization process.

**Note:** EMC recommends that you do not use a **high** synchronization rate on a storage system with a single SP.

**- UseProtectedRestore 0 | 1**

The way the software reacts when I/O is received during a reverse sync operation.

**0** Updates the clone with the incoming server I/O. This is the default.
**1** Does not update the clone with the I/O.

## Output

None if the command succeeds; status or error information if it fails.

## Examples

**naviseccli -password** *password* **-scope 0 -addusersecurity**

This command adds the current logged in user to the security file. Once you enter this command, you do not have to re-enter your username, password, or scope in the command line.

**naviseccli -address ss1_spa clone -allocatecpl -spA 54 -spB 55**
**naviseccli -address ss1_spa clone -createclonegroup -name db3_clone -luns 22**
**naviseccli -address ss1_spa clone -addclone -name db3_clone -luns 43 -o**
**naviseccli -address ss1_spa clone -listclonegroup**

```
Name:  db3_clone
CloneGroupUid:
50:06:01:60:20:04:21:3E:0C:00:00:00:00:00:00:00
InSync:  No
```

These commands allocate clone private LUNs (LUNs 54 and 55), create a clone group named **db3_clone** for source LUN 22, add a clone LUN (LUN 43) to the group, and then list clone groups. Synchronization starts by default with the **-addclone** command since that command does not specify **-issyncrequired 0.**

To remove the logged in user from the security file, issue the following command:

**naviseccli -removeusersecurity**

# clone | snapview -allocatecpl

**Reserves a clone private LUN for use with clones.**

## Description

The **naviseccli clone** or the **naviseccli snapview** command with **-allocatecpl** allocates or reallocates a clone private LUN on both SPs for clone operations. You must allocate one clone private LUN for each SP before you can create a clone group.

If you use **-allocatecpl** to reallocate (change an existing allocation), you must change the clone private LUN for both SPs. When you reallocate the clone private LUNs, the software transfers the information stored in the previous clone private LUNs to the new clone private LUNs.

**Note:** When you allocate clone private LUNs, SnapView automatically allocates and retains additional memory. This additional memory counts against the total replication software memory budget.

## Syntax

**clone -allocatecpl** or **snapview -allocatecpl** is used with **naviseccli** (described on page 42) as follows:

**clone | snapview -allocatecpl -spA** *lun_numbe*r **-spB** *lun_number* **-o**

## Prerequisites

You must have a user account on the storage system on which you want the command executed.

Each clone private LUN must be a minimum of 1 Gigabyte. Clone private LUNs larger than 1 Gigabyte provide no performance benefit. You must bind these LUNs before you allocate them as a clone private LUN. You can use any LUN that is at least 1 Gigabyte in size as a clone private LUN, *except* for the following:

- Hot spare LUNs
- MirrorView remote mirror LUNs (LUNs used as either a primary or secondary image)
- SnapView clone, snapshot, or source LUNs
- SAN Copy source or destination logical units
- Private LUNs (LUNs reserved as clone private LUNs or for use by the reserved LUN pool)

You should bind clone private LUNs in a RAID group that normally does not see heavy I/O.

**Options**

**-spA** *lun_number*

*lun-number* identifies the LUN to be used for the clone private LUN for SP A. The LUN must be bound. You must specify a clone private LUN for both SP A and SP B. However, the ownership of these LUNs does not matter, since the software will trespass each LUN you specify in each **-sp** switch to the specified SP.

**-spB** *lun_number*

*lun_number* identifies the LUN to be used for the clone private LUN for SP B. The LUN must be bound. See comments under **-spA**.

**-o**

Executes the command without prompting for confirmation.

**Output**

None if the command succeeds; status or error information if it fails.

**Example**

**naviseccli -address ss1_spa clone -allocatecpl -spA 54 -spB 55 -o**

These commands allocate a clone private LUN to SP A (the LUN with LUN ID 54) and SP B (LUN with ID 55) for clone operations. See also the **-addclone** example earlier.

# clone | snapview -changeclone

**Changes one or more properties of a clone.**

## Description

The **naviseccli clone** or the **naviseccli snapview** command with **-changeclone** changes one or more clone properties. The clone can be in any state when you use this function. If you omit the override option (**-o**), the software displays a message to confirm the change.

## Syntax

**clone -changeclone** or **snapview -changeclone** is used with **naviseccli** (described on page 42) as follows.

**clone | snapview -changeclone -name** *name* | **-CloneGroupUid** *uid* **-cloneid** *id*
**[-RecoveryPolicy manual | auto]**
**[-SyncRate high | medium | low |** *value*]
**[-UseProtectedRestore 0 | 1] [-o]**

## Prerequisites

You must have a user account on the storage system on which you want the command executed.

## Options

**-name** *name*

Specifies the name that you gave to the clone group with the **-createclonegroup** function. Use either this name or the clone group unique ID (16-byte WWN).

**-CloneGroupUid** *uid*

Specifies the 16-byte unique ID the software assigned when the clone group was created. The clone group ID consists of the numbers 0 through 9 and the letters A through F (these letters are not case sensitive). You can use the **-listclonegroup** command function to obtain the clone group ID (see page 85).

**-cloneid** *id*

Specifies the 16-character clone ID the storage system assigned when the clone was added. You can obtain this using the **-listclonegroup** command function. The *id* is numeric.

**-RecoveryPolicy manual | auto**

Specifies the synchronization action the storage system will take after a hardware component (such as an SP) fails. Values are:

- **manual** - Manual recovery. The administrator must act to start synchronization.
- **auto** - Automatic. Synchronization will start automatically using the other path (if available). This is the default.

The value you select also applies if reverse-synchronization was in progress before the failure.

**-SyncRate high | medium | low |** *value*

Sets the interval between writes to a clone being synchronized to **high**, **medium**, or **low**, or a *value* in milliseconds. Default is **medium**. A high rate synchronizes the clone fastest, but may degrade overall performance. A low value has minimal effect on performance, but prolongs the synchronization process.

**-UseProtectedRestore 0 | 1**

Specifies the way the software will react when I/O is received during a reverse sync operation.

**0 -** Updates the clone with the I/O. Default.

**1** - Does not update the clone with I/O.

This property is available only if the option is supported on the targeted storage system.

---

**Note:** When you change the property for the **-UseProtectedRestore** option, SnapView automatically allocates and retains additional memory. This additional memory counts against the total replication software memory budget.

---

**-o**

Executes the command without prompting for confirmation.

**Output**

None if the command succeeds; status or error information if it fails.

**Example**

**naviseccli -address ss1_spa clone -changeclone -name db3_clone -cloneid 0100000000000000 -syncrate high -o**

This command changes the sync rate for the clone **0100000000000000** in clone group **db3_clone**.

# clone | snapview -changeclonefeature

**Enables or disables clone features such as protected restore.**

**Description**

The **naviseccli clone** or the **naviseccli snapview** command with **-changeclonefeature** can globally enable or disable the **Allow Protected Restore** option. This option allows you to use the protected restore feature on a per clone basis. For information on the protected restore feature, refer to "clone | snapview -reversesyncclone" on page 93.

If you omit the override option (**-o**), the software displays a message to confirm the change.

**Syntax**

**clone -changeclonefeature** or **snapview -changeclonefeature** is used with **naviseccli** (described on page 42) as follows.

**clone | snapview -changeclonefeature [-AllowProtectedRestore 0|1] [-o]**

**Prerequisites**

You must have a user account on the storage system on which you want the command executed.

**Options**

**-AllowProtectedRestore 0|1**

Enables or disables the ability to use protected restores on LUNs in a storage system. Be aware that enabling protected restores reserves SP memory and prevents it from being used for any other purpose.

**0** Does not allow protected restores on this storage system. Default.
**1** Allows protected restores on this storage system.

**Note:** When you enable the **-AllowProtectedRestore** option, SnapView automatically allocates and retains additional memory. This additional memory counts against the total replication software memory budget.

**-o**

Executes the command without prompting for confirmation.

**Output**

None if the command succeeds; status or error information if it fails.

**Example** **naviseccli -address ss1_spa clone -changeclonefeature -AllowProtectedRestore 1**

```
Do you really want to perform this operation? y
```

This command allows protected restores on the storage system that contains the SP **ss1_spa**.

# clone | snapview -changeclonegroup

**Changes one or more parameters of an existing clone group.**

## Description

The **naviseccli clone** or the **naviseccli snapview** command with **-changeclonegroup** can change certain settings of an existing clone group.

## Syntax

**clone -changeclonegroup** or **snapview -changeclonegroup** is used with **naviseccli** (described on page 42) as follows:

**clone | snapview -changeclonegroup -name** *name* | **-CloneGroupUid** *uid* [**-description** *description*] [**-newname** *name*] [**-o**] [**-quiescethreshold** *seconds*]

## Prerequisites

You must have a user account on the storage system on which you want the command executed.

## Options

**-name** *name*

Identifies the clone group. Use the name assigned when the clone group was created with **-createclonegroup**. To identify a clone group, you can use either the clone name or the clone group unique ID.

**-CloneGroupUid** *uid*

Specifies the 16-byte unique ID the software assigned when the clone group was created. The clone group ID consists of the numbers 0 through 9 and the letters A through F (these letters are not case sensitive). You can use the **-listclonegroup** command function to obtain the clone group ID (see page 85).

**-description** *description*

Lets you enter a description about the clone group—up to 256 characters. If you want to include any spaces, enclose the entire description in double quotation marks. The quotation marks do not count towards the 256-character limit.

**-newname** *name*

Lets you specify a new name for the clone group. A valid name consists of at least one printable character and must not exceed 64 characters. Each clone group name must be unique on its storage system.

**-o**

Executes the command without prompting for confirmation.

**-quiescethreshold** *seconds*

Lets you specify the amount of time that must pass after an I/O to the source LUN before the software places any of its unfractured clones in the synchronized state. The minimum value is 10, the maximum value is 3600.

**Output**

None if the command succeeds; status or error information if it fails.

**Example**

**naviseccli -address ss1_spa clone -changeclonegroup -name db3_clone -quiescethreshold 30**

```
Do you really want to perform this operation? y
```

This command changes the quiesce threshold for clone group **db3_clone** to 30 seconds.

# clone | snapview -consistentfractureclones

**Splits a set of clones from their source LUNs.**

## Description

The **naviseccli clone** or the **naviseccli snapview** command with **-consistentfractureclones** fractures more than one clone at the same time in order to preserve the point-in-time restartable copy across the set of clones. The SnapView driver will delay any I/O requests to the source LUNs of the selected clones until the fracture has completed on all clones (thus preserving the point-in-time restartable copy on the entire set of clones).

**Note:** A restartable copy is a data state having dependent write consistency and where all internal database/application control information is consistent with a database management system/application image.

After the consistent fracture completes, no group association exists between the clones. If a failure occurs on any of the clones, the consistent fracture will fail on all of the clones. If any clones within the group were fractured prior to the failure, the software will resynchronize those clones. For more information on the effects of fracturing a clone, refer to the **-fractureclone** description on page 76.

## Syntax

**clone -consistentfractureclones** or **snapview -consistentfractureclones** is used with **naviseccli** (described on page 42) as follows.

**clone | snapview -consistentfractureclones -CloneGroupNameCloneId** *name cloneId* **|-CloneGroupUidCloneId** *CGUid cloneId* **[-o]**

## Prerequisites

You must have a user account on the storage system on which you want the command executed.

The clones you want to fracture must be within different clone groups. You cannot perform a consistent fracture on clones belonging to different storage systems. You must specify two or more clones that you want to fracture or an error will be displayed.

**Options**

**-CloneGroupNameCloneId** *name cloneId*

Lets you specify the clone group name and the clone ID for the clones you want to fracture. If you specify the clone group name and clone ID for one clone, you must specify the same information for all the other clones you want to fracture at the same time.

The clone group name is the name you specified when you created the clone group. The clone ID is the 16-character ID the storage system assigned when the clone was added to the clone group. The clone ID is always numeric.

Use the **-listclonegroup** command to display all the clone group names and clone IDs on the storage system (see page 85).

**-CloneGroupUidCloneId** *CGUid cloneId*

Lets you specify the clone group ID and the clone ID for the clones you want to fracture. If you specify the clone group ID and clone ID for one clone, you must specify the same information for all the other clones you want to fracture at the same time.

The clone group ID is the 16-byte unique ID the software assigned when you created the clone group. The clone group ID consists of the numbers 0 through 9 and the letters A through F (these letters are not case sensitive). The clone ID is the 16-character ID the storage system assigned when the clone was added to the clone group. The clone ID is always numeric.

Use the **-listclonegroup** command to display all the clone group IDs and clone IDs on the storage system (see page 85).

**-o**

Executes the command without prompting for confirmation.

**Output**

None if the command succeeds; status or error information if it fails.

**Examples**

**naviseccli -address ss1_spa clone -consistentfractureclones -CloneGroupNameCloneId CGname1 0100000000000000 CGname2 0200000000000000 CGname3 0300000000000000 -o**

or

**naviseccli -address ss1_spa snapview -consistentfractureclones -CloneGroupUidCloneId 50:06:01:60:80:60:03:E5:3A:00:00:00:00:00:00:00 0100000000000000 50:06:01:60:80:60:03:E5:3A:00:00:00:00:00:00:00 0200000000000000 50:06:01:60:80:60:03:E5:3A:00:00:00:00:00:00:00 0300000000000000 -o**

# clone | snapview -createclonegroup

**Creates a new clone group based on a source LUN.**

## Description

The **naviseccli clone** or the **naviseccli snapview** command with **-createclonegroup** creates a new clone group using an existing LUN as a source.

**Note:** When you create a clone group, SnapView automatically allocates and retains additional memory. This additional memory counts against the total replication software memory budget.

## Syntax

**clone -createclonegroup** or **snapview -createclonegroup** is used with **naviseccli** (described on page 42) as follows.

**clone | snapview -createclonegroup -name** *name* **-luns** *lun_number* [**-description** *description*] [**-quiescethreshold** *seconds*] [**-o**]

## Prerequisites

You must have a user account on the storage system on which you want the command executed.

Before allocating LUNS as clone private LUNs, you must bind two LUNs that are at least 1GB in size (one for each SP) in a RAID Group that normally does not see heavy I/O.

Any source LUN that is at least 1 GB in size is eligible to be cloned, except for the following:

- Hot spare LUNs
- Clone LUNs (LUNs participating in any clone group as either a source LUN or a clone LUN)
- Snapshot LUNs
- MetaLUNs
- Private LUNs (LUNs reserved as clone private LUNs, in a reserved LUN pool, or in a write intent log)
- A LUN in the process of a migration

**Note:** With SnapView version 6.24 or later, a clone's source LUN can be a remote mirror LUN (a LUN participating as either a primary or secondary image).

## Post requisites

You will need to add a clone using the **clone | snapview -addclone** command and have the clone synchronize before you can fracture and use it.

You can use the **-changeclonegroup** function to change some of the create parameters.

**Options**

**-name** *name*

Lets you specify the name of the clone group. A valid name consists of at least one printable character and must not exceed 64 characters. Each clone group name must be unique on its storage system.

**-luns** *lun_number*

The *lun_number* identifies the source LUN of the clone group.

**-description** *description*

Lets you enter a description about the clone group—up to 256 characters. If you want to include any spaces, enclose all text with double quotation marks.

**-quiescethreshold** *seconds*

Lets you specify the amount of time that must pass after an I/O to the source LUN before the software places any of its unfractured clones in the synchronized state. The minimum value is 10, the maximum value is 3600.

**-o**

Executes the command without prompting for confirmation.

**Output**

None if the command succeeds; status or error information if it fails.

**Example**

**naviseccli -address ss1_spa clone -createclonegroup -name db3_clone -luns 22 -o**

This command creates a clone group named **db3_clone** for source LUN 22. Refer to the example for **-addclone**, page 57.

# clone | snapview -deallocatecpl

**Deallocates the clone private LUNs.**

**Description**

The **naviseccli clone** or the **naviseccli snapview** command with **-deallocatecpl** removes the LUNs that were previously allocated as clone private LUNs. You can also use the **-deallocatecpl** command to reallocate clone private LUNs, for instance, if a clone private LUN fails or if you decide that you want to use a different LUN as your clone private LUN. If you use the **-deallocatecpl** command to *reallocate* the clone private LUNs, you can do so with existing clone groups and clones. When you reallocate the clone private LUN, the software transfers the information stored in the previous clone private LUN to the new clone private LUN.

**Note:** If you reallocate one clone private LUN, you must reallocate the other clone private LUN.

**Syntax**

**clone -deallocatecpl** or **snapview -deallocatecpl** is used with **naviseccli** (described on page 42) as follows.

**clone | snapview -deallocatecpl [-o]**

**Prerequisites**

You must have a user account on the storage system on which you want the command executed.

You must remove all clones and destroy any clone groups before deallocating the clone private LUNs. You must also globally disable the **Allow Protected Restore** option. To do this, issue the **clone | snapview -changeclonefeature -AllowProtectedRestore 0** command (refer to page 65 for more information on this command).

**Options**

**-o**

Tells the CLI to execute the command without waiting for confirmation.

**Output**

None if the command succeeds; status or error information if it fails.

**Example**

**naviseccli -address ss1_spa clone -deallocatecpl -o**

# clone | snapview -destroyclonegroup

**Deletes a clone group.**

**Description**

The **naviseccli clone** or the **naviseccli snapview** command with **-destroyclonegroup** permanently removes a clone group from the storage system and releases the source LUN specified when you created the clone group.

**Syntax**

**clone -destroyclonegroup** or **snapview -destroyclonegroup** is used with **naviseccli** (described on page 42) as follows.

**clone | snapview -destroyclonegroup -name** *name* | **-CloneGroupUid** *uid* **[-o]**

**Prerequisites**

You must have a user account on the storage system on which you want the command executed.

Before you can destroy a clone group, you must remove all clones in the clone group.

**Options**

**-name** *name*

Identifies the clone group. Use the name assigned when the clone group was created with **-createclonegroup**. To identify a clone group, you can use either the clone name or the clone group unique ID.

**-CloneGroupUid** *uid*

Specifies the 16-byte unique ID the software assigned when the clone group was created. The clone group ID consists of the numbers 0 through 9 and the letters A through F (these letters are not case sensitive). You can use the **-listclonegroup** command function to obtain the clone group ID (see page 85).

**-o**

Tells the CLI to execute the command without waiting for confirmation.

**Output**

None if the command succeeds; status or error information if it fails.

**Example**

**naviseccli -address ss1_spa clone -destroyclonegroup -name db3_clone -o**

# clone | snapview -fractureclone

**Splits a clone from its source LUN.**

## Description

The **naviseccli clone** or the **naviseccli snapview** command with **-fractureclone** splits a clone from its source LUN, allowing a secondary server to access the clone as an independent LUN.

**Note:** To fracture more than one clone at the same time, you must use the **-consistentfractureclones** command. For information on using this command, refer to page 69.

Fracturing a clone or a set of clones (**-consistentfracture**) does not remove the clone(s) from its clone group. It remains a part of the clone group—although other servers can access it as an independent LUN—until you explicitly remove it (**-removeclone**). While a clone remains in its clone group, you can synchronize it again using just the **-syncclone** function.

**Note:** Unless you have additional software that supports same host access, you must assign the clone LUN to a storage group other than the storage group that holds the source LUN(s). You also must assign multiple fractured clones, of the same source LUN(s), to different storage groups. For information on software that supports same host access, refer to the "Prerequisites for setting up clones" on page 47.

Two types of fractures can occur - administrative or system.

An *administrative fracture* will occur after any of the following events:

- The production server performed a fracture operation on the clone or set of clones.
- The secondary server performed a reverse synchronization operation on another clone in the clone group.
- The clone driver receives an error reading or writing to the source or clone during a synchronization or reverse synchronization. Possible errors include a failed sector or a bad block.

  **Note:** To verify if an error occurred, issue the **-listclonegroup** command and determine if the value for **SourceMediaFailure** or **CloneMediaFailure** is set to `TRUE`. If an error did occur, you must correct the failure, then re-issue the synchronization or reverse synchronization operation. If the error persists, contact you EMC service provider.

A *system fracture* will occur after any of the following events:

- The clone driver encountered an internal driver error and fractured the clone as a result. If this occurs, you must correct the error condition, perform a fracture (administrative fracture), and then synchronize the clone.
- The source LUN trespasses to the peer SP, either manually or due to an SP failure. If this occurs, the system fracture is temporary and the clone will automatically synchronize with its source LUN when the source trespasses back to its SP.
- Another replication software uses the clone or its source, for example, to start a SAN Copy session or to create a snapshot of the LUN. If this occurs, the system fracture is temporary and the clone will automatically synchronize with its source LUN after the event completes.

**When to fracture a clone LUN**

You can fracture a clone in any state. However, to make a fractured clone available for server write requests, you must fracture the clone while it is in a *synchronized* or *consistent* state. Once you fracture the clone in either of these states, it transitions to a consistent state.

If you want the clone to contain a predictable point-in-time copy, you must ensure that no server writes are occurring when you fracture the clone. You can determine that no writes have been issued to the source LUN for a period of time equal to the quiesce threshold if you fracture the clone while it is in a synchronized state. However, it does not guarantee that another server write will not occur around the same time you fracture the clone.

If the clone never enters the synchronized state, server writes are continuing. Any server writes to the source that occur after the fracture are not copied to the clone. If you believe you have stopped all I/O to the clone's source LUN, it may be difficult to determine the reason for those writes. For example, they can be due to heartbeats in some clustered systems.

In some cases it is useful to fracture the clone in a consistent state. For instance, you would fracture a clone in a consistent state when:

- you can determine that there are no writes being issued to the source LUN and you prefer not to wait for the quiesce threshold to elapse,
- your application has a hot backup mode in which the application continues to write to the source, but internally understands how to disregard writes generated after your specified point in time,

- you need only a crash consistent image (the image you would have if the server failed or shut down improperly).

If the clone synchronization is detracting from the storage system's performance, you can fracture the clone in any other state and then synchronize it at a time when performance is not an issue.

**Writes to the fractured clone LUN**
When a fractured clone receives a server write request, the software marks the clone as dirty, which indicates that the clone has been modified and is no longer a byte-for-byte copy of its source LUN.

**Writes to the source LUN**
After you fracture the clone, the software does not copy any server write requests made to the source LUN unless you manually perform a synchronization. Synchronizing a fractured clone unfractures the clone and updates the contents on the clone with its source LUN.

## Syntax

**clone -fractureclone** or **snapview -fractureclone** is used with **naviseccli** (described on page 42) as follows.

**clone | snapview -fractureclone -name** *name* **| -CloneGroupUid** *uid* **-cloneid** *id* **[-o]**

## Prerequisites

You must have a user account on the storage system on which you want to execute the command.

## Options

**-name** *name*

Identifies the clone group. Use the name assigned when the clone group was created with **-createclonegroup**. To identify a clone group, you can use either the clone name or the clone group unique ID.

**-CloneGroupUid** *uid*

Specifies the 16-byte unique ID the software assigned when the clone group was created. The clone group ID consists of the numbers 0 through 9 and the letters A through F (these letters are not case sensitive). You can use the **-listclonegroup** command function to obtain the clone group ID (see page 85).

**-cloneid** *id*

Specifies the 16-character clone ID the storage system assigned when the clone was added. You can obtain this using the **-listclonegroup** command function. The *id* is numeric.

**-o**

Tells the CLI to execute the command without waiting for confirmation.

**Output** None if the command succeeds; status or error information if it fails.

**Example** **naviseccli -address ss1_spa clone -fractureclone -name db3_clone -o**

# clone | snapview -listclone

**Displays information about a specific clone LUN.**

## Description

The **naviseccli clone** or the **naviseccli snapview** command with **-listclone** displays information about one clone, not clone groups. You must specify the clone name or clone group ID. You can obtain more specific information with function switches.

## Syntax

**clone -listclone** or **snapview -listclone** is used with **naviseccli** (described on page 42) as follows.

**clone | snapview -listclone -Name** *name* **| -CloneGroupUid** *uid* **[-AvailableForIO] [-CloneCondition] [-cloneid** *id***] [-CloneLuns] [-CloneLunWWNs] [-CloneState] [-CloneMediaFailure] [-IsDirty] [-IsFractured] [-PercentSynced] [-RecoveryPolicy] [-SyncRate] [-UseProtectedRestore 0 | 1]**

## Prerequisites

You must have a user account on the storage system on which you want the command executed.

## Options

**-Name** *name*

Specifies the name of the clones clone group. This is required for getting information on a clone.

**-CloneGroupUid** *uid*

Specifies the 16-byte unique ID the software assigned when the clone group was created. The clone group ID consists of the numbers 0 through 9 and the letters A through F (these letters are not case sensitive). You can use the **-listclonegroup** command function to obtain the clone group ID (see page 85).

**-AvailableForIO**

Displays whether the clone is available for I/O from an attached server (`yes` or `no`).

**-CloneCondition**

Displays the clone condition. Typically, this shows the nature of a faulted clone. The values include **Administratively Fractured**, **System Fractured**, **Waiting on administrator to initiate synchronization**, **Queued to be synchronized**, **Synchronizing**, **Normal**, **Unknown**.

**Note:** If you fractured a clone (**-fractureclone**) or a group of clones (**-consistentfractureclones**), the clone condition will display as Administratively Fractured.

**-cloneid** *id*

Specifies the 16-character clone ID the storage system assigned when the clone was added. You can learn this using the **-listclonegroup** command function. The *id* is case sensitive.

**-CloneLuns**

Displays a list of clone LUN numbers.

**-CloneLUNWWNs**

Displays the WWN of the clone.

**-CloneState**

States are consistent, out-of-sync, remote mirror synchronizing, reverse-out-of-sync, reverse-synchronizing, synchronized or synchronizing.

**-CloneMediaFailure**

Indicates whether an error occurred during I/O with the clone LUN (`yes` or `no`).

**-IsDirty**

Indicates whether the clone received a server write request, which means it is not a byte-for-byte duplicate of its source LUN (`yes` or `no`).

**-IsFractured**

Indicates whether the clone was fractured (`yes` or `no`).

**-PercentSynced**

Displays the percentage of the clone that is synchronized with its source.

**-RecoveryPolicy**

Indicates the recovery policy (see **-addclone**), `Automatic` or `Manual`.

**-SyncRate**

Displays `high,medium,low` or a user-specified positive integer.

**-UseProtectedRestore 0|1**

Indicates the state of the Protected Restore option. Values are `0` (clone may be updated with new I/O during a restore) or `1` (clone may not be updated with new I/O during restore).

**Output** Clone information; see example below.

**Example** **naviseccli -address ss1_spa clone -listclone -name db3_clone**

```
Name:  db3_clone
CloneGroupUid:
50:06:01:60:80:60:03:E5:3A:00:00:00:00:00:00:00

CloneID:  0100000000000000
CloneState:  Synchronized
CloneCondition:  Normal
AvailableForIO:  No
CloneMediaFailure:  No
IsDirty:  No
IsFractured:  No
PercentSynced:  100
RecoveryPolicy:  Auto
SyncRate:  Medium
CloneLUNs:  1
UseProtectedRestore:  No
```

# clone | snapview -listclonefeature

**Displays general information for a storage system that supports clones.**

## Description

The **naviseccli clone** or the **naviseccli snapview** command with **-listclonefeature** and no other switches lists general clone information for a storage system that supports clones. You can obtain more specific information with function switches.

## Syntax

**clone -listclonefeature** or **snapview -listclonefeature** is used with **naviseccli** (described on page 42) as follows.

**clone | snapview -listclonefeature**
**[-AllowProtectedRestore] [-AvailableLUNs] [-CanCreate]**
**[-Cpl] [-CplSPA] [-CplSPB] [-DriverVersion]**
**[-MaxCloneGroupsUsingCPL] [-MaxCloneMemory]**
**[-MaxClonesPerGroup] [-MaxLUNsPerSource] [-MinCplSize]**

## Prerequisites

You must have a user account on the storage system on which you want the command executed.

## Options

**-AllowProtectedRestore**

Displays the state of the storage-system Protected Restore option. The state can be `no`, which indicates that the storage system should update the clone with the I/O, or `yes`, which indicates that it should not update the clone with the I/O. This option is available only if supported on the targeted storage system.

**-AvailableLUNs**

Lists LUN numbers currently available to be used as either a source or clone.

**-CanCreate**

Indicates if a clone group can be created on this system. Valid answers are `yes` or `no`.

**-Cpl**

Indicates whether clone private LUNs are allocated and enabled on this system. Valid answers are `yes` or `no`.

**-CplSPA**

Displays the LUN number of the clone private LUN for SP A.

**-CplSPB**

Displays the LUN number of the clone private LUN for SP B.

**-DriverVersion**

Displays the version of the clone driver.

**-MaxCloneGroupsUsingCPL**

Displays the maximum number of clone groups that will use the clone private LUNs that this system supports.

**-MaxCloneMemory**

Displays the maximum amount of memory that the clone driver can allocate.

**-MaxClonesPerGroup**

Displays the maximum number of clones per clone group.

**-MaxLUNsPerSource**

Displays the maximum number of LUNs allowed in a source.

**-MinCplSize**

Displays minimum size allowed, in blocks, for each clone private LUN.

**Output** Clone information; see example below.

**Example** **naviseccli** -address ss1_spa **clone** -listclonefeature

```
CanCreate:  No
MaxCloneGroupsUsingCPL:  50
MaxClonesPerGroup :  8
MaxLUNsPerSource:  1
MaxCloneMemory:  10485760
Cpl:  Yes
CplSPA:  2
CplSPB:  3
MinCplSize:  1 GB
DriverVersion:  2
AvailableLUNs:
AllowProtectedRestore:  No
```

# clone | snapview -listclonegroup

**Displays clone group information.**

**Description**

The **naviseccli clone** or the **naviseccli snapview** command with **-listclonegroup** and no other switches lists general information for a clone group. You can obtain more specific information with function switches.

**Syntax**

**clone -listclonegroup** or **snapview -listclonegroup** is used with **naviseccli** (described on page 42) or **navicli.jar** as follows.

**clone | snapview -listclonegroup** [**-CloneCount**] [**-CloneGroupUid**] [**-Clones**] [**-CloneLunWWNs**] [**-Description**] [**-InSync**] [**-IsControllingSP**] [**-Name** [*name*]] [**-QuiesceThreshold**] [**-SourceLUNSize**] [**-SourceMediaFailure**] [**-Sources**]

**Prerequisites**

You must have a user account on the storage system on which you want the command executed.

**Options**

**-CloneCount**

Displays the number of clones and clone names defined in the clone group.

**-CloneGroupUid**

Displays the Uid and clone name. The Uid is the 16-byte unique ID assigned when the clone group was created.

**-Clones**

Displays the following information about each clone:

| | |
|---|---|
| `CloneID:` | Identification number of the clone. This number is unique per clone group, not storage system. |
| `CloneState:` | Values include consistent, out-of-sync, remote mirror synchronizing, reverse-out-of-sync, reverse-synchronizing, synchronized or synchronizing |
| `CloneCondition` | Values include Administratively Fractured, System Fractured, Waiting on administrator to initiate synchronization, Queued to be synchronized, Synchronizing, Normal, or Unknown. |
| `AvailableForIO` | Clone LUN availability for I/O from an attached server: yes or no. |
| `CloneMedia Failure` | Whether clone media failure occurred: yes or no. |
| `IsDirty` | Clone containing writes occurred from a source other than the clone group source: yes or no. |
| `IsFractured` | Clone fractured state: yes or no. |
| `PercentSynced` | Percentage of the clone instance that is currently synchronized with its source. Also applies to reverse-sync operations. |
| `RecoveryPolicy:` | A user-defined value, Automatic or Manual. |
| `SyncRate:` | A user-defined value: high/medium/low. |
| `CloneLUNs` | List of clone LUN numbers. |
| `UseProtected Restore` | State of the clone Protected Restore option: yes or no. |

**-CloneLUNWWNs**

Displays the WWN of the clone.

**-Description**

Displays the descriptions and clone names in the clone group.

**-InSync**

Indicates whether the source LUN is available for I/O.

**-IsControllingSP**

Indicates whether the SP specified in the command is the SP that owns the clone group (`yes` or `no`).

**-Name** *name*

Displays information for the clone group *name* as given to the clone group with the **-createclonegroup** function.

**-QuiesceThreshold**

Displays the amount of time that must pass after an I/O to the source LUN before the software places any of its unfractured clones in the Synchronized state. The minimum value is 10, the maximum value is 3600.

**-SourceLUNSize**

Displays the names and combined size of the source LUNs.

**-SourceMediaFailure**

Indicates whether an error occurred during I/O with the source LUNs.

**-Sources**

Displays the LUN numbers that constitute the source in the clone group.

**Output** Clone group information; see example below.

**Example** **naviseccli -address ss1_spa clone -listclonegroup**

```
Name:  db3_clone
CloneGroupUid:
50:06:01:60:80:60:03:E5:3A:00:00:00:00:00:00:00
InSync:  Yes
Description:  database3 source lun
QuiesceThreshold:  60
SourceMediaFailure:  No
IsControllingSP:  No
SourceLUNSize:  250000
CloneCount:  1
Sources:  0
Clones:
CloneID:  0100000000000000
CloneState:  Synchronized
CloneCondition:  Normal
AvailableForIO:  No
CloneMediaFailure:  No
IsDirty:  No
PercentSynced:  100
RecoveryPolicy:  Auto
SyncRate:  Medium
CloneLUNs:  1
UseProtectedRestore:  No
IsFractured:  No
```

# clone | snapview -removeclone

**Removes a clone LUN from a clone group.**

## Description

The **naviseccli clone** or the **naviseccli snapview** command with **-removelone** removes a clone from its clone group. When you remove a clone from a clone group, the clone becomes a conventional (regular) LUN and is no longer associated with its source or clone group.

The clone group itself persists until you remove it using **-destroyclonegroup**.

## Syntax

**clone -removeclone** or **snapview -removeclone** is used with **naviseccli** (described on page 42) as follows.

**clone | snapview -removeclone -name** *name* | **-CloneGroupUid** *uid* **-cloneid** *id* **[-o]**

## Prerequisites

You must have a user account on the storage system on which you want the command executed.

To remove a clone from its clone group, the clone:

- must be fractured or in a synchronized or consistent state.
- *cannot* be in a synchronizing or reverse synchronizing state.
- *cannot* be in a queue to be synchronized.

**Note:** In an effort to control resource consumption, the SnapView driver limits a maximum of 40 concurrent synchronizations and/or reverse synchronizations per SP. Once this limit is met, the software queues all other synchronization and reverse synchronization requests.

- *cannot* be removed if it is in a reverse-out-of-sync state and it is the only clone in the clone group. If you remove a clone under these conditions, you cannot restore it. It is recommended that you perform a reverse synchronization operation from the clone and, when the reverse synchronization is successful, you should remove the last clone from the clone group.

**Options**

**-name** *name*

Identifies the clone group. Use the name assigned when the clone group was created with **-createclonegroup**. To identify a clone group, you can use either the clone name or the clone group unique ID.

**-CloneGroupUid** *uid*

Specifies the 16-byte unique ID the software assigned when the clone group was created. The clone group ID consists of the numbers 0 through 9 and the letters A through F (these letters are not case sensitive). You can use the **-listclonegroup** command function to obtain the clone group ID (see page 85).

**-cloneid** *id*

Specifies the 16-character clone ID the storage system assigned when the clone was added. You can obtain this using the **-listclonegroup** command function. The *id* is numeric.

**[-o]**

Tells the CLI to execute the command without waiting for confirmation.

**Output**

None if the command succeeds; status or error information if it fails.

**Example**

**naviseccli -address ss1_spa clone -removeclone -name db3_clone -o**

# clone | snapview -resetfracturelog

**Marks all data chunks of the source or clone LUN as being modified in the clone private LUN log. This allows you to perform a *full* synchronization or reverse-synchronization.**

## Description

The **naviseccli clone** or the **naviseccli snapview** command with **-resetfracturelog** simulates server writes to the source or clone LUN, marking all data chunks on the LUN as being modified (changed).

## Syntax

**clone -resetfracturelog** or **snapview -resetfracturelog** is used with **naviseccli** (described on page 42) as follows.

**clone | snapview -resetfracturelog -name** *name* | **-CloneGroupUid** *uid*
**-cloneid** *id* **[-o]**

## Prerequisites

You must have a user account on the storage system on which you want the command executed.

You can use this switch only if the clone was administratively fractured and the clone is in an *out-of-sync* or *reverse-out-of-sync* state.

## Post requisites

After issuing the **-resetfracturelog** command, you must issue a synchronization or reverse-synchronization. This will initiate a *full* synchronization or reverse-synchronization since all data chunks were marked as being modified.

---

**Note:** Performing multiple *full* synchronizations or reverse-synchronizations at the same time, may affect performance on other system resources.

---

## Options

**-name** *name*

Identifies the clone group. Use the name assigned when the clone group was created with **-createclonegroup**. To identify a clone group, you can use either the clone name or the clone group unique ID.

**-CloneGroupUid** *uid*

Specifies the 16-byte unique ID the software assigned when the clone group was created. The clone group ID consists of the numbers 0 through 9 and the letters A through F (these letters are not case sensitive). You can use the **-listclonegroup** command function to obtain the clone group ID (see page 85).

**-cloneid** *id*

Specifies the 16-character clone ID the storage system assigned when the clone was added. You can obtain this using the **-listclonegroup** command function. The *id* is numeric.

**[-o]**

Tells the CLI to execute the command without waiting for confirmation.

**Output** None if the command succeeds; status or error information if it fails.

**Example**

```
naviseccli  -address ss1_spa  clone -resetfracturelog
-name db3_clone
```

# clone | snapview -reversesyncclone

**Overwrites the source LUN with data from the clone LUN.**

## Description

The **naviseccli clone** or the **naviseccli snapview** command with **-reversesyncclone** replaces the data on the source LUN with the data on the clone. This allows you to revert to an earlier copy of the source LUN, for instance if the source became corrupted.

To ensure that there is no data corruption on the source LUN, you have to take the source LUN offline before you initiate the reverse synchronization. Once the operation begins, you can bring the source LUN back online.

When you initiate a reverse synchronization from Navisphere, the software immediately fractures all the clones in the clone group and then the software unfractures the clone from which you initiated the reverse synchronization. The software then copies the data chunks that differ between the source and the clone to the source LUN. The source LUN can instantly access the data, while the actual copying continues in the background.

**Writes to the source LUN**
Any server writes made to the source LUN after the clone was originally fractured are overwritten with the data on the clone. Any unmodified (unchanged) chunks of data are not rewritten to the source. After the reverse synchronization has completed, the clone that initiated the reverse synchronization remains unfractured.

---

**Note:** If you issue the **-UseProtectedRestore 1** switch with the **clone | snapview -reversesyncclone** command, after the reverse synchronization has completed, SnapView *fractures* the clone that initiated the reverse synchronization.

---

During a reverse synchronization, the software automatically copies any server writes made to the source LUN to the clone. If you do not want source writes copied to the clone during a reverse synchronization, you must issue the **-UseProtectedRestore 1** switch with the **clone | snapview -reversesyncclone** command.

---

**Note:** If you modify the same data chunk on the clone more than once, the software copies only the last modification to the source LUN.

---

The **Protected Restore** feature protects the data on a clone during a reverse synchronization. When you select this feature, during a reverse synchronization the software will not copy any server writes made to the source LUN to the clone. Instead, the software records information in the clone private LUN to identify the source LUN writes for subsequent synchronizations.

Once you initiate a reverse synchronization, the software immediately unfractures the clone that initiated the reverse synchronization. Then the software fractures any other clones in the clone group in order to protect them from corruption should the reverse synchronization operation fail. The software then begins to copy its data to its source LUN. After the reverse synchronization has completed, the software fractures the clone that initiated the reverse synchronization.

You can enable this feature on a per-clone basis (and not on a per-clone-group basis) when you first add a clone to a clone group (**clone | snapview -addclone -UseProtectedRestore 1**) or at any time *before* you perform a reverse synchronization (**clone | snapview -listclone -UseProtectedRestore 1**).

---

**Note:** Before you can initiate the **Protected Restore** feature, you must globally enable it by issuing the **clone | snapview -changeclonefeature [-AllowProtectedRestore 1]** command. When you select this option, the SnapView driver automatically allocates 8 MB in additional memory per SP. The additional memory is fixed and is used to monitor modified blocks on the source LUN, in order to prevent these blocks from being overwritten by the clone during a reverse synchronization. This additional memory counts against the total replication software memory budget.

---

While the clone is reverse synchronizing, you *cannot*

- add a clone to the clone group
- remove the clone that is reverse synchronizing from the clone group
- synchronize any clone in the clone group
- reverse synchronize any clone in the clone group

## Syntax

**clone -reversesyncclone** or **snapview -reversesyncclone** is used with **naviseccli** (described on page 42) as follows.

**clone | snapview -reversesyncclone -name** *name* **| -clonegroupUid** *uid* **-cloneid** *id* **[-UseProtectedRestore 0 | 1] [-o]**

**Prerequisites** You must have a user account on the storage system on which you want the command executed.

**IMPORTANT**

**You cannot reverse synchronize the clone if the clone's source is a MirrorView secondary image or a MirrorView primary image with unfractured secondaries.**

To reverse synchronize a fractured clone, the following must be true:

- The clone is *not* in an out-of-sync state.
- Another clone in the clone group is *not* in a synchronizing or reverse synchronizing state.

For a Windows operating system, you must disable the indexing service and recycle bin on the source LUN of the clone. This prevents data corruption during reverse synchronization.

**Options** **-clonegroupUid** *uid*

Specifies the 16-byte unique ID the software assigned when the clone group was created. The clone group ID consists of the numbers 0 through 9 and the letters A through F (these letters are not case sensitive). You can use the **-listclonegroup** command function to obtain the clone group ID (see page 85).

**-cloneid** *id*

Specifies the 16-character clone ID the storage system assigned when the clone was added. You can obtain this using the **-listclonegroup** command function. The *id* is numeric.

**-name** *name*

Identifies the clone group. Use the name assigned when the clone group was created with **-createclonegroup**. To identify a clone group, you can use either the clone name or the clone group unique ID.

**-o**

Tells the CLI to execute the command without waiting for confirmation.

**-UseProtectedRestore 0 | 1**

Specifies the way the software will react when I/O is received during a reverse sync operation.

**0** Updates the clone with I/O to the source LUN.
**1** Does not update the clone with I/O to the source LUN.

This property is available only if the option is supported on the targeted storage system.

**Output** None if the command succeeds; status or error information if it fails.

**Example** **naviseccli -address ss1_spa clone -reversesyncclone -name db3_clone -cloneid 0100000000000000 -o**

# clone | snapview -syncclone

**Starts copying the data on a source LUN to its clone.**

## Description

The **naviseccli clone** or the **naviseccli snapview** command with **-syncclone** begins the process of synchronization: the copying of data on the source LUN to its clone. The purpose of synchronizing a fractured clone is to update the contents on the clone with its source LUN. Once you synchronize a fractured clone, the clone becomes unfractured and behaves the same as a newly added clone in a clone group.

When you initiate a synchronization, the software copies the data chunks that are different between the source LUN and the clone LUN to the clone. The software also copies any server writes from the source LUN to the clone LUN. The software does not copy any unmodified chunks to the clone.

---

**Note:** If the same chunk of data is modified on the source LUN more than once, only the last modification is copied to the clone.

---

While the clone is synchronizing, you:

- *cannot* remove the clone in a synchronizing state,
- *cannot* perform a reverse synchronization with any other clone in the clone group.

---

**Note:** In an effort to control resource consumption, the SnapView driver limits a maximum of 40 concurrent synchronizations and/or reverse synchronizations per SP. Once this limit is met, the software queues all other synchronization and reverse synchronization requests.

---

## Syntax

**clone -syncclone** or **snapview -syncclone** is used with **naviseccli** (described on page 42) as follows.

**clone | snapview -syncclone -name** *name* | **-CloneGroupUid** *uid* **-cloneid** *id* **[-o]**

## Prerequisites

You must have a user account on the storage system on which you want the command executed.

To synchronize a fractured clone, the following must be true:

- The clone *cannot* be in a reverse-out-of-sync state.

- Another clone in the clone group *cannot* be in a reverse synchronizing state.
- If a clone is system fractured, you must manually fracture the clone before you can synchronize it.

For a Windows operating system, the indexing service and recycle bin must be disabled on the clone LUN. This prevents data corruption during the synchronization.

**Options**

**-name** *name*

Identifies the clone group. Use the name assigned when the clone group was created with **-createclonegroup**. To identify a clone group, you can use either the clone name or the clone group unique ID.

**-CloneGroupUid** *uid*

Specifies the 16-byte unique ID the software assigned when the clone group was created. The clone group ID consists of the numbers 0 through 9 and the letters A through F (these letters are not case sensitive). You can use the **-listclonegroup** command function to obtain the clone group ID (see page 85).

**-cloneid** *id*

Specifies the 16-character clone ID the storage system assigned when the clone was added. You can obtain this using the **-listclonegroup** command function. The *id* is numeric.

**-o**

Tells the CLI to execute the command without waiting for confirmation.

**Output**

None if the command succeeds; status or error information if it fails.

## Examples

**naviseccli -address ss1_spa clone -listclonegroup**

```
Name:  db3_clone
CloneGroupUid:
50:06:01:60:20:04:21:3E:0C:00:00:00:00:00:00:00
InSync:  Yes
Description:
QuiesceThreshold:  60
SourceLUNSize:  2097152
CloneCount:  2
Sources:  16
Clones:
CloneID:  0100000000000000.
.
```

**Note:** Before issuing the synchronization command, you should flush all cached data to the source LUN.

**naviseccli -address ss1_spa clone -syncclone -name db3_clone -cloneid 0100000000000000 -o**

These commands list the state of all clone groups on storage system SP **ss1_spa**, and then start synchronization of the clone **db3_clone**.

## What next?

For step-by-step examples on using admsnap and CLI for clones, refer to Chapter 6.

# 3

# Admsnap Commands for Clones

This chapter explains how to use the admsnap utility software with SnapView clones.

**Note:** CX4 series storage systems do not support Windows 2000 servers as an attached host.

The clone feature is not supported on AX4-5 series or AX series storage systems.

Major topics in this chapter are:

## About admsnap software and clones

The admsnap utility is an executable program that you can run interactively with a script to manage clones. The admsnap utility resides on the servers connected to the storage system with the SnapView driver. You must execute the following admsnap commands from one of these attached servers:

- **admsnap clone_activate** – scans for new devices and, for Windows, assigns a drive letter.
- **admsnap clone_deactivate** – flushes server buffers and removes the drive letter assigned with clone_activate (Windows only).
- **admsnap flush** – flushes data to the specified source LUN. Use this command before synchronizing and fracturing a clone.
- **admsnap list** – lists devices and their World Wide Name (WWN) within the operating system.

---

**Note:** The admsnap software supports native device names and PowerPath® pseudo-device names. If you use a PowerPath pseudo-device name, you can add this device name to the **etc/fstab** file (for Linux) or to the **etc/vfstab** file (for Solaris).

Admsnap only supports device naming schemes for Veritas in Linux and HP-UX, and device naming schemes and enclosure based naming schemes in Solaris.

---

For a step-by-step overview and examples on setting up and using clones with admsnap and Navisphere CLI, refer to Chapter 6.

# Requirements for admsnap and clones

Before you can use admsnap for clones, the following must be true:

### General requirements

- The server must have access to Navisphere CLI. This is required to create a prerequisite LUN called a clone private LUN and to create clones.
- The storage-system must have the SnapView software installed.
- The production and secondary servers must have both the Navisphere Host Agent and admsnap software installed. To install the host agent and admsnap, refer to the EMC CLARiiON server support products guide for your operating system.

### Platform-specific requirements

- For HP-UX, admsnap requires a volume group entry for each path to the storage system. The entry or entries can be contained in a single volume group or multiple volume groups. On the destination storage system, the volume group entry or entries must specify a clone.

# admsnap clone_activate

**Scans for new LUNs on the secondary server.**

## Description

The **admsnap clone_activate** command scans for new devices and, for Windows, assigns a drive letter to every new device it finds. When you activate a clone, the clone becomes accessible on the secondary server.

**Note:** Although the **clone_activate** command is prefixed with the term clone, you may issue this command on any device that requires a scan for new LUNs and, for Windows only, for the assignment of a drive letter.

Depending on your operating system, the **admsnap clone_activate** command will do the following:

| Server OS | Function description |
|---|---|
| **Windows** | Scans for new devices and assigns a drive letter to every new device it finds.<br>If the clone was already mounted on the secondary server, a reboot is recommended but not required after you activate the fractured clone.<br>**For Windows Terminal Services Edition and Remote Desktop** - If a terminal server is running on the secondary server, you must remove and disable it before using the **clone_activate** command. If you have the terminal server enabled, it prevents the command from working properly. |
| **AIX, HP-UX,** and **Solaris** | Rescans the bus for new devices. |
| **Linux** | The **clone_activate** command is not supported. You must either reboot the server or unload and load the HBA driver. |
| **NetWare** | The **clone_activate** command is not supported. Run the command **list devices** or use the command **scan all LUNs** on the console. |

## Syntax

**admsnap clone_activate [-h]**

## Prerequisites

You must have administrative privileges on the secondary server.

Before this command will work, a clone group must exist (**clone | snapview -createclonegroup**) and a clone must already exist (that is, you must start one by adding a LUN to the clone group using **clone | snapview -addclone**) and the clone must be fractured.

**Options** **[-h]**

Lists the help output for a given operation. If you use this option on the command line, a help message is displayed for each option you specify. Admsnap will not perform the command, it lists the help output only.

This option is valid for Windows, Solaris, HP-UX, and AIX systems *only*.

**Example** **admsnap clone_activate**

# admsnap clone_deactivate

**Flushes server buffers and removes assigned drive letter. This command is supported on Windows systems only.**

## Description

On a Windows server, the **admsnap clone_deactivate** command flushes all server buffers and removes the drive letter assigned by **clone_activate**. When you deactivate a clone, the software takes the clone offline making it inaccessible to the secondary server.

---

**Note:** Although the **clone_deactivate** command is prefixed with the term clone, you may issue this command on any Windows device that requires a volume to be taken offline, making it inaccessible to the secondary server.

---

| Server OS | Function description |
|---|---|
| **Windows** | Flushes all server buffers and removes the drive letter assigned by **clone_activate**. This operation will attempt to take the device offline and to make it inaccessible by the operating system. To bring the device back online, you must issue the **clone_activate** command.<br>**For Windows Terminal Services Edition** - If a terminal server is running on the secondary server, you must remove and disable it before using the **clone_deactivate** command. If you have the terminal server enabled, it prevents the **clone_deactivate** command from working properly.<br>**For Microsoft Cluster Services (MSCS)** - When you need to stop I/O to a clone or a clone's source LUN, you must take the clone LUN offline in order to ensure MSCS stops generating I/O. You must do this in addition to using the **admsnap clone_deactivate** command. You can take the clone LUNs offline by doing the following:<br>1. Locate the MSCS Disk Group resource that includes the clone LUNs.<br>2. Use the Microsoft Cluster Administrator to take the Disk Group offline by right-clicking the Disk Group and selecting **Take Offline**.<br>3. Perform the clone operation.<br>When you are ready to resume I/O to the disk, use the Microsoft Cluster Administrator to bring the Disk Group(s) back online by right-clicking the Disk Group and selecting **Bring Online**. |
| **AIX, HP-UX,** and **Solaris** | The **clone_deactivate** command is not supported; it is provided for script compatibility between Windows and UNIX only. If you issue the command on this server, admsnap will return an error.<br>Unmount the file system by issuing the **umount** command. If you cannot unmount the file system, you can use the **sync** command to flush buffers. The **sync** command is not considered a substitute for unmounting the file system, but you can use it to reduce the number of times you have to issue the **fsck** command on your backup server's file system. Refer to your system's man pages for **sync** command usage. |
| **Linux** | The **clone_deactivate** command is not supported. You must either reboot the server or unload and load the HBA driver. |
| **NetWare** | The **clone_deactivate** command is not supported. Use the **dismount** command on the clone volume to dismount the file system. |

### Syntax

**admsnap clone_deactivate** [**-h**] **-o** *object_name* [**-t** *object_type*]

### Prerequisites

You must have administrative privileges on the secondary server.

You must use this command on a secondary server *after* the clone has been activated and *before* you synchronize the fractured clone or on the production server *before* you reverse synchronize a fractured clone.

### Options

[**-h**]

Lists the help output for a given operation. If you use this option on the command line, a help message is displayed for each option you specify. Admsnap will not perform the command, it lists the help output only.

**-o** *object_name*

Specifies an object name. The object name must be a device name or drive letter. If the device name is provided as an object name, admsnap will find the associated device and drive letter. It will then delete the drive letter, dismount the device and take the device offline.

If you use the **-t** option, the device name must be a valid name for the type of object specified with the **-t** option.

[**-t** *object_type*]

Specifies the object type. Valid Windows object types are device and drive letter.

If you do not specify the **-t** option, admsnap will attempt to determine the correct object type and report a failure if it cannot.

**Example** Windows example:

**admsnap clone_deactivate -o e:**

# admsnap flush

**Flushes all data that has been written to the source LUN.**

## Description

The **admsnap flush** command clears buffers and flushes all data to a LUN.

Use it on the source LUN's production server before synchronizing or fracturing a clone to ensure that all cached data has been written to the clone.

Depending on your operating system, the **admsnap flush** command will do the following:

| Server OS[a] | Function description |
|---|---|
| **Windows** | Writes the data from the cache to the disk. |
| **AIX, HP-UX, Linux,** and **Solaris** | Unmount the file system by issuing the **umount** command. If you are unable to unmount the file system, you can issue the admsnap **flush** command[b]. |
| **NetWare** | The **flush** command is not supported. Dismount the file system by using the **dismount** command on the volume. |

a. With some operating systems, you may need to shut down the application to flush the data. Specific operating systems have different requirements.

b. Neither the **flush** command nor the **sync** command is a substitute for unmounting the file system. Both commands only complement unmounting the file system.

## Syntax

**admsnap flush** [**-h**] **-o** *object_name* [**-t** *object_type*]

## Prerequisites

You must have administrative privileges on the production server.

## Options

[**-h**]

Lists the help output for a given operation. If you use this option on the command line, a help message is displayed for each option you specify. Admsnap will not perform the command; it lists the help output only.

This option is valid for Windows, Solaris, HP-UX, Linux, and AIX systems *only*.

**-o** *object_name*

Specifies an object name. Valid Windows object names are drive letter and Physical drive name. Valid Solaris object names are device names, file system names, Veritas volume names, or

volume group names. For HP-UX and Linux, a valid object name is either a device name, a file system name or Veritas volume name. For AIX, a valid object name is either a device name or a file system name.

If you use the **-t** option, the device name must be a valid name for the type of object specified with the **-t** option.

[**-t** *object_type*]

Specifies the object type. This switch is supported on Windows or Solaris operating systems only. Valid Windows object types are device and drive letter. Valid Solaris object types are device, volume, and file system.

If you do not specify the **-t** option, admsnap will attempt to determine the correct object type and report a failure if it cannot.

**Example** Windows example:

**admsnap flush -o E:**

# admsnap list

**Performs a list operation.**

## Description

The **admsnap list** command will attempt to list devices and their World Wide Name (or WWN) within the operating system. You can use the WWN to distinguish one operating system device from another.

The **admsnap list** command is supported on Windows, Solaris, HP-UX, Linux, and AIX systems *only*.

## Syntax

**admsnap list** [**-h**] [**-l** *lun_wwn*] [**-d** *object_name*] [**-a** *output_format*]

## Prerequisites

You must have administrative privileges on the production and/or secondary server.

## Options

[**-h**]

Lists the help output for a given operation. If you use this option on the command line, a help message is displayed for each option you specify. Admsnap will not perform the command; it lists the help output only.

This option is valid for Windows, Solaris, HP-UX, Linux, and AIX systems.

[**-l** *lun_wwn*]

Lists the device name for the WWN you specify. The WWN must be in a valid WWN format.

[**-d** *object_name*]

Specifies an object name that must be a device name, file system mount point, a Veritas volume name, or a drive letter.

---

**Note:** The **-a** option is ignored when used with the **-d** option and the object name argument is a Windows device name, for instance **\\.\PhysicalDrive1**.

---

[**-a** *output_format*]

Specifies an alternative output style on a Windows server. The default Windows output style is to list the drive letter and its WWN. The output style must be a valid style that is either a drive letter (default if **-a** not specified at all) or a physical drive.

**Note:** The **-a** option is ignored when used with the **-d** option and the object name argument is a Windows device name, for instance **\\.\PhysicalDrive1**.

**Example**

For Windows:

**admsnap list -d E:**

# 4

# SnapView Snapshot CLI Commands

This chapter explains the SnapView snapshot CLI commands that manage SnapView snapshots.

**Note:** If you currently use Classic CLI to issue CLI commands, please note that Secure CLI is replacing Classic CLI; therefore, only Secure CLI will support new features. We recommend that you use Secure CLI. Secure CLI is faster and more secure than Classic CLI. Only commands issued to host agents are not currently supported using Secure CLI.

Major topics in this chapter are:

# About Secure CLI

Secure CLI is a comprehensive Navisphere CLI solution that provides one application and one security model for all CLI commands. Secure CLI provides role-based authentication, audit trails of CLI events, and SSL-based data encryption. You do not need to install a JRE to run Secure CLI.

**Note:** Refer to the Host Agent/CLI and Utilities Release Notes, available on Powerlink, for a list of supported operating systems. You must be running FLARE Operating Environment version 02.19.xxx.5.yyy or higher.

Secure CLI commands run in a command window. Each command consists of the **naviseccli** command (and switches) together with another subcommand (and its switches).

**Note:** For commands that originated in Classic CLI, some command output may be enhanced; for example, Secure CLI can retrieve and display information from peer SPs. For Classic CLI commands that produce multiple warnings and require multiple confirmations, Secure CLI provides a single summary of warnings and a single confirmation. In general, Secure CLI preserves original command syntax and output for script compatibility.

Secure CLI does not support commands targeted to host agents, such as **-lunmapinfo**. For commands that you can issue to either an SP or host agent, such as **–getagent**, if you issue the command to a host agent, Secure CLI displays an error message. You must use Classic CLI to issue commands to host agents. For a list of commands that are not supported, refer to the *EMC Navisphere Command Line Interface (CLI) Reference*.

**Note:** Secure CLI does not distinguish case of characters, so, regardless of the host operating system, you can use either uppercase, lowercase, or any combination of characters as you type commands.

If a Secure CLI command fails and the CLI does not generate its own error message, it displays an error message from the SP Agent. Secure CLI generates errors about command line syntax for commands and options and their values.

Secure CLI commands return 0 if the command is successful, or greater than zero if the command is unsuccessful.

# Getting started with Secure CLI

Before you begin to issue Secure CLI commands, you must create a Navisphere user account on the storage system. To create the required user accounts using Navisphere CLI, refer to the *EMC Navisphere Command Line Interface (CLI) Reference*. For details on using Navisphere 6.X security, refer to the Navisphere Manager online help.

You can also choose to configure a Navisphere 6.X security file to issue Secure CLI commands (see the next section, "Overview of using Navisphere 6.X security with Secure CLI") on the host. If you establish a security file, you do not need to include the switches **-user**, **-scope,** and -**password** (or the password prompt), in each command you issue.

**Note:** Establishing a security file to use commands that originated in Classic CLI ensures that other than the update from navicli to the naviseccli command, you do not need to modify any established scripts you may have. Secure CLI, unlike Classic CLI, requires the switches **-user**, **-scope**, and **-password** (or the password prompt) in each command line; you do not need to provide these switches in the command line if you establish a security file.

## Overview of using Navisphere 6.X security with Secure CLI

A storage system will not accept a command from Secure CLI unless the user who issues the command has a valid user account on the storage system. You can specify a valid account username, password, and scope (global, local, or lightweight directory access protocol (LDAP)) for each command you issue, or, more conveniently, you can create a Navisphere security file.

The Navisphere security file is an encrypted file stored for each user on each host. You can add or delete a user security file using the **-AddUserSecurity** or **-RemoveUserSecurity** functions as arguments to the **naviseccli** command. You cannot copy a security file to another host. You must issue the **-AddUserSecurity** function on the host for which you want to create the security file.

When you create a security file, the username you use to log in to the current host is automatically stored in the security file, or you can specify an alternative username for the security file in the **-AddUserSecurity** request using the optional **-user** switch. If you omit the -**user** switch, the security file uses your current username.

For example, to add yourself to the security file on the current host, given the alternative username **altusername**, the password **mypass** and the scope **0** (global scope), type:

**naviseccli -address SP-IP -AddUserSecurity -password mypass -scope 0 -user altusername**

Then, on this host, you can enter CLI commands to any storage system on which you have an account that matches the username **altusername**, with password **mypass** and global scope (**scope 0**).

---

**Note:** Username and password are case sensitive.

---

The security file is stored in your default home directory. With Secure CLI, you can specify an alternative file path using the optional **-secfilepath** switch.

---

**Note:** If you specify an alternative location for the security file, you must specify the file path in every subsequent CLI command you issue to ensure the CLI locates the security file.

---

To save the example used above to the alternative location "c:\altlocation\" type:

**naviseccli -AddUserSecurity -password mypass -scope 0 -user altusername - secfilepath c:\altlocation**

Then, for each subsequent command you issue, you must specify the **-secfilepath** switch with the security file path location "c:\altlocation\" in the command line.

# naviseccli

**Sends status or configuration requests to a storage system via command line.**

## Description

The **naviseccli** command sends storage-system management and configuration requests to a storage system via the Internet.

## Syntax

The **naviseccli** command is used as follows:

**naviseccli -help**
or
**naviseccli**

[**-address** *IPaddress* | *NetworkName* | **-h** *IPaddress* | *NetworkName*]
[**-AddUserSecurity**]
[**-f** *filename]*
[**-m**]
[-**nopoll** | **-np**]
[-**parse** | **-p**]
[-**password** *password*]
[-**port** *port*]
[**-q**]
[-**RemoveUserSecurity**]
[-**scope 0** | **1** | **2**]
[**-secfilepath** *filepath*]
[-**timeout** | **-t** *timeout*]
[-**user** *username*]
[**-v**]
[**-xml**]
*CMD* [*optional_command_switches*]

## Prerequisites

To use the **naviseccli** command, you must be running Navisphere CLI 6.X or later.

## Options

**-help**

Displays the help screen and does not start the **naviseccli** process. To start the **naviseccli** process, use one or more of the switches that follow instead.

**-address** *IPaddress* | *NetworkName* | **-h** *IPaddress* | *NetworkName*

Specifies the IP address or network name of the targeted SP on the desired storage system. The default, if you omit this switch, is **localhost**.

**-AddUserSecurity**

Directs the CLI to add user security information to the security file on this server. You must use the **-scope** switch to add scope information to the security file. You can use the -**password** switch or enter your password into the password prompt (see -**password**), to supply the required password information to the security file. The **-user** and **-secfilepath** switches are optional with this command.

---

**Note:** If you specify the -**user** switch, you can create an alternative username to your server login name in the security file you create on this server. If you use the -**secfilepath** switch, you can specify an alternative location to your default home directory, for the security file on this server. You must then use the -**secfilepath** switch in each subsequent command you issue.

---

**-f** *filename*

Specifies to store data in a file.

**-m**

Suppresses output except for values. This option is most useful when used as part of a script.

---

**Note:** Only supported for commands that originated in Classic CLI.

---

**-nopoll** | **-np**

Directs the feature provider not to issue a poll request. This switch significantly increases performance when dealing with large or multiple storage systems. The feature provider automatically polls unless this switch is specified.

---

**Note:** When the **-nopoll** switch is set, **get** commands may return stale data and **set** commands may erase previously changed settings. Use caution when the **-nopoll** switch is set.

---

**-parse | -p**

Directs the CLI to validate the command. The CLI verifies the command syntax and displays a message stating whether the command was valid. The CLI takes no other action.

**-password** *password*

Specifies the password on the storage system you want to log in to. The password is visible in the command line. Passwords are case sensitive.

If you want to mask the password, and you are not using a security file, you can omit this switch from the command line. The CLI then prompts you to enter a password. The information you enter into the password prompt is concealed.

---

**Note:** You can omit this switch if you are using a security file. See -**AddUserSecurity**.

---

**-port** *portnumber*

Sets the port (type) of the storage system. The default is 443. If you choose to change the default port number, management port 2163 will be supported; however, you will need to specify the -**port** switch and number 2163 in every subsequent command you issue.

**-q**

Suppresses error messages. This switch is useful when included as part of a script.

---

**Note:** Only supported for commands that originated in Classic CLI.

---

**-RemoveUserSecurity**

Directs the CLI to remove user security information about the current user from the security file on this server.

**-secfilepath** *filepath*

Stores the security file in a file path location you specify. When you create a security file on a server using the **-addusersecurity** command, the security file is saved to your default home directory. If you want to store the security file in an alternative location, you can use the optional **-secfilepath** switch with the **-addusersecurity** command.

---

**Note:** If you use the **-secfilepath** switch to set up an alternative path for your security file, you must use this switch in every subsequent CLI command you issue, to ensure the CLI locates the security file.

---

**-scope 0 | 1 | 2**

Specifies whether the user account on the storage system you want to log in to is global, local, or lightweight directory access protocol (LDAP). **0** (default) indicates global, **1** indicates local, and 2 indicates LDAP.

A global account is effective throughout the domain. When the administrator creates a global account, the software copies the definition of this account to the domain directory, which makes it accessible on all storage systems in the domain.

A local account is effective on only the storage systems for which the administrator creates the account. The user can log in to only those storage systems on which he has a local account.

LDAP maps the username/password entries to an external LDAP or active directory server for authentication.
Username/password pairs whose roles are not mapped to the external directory will be denied access.

---

**Note:** For more information on LDAP refer to the *EMC Navisphere Command Line Interface (CLI) Reference* or the Navisphere Manager online help.

---

**-timeout** | **-t** *timeout*

Sets the timeout value in seconds. The default is 600 seconds.

**-user** *username*

Specifies the username on the storage system you want to log in to. Usernames are case sensitive. You can omit this if your username has been added to the security file.

**Note:** You can use this switch when establishing a security file, to specify an alternative username. See -**AddUserSecurity.**

**-v**

Enables verbose error descriptions. This is the default unless **-q** is specified.

**Note:** Only supported for commands that originated in Classic CLI.

**-xml**

Specifies command output in XML format. Use the **-o** (override switch) when specifying **-xml** on commands that require confirmation. Otherwise, the XML output will contain your confirmation string.

*CMD*

One of a set of commands used with the **naviseccli** command.

## CMD switches

The *CMD* switches are described on the pages that follow.

## Spaces in arguments

Normally, each argument to a CLI command consists of numbers, letters, and other valid printable characters for the operating system on which the CLI is running.

If a filename or other argument includes a space, you must enclose the entire string that includes the name in quotation marks ("xx xx").

# About Classic CLI

**Note:** Starting with the 02.26 and 03.26 release of FLARE, Classic CLI can be disabled via the Navisphere UI or a Secure CLI command. Only Secure CLI supports new features. SecureCLI is more secure than Classic CLI and is the preferred interface. Other than commands issued to host agents, the R26 version of SecureCLI includes all the same commands as Classic CLI.

If you plan to use Classic CLI commands, you must make sure that Classic CLI is enabled using the Navisphere Manager UI or SecureCLI command. If you have an AX4-5 series system, you cannot enable Classic CLI. For a list of commands that SecureCLI does not support, refer to Appendix B in the *EMC Navisphere Command Line Interface* manual (P/N 300-003-628).

Classic CLI commands run in a command window. Each command consists of the **navicli** command (and switches) together with another subcommand (and its switches). The **navicli** command does nothing when executed by itself. However, when you use it in combination with the other commands described in this manual, you can configure and manage any storage system supported by Navisphere CLI version 6.X.

If a CLI command fails and the CLI does not generate its own error message, it displays an error message generated by the host agent or SP agent. The CLI generates errors about command line syntax for commands and options and their values. CLI error codes appear in Appendix A.

**Note:** For a step-by-step overview and examples on setting up and using snapshots with admsnap and the CLI, refer to Chapter 6.

For information on the reserved LUN pool commands, refer to the latest revision of the *EMC Navisphere Command Line Interface (CLI) Reference*.

For information on the reserved LUN pool with SnapView, refer to "Reserved LUN pool with SnapView" on page 129.

# navicli

**Sends status or configuration requests to the storage system via command line.**

## Description

The CLI **navicli** command sends storage-system management and configuration requests to a Navisphere agent running on a local or remote server. The agent then sends the request to the specified storage system. Note that snapshot CLI differs from the clone CLI (refer to Chapter 2).

## Syntax

The **navicli** command is used as follows:

**navicli -help**
or
**navicli** [**-f** *filename*]**-h** *hostname*] [ **-help**] [**-m**] [**-np**] [**-p**] [**-t** *timeout*] [**-v** | **-q**] *CMD [optional-command-switches]*

## Prerequisites

Before you can use Navisphere SnapView CLI commands to configure a storage system (as opposed to receive storage-system configuration or status information), you must be a privileged user in the SP agent configuration file on each SP in the storage system.

## Options

**-help**

Displays the help screen and does not start the **navicli** process.

**-f** ***filename***

Specifies that data be stored in a file.

**-h** *hostname*

Specifies the server name of the storage-system SP or server. For a CX4 series, CX3 series, or CX series storage system, each SP is a server, therefore the server name is the IP address or network name of the destination SP. The environment variable **RaidAgentHost** is the default for the server if the **-h** switch is not present. The **-h** switch overrides the **RaidAgentHost** setting. If **RaidAgentHost** is not set and you omit the **-h** switch, the CLI uses the local server name.

**-m**

Suppresses output except for values. This option is most useful when used as part of a script.

**-np**

Suppresses polling on initial contact with the agent. This switch significantly increases performance when dealing with large or multiple storage systems. The agent automatically polls unless this switch is specified.

---

**Note:** When the **-np** switch is set, get commands may return stale data and set commands may erase previously changed settings. Use caution when the -np switch is set.

---

**-p**

Parses the entered *CMD* without making a network call to the API. If the string does not parse correctly, an error message prints to **stderr**; otherwise a message verifying that the string parsed correctly prints to **stdout**.

**-t** *timeout*

Sets the time-out value. *timeout* specifies the value in seconds. The default time-out value is 240 seconds unless the **RaidAgentTimeout** environment variable (described later) is set differently.The time-out value includes the time allowed for the entire command to execute (including CLI, agent, and storage-system software execution time). The time-out applies to all commands except **firmware**.

**-q**

Suppresses error messages. This switch is useful when included as part of a script.

**-v**

Enables verbose error descriptions. This is the default unless **-q** is specified.

*CMD*

One of a set of commands used with the **navicli** command to configure and manage a storage system.

**CMD switches** The *CMD* switches are described on the pages that follow.

### Spaces in arguments

Normally, each argument to a CLI command consists of numbers, letters, and other valid printable characters for the operating system on which the CLI is running. For example:

**navicli -h server1 convertEMlog -pathname G:\logs\c4.log -20**

If a filename or other argument (such as a storage group name) includes a space, you must enclose the entire string that includes the name in quotation marks (**"xx xx"**). For example, if the filename in the above were **c 4.log**, the valid command line would be:

**navicli -h server1 convertEMlog -pathname "G:\logs\c 4.log" -20**

### RaidAgent environment variables

The environment variables **RaidAgentHost**, **RaidAgentDevice**, and **RaidAgentTimeout** contain default values for the server, the device, and the time-out values, respectively, if you omit the **-h**, **-d**, and **-t** switches from the **navicli** command. Note that **RaidAgentDevice** and the **-d** switch are ignored on CX4 series, CX3 series, or CX series storage systems. If you do not set **RaidAgentTimeout** and omit the **-t** switch, the default time-out value is 240 seconds.

### Examples

The following examples show sample environment variable settings.

---

**Note:** The commands in the following examples must be executed from the root directory.

---

**set RaidAgentDevice \\.\A-9162244**
**set RaidAgentHost server1** (or for CX4 series, CX3 series, or CX series, **ss1_spa** )
**set RaidAgentTimeout 45**

### Error codes

The **navicli** command might return any of a numeric sequence of errors explained in Appendix A.

# Setting up SnapView to use snapshots

This section describes how to set up SnapView to use snapshots.

## Prerequisites for setting up snapshots

Before you can set up and use snapshots, the following must be true:

- **Source LUNs must be bound**. For a client or production server to access a source LUN, you must assign the source LUN to a storage group and connect the storage group to the production server. To do this, you must enable data access control on the storage system.
- **For VMware ESX Servers, verify that the source LUN is presented to the virtual machine** (guest operating system running on the virtual machine). For information on how to present a LUN to the virtual machine, refer to the VMware documentation that shipped with your ESX Server.
- **For configuration restrictions when using VMFS volumes**, go to the E-Lab Interoperability Navigator on EMC Powerlink and under the **PDFs and Guides** tab, open the **VMware ESX server.pdf** file.
- **For a secondary server to access the snapshot, you must assign the snapshot to a storage group**. The storage group must be connected to the secondary server that will activate the snapshot. You must assign the snapshot to a storage group other than the storage group that holds the source LUN. EMC supports placing a snapshot in the same storage group as its source LUN only if you use Replication Manager or Replication Manager/SE to put the snapshot in the storage group. This software provides same host access to the snapshot and the source LUN. For information on using these software products, refer to the documentation for the product.

  If you have a VMware ESX Server, the snapshot and source LUNs must be accessed by different virtual machines, unless the virtual machine is running one of the software programs that support same host access.

  **Note:** VMware ESX Servers *must* activate the snapshot before adding it to a storage group.

- **Reserved LUNs must be bound and added to the reserved LUN pool before starting a SnapView session.** You must add a reserved LUN to the reserved LUN pool for each source LUN's you will replicate. To configure the reserved LUN pool, refer to the chapter on the reserved LUN pool in the latest revision of the *EMC Navisphere Command Line Interface (CLI) Reference*.

## Overview of setting up SnapView to use snapshots

The following is a checklist for setting up SnapView snapshots.

**Note:** You must complete the prerequisites for setting up snapshots, as listed on page 127, before you can perform any of the following procedures.

- ❑ Start a SnapView session, refer to page 164.

  **Note:** You can create a snapshot before starting a session but the snapshot has no use until you start a session on it. A secondary server can then activate the snapshot to the session.

- ❑ Create a snapshot, refer to page 139.
- ❑ **If you do *not* have a VMware ESX Server** - Add the snapshot to a storage group connected to the server that will access the snapshot. Refer to the *EMC Navisphere Command Line Interface (CLI) Reference*.

  **If you have a VMware ESX Server** - Activate the snapshot, refer to page 133. After you have activated the snapshot, do the following:

  a. Add the snapshot to a storage group connected to the ESX Server that will access the snapshot. Refer to the *EMC Navisphere Command Line Interface (CLI) Reference*.

  b. Rescan the bus at the ESX Server level.

  c. If a virtual machine is already running, power off the virtual machine and use the Service Console of the ESX Server to assign the snapshot to the virtual machine.

     If a virtual machine is *not* running, create a virtual machine on the ESX Server and assign the snapshot to the virtual machine.

d. Power on the virtual machine. If the snapshot is already presented to the virtual machine, rescan the bus at the virtual machine level. For virtual machines running Windows, you can use the **admsnap activate** command to rescan the bus.

## Reserved LUN pool with SnapView

The reserved LUN pool consists of one or more private LUNs and works with SnapView sessions and snapshots. The reserved LUN pool stores the original source LUN data chunks that have been modified since the start of the session. For any one session, the contents of a reserved LUN(s) and any unchanged source LUN(s) blocks compose the snapshot.

Server writes made to an activated snapshot are also stored on a reserved LUN in the global reserved LUN pool. When you deactivate the snapshot, the reserved LUN space is freed and all snapshot writes are destroyed.

Before starting a session, the reserved LUN pool must contain at least one LUN for each source LUN that will be starting a session. You can add any LUNs that are available to the reserved LUN pool. Each storage system manages its own LUN pool and assigns a separate reserved LUN (or multiple LUNs) to each SnapView source LUN. Multiple sessions of a single source LUN will share the same reserved LUN or LUNs.

If the reserved LUN fills up and there are no available reserved LUNs in the reserved LUN pool, the software automatically terminates the session that is trying to allocate reserved LUN space, logs an error, releases the reserved LUN(s) used by this session, and returns them to the reserved LUN pool. The software also destroys all copy-on-first-write data stored in the reserved LUN pool for that session. At this point, the snapshot becomes inactive and any server that has mounted volumes on the snapshot will receive I/O errors and lose access.

If you have multiple sessions of a single source LUN and the reserved LUN fills up, when the production server modifies a chunk on the source LUN, resulting in a copy-on-first-write, every session that has the same chunk will be terminated if no additional LUNs are available in the reserved LUN pool. Other sessions that did not have this chunk will continue to run and use the reserved LUN space that the terminated sessions were using.

**Note:** SnapView, incremental SAN Copy, and MirrorView/Asynchronous share the LUN resources of the reserved LUN pool. For example, if you are running an incremental SAN Copy session on one LUN and a SnapView session on another LUN, the reserved LUN pool must contain at least two LUNs - one for each source LUN.

If both sessions are running on the same source LUN, the sessions will share a reserved LUN.

### Configuring the reserved LUN pool

You must configure the reserved LUN pool before you start a SnapView session. Use Navisphere CLI to configure the reserved LUN pool (refer to the latest revision of *EMC Command Line Interface (CLI) Reference)*. The CLI Reference also explains how you can manually estimate a suitable LUN pool size.

## SnapView command functions

The snapview functions, listed alphabetically, are:

- **snapview -activatesnapshot**
- **snapview -chgrollback**
- **snapview -chgsnapshot**
- **snapview -createsnapshot**
- **snapview -deactivatesnapshot**
- **snapview -destroysnapshot**
- **snapview -listrollback**
- **snapview -listsnapshots**
- **snapview -listsessions**
- **snapview -listsnapableluns**
- **snapview -lunpool**
- **snapview -preparesnapshot**
- **snapview -rmsnapshot**
- **snapview -startrollback**
- **snapview -startsession**
- **snapview -stopsession**

**Note:** For information on which commands are supported with AX4-5 series or AX series storage systems, refer to "SnapView CLI commands supported for AX4-5 Series or AX Series storage systems" on page 28.

In the following table, the SnapView snapshot CLI functions are listed alphabetically according to essential and optional functions. For information on which commands are supported on AX4-5 series or AX series storage systems, refer to Table 2 on page 1-28.

**Table 9 naviseccli or navicli SnapView command functions**

| Essential functions (alphabetically) | |
|---|---|
| **snapview -activatesnapshot** or **admsnap -activate** | Associates a snapshot LUN with a session, required for multiple sessions on a LUN(s). The **admsnap activate** command returns a drive letter or device name of the snapshot. |
| **snapview -createsnapshot** or **snapview -preparesnapshot** | Creates a snapshot LUN. The prepare snapshot also allocates the required disk resources. The **-preparesnapshot** switch is supported with Navisphere Express *only*. |
| **snapview -startrollback** | Starts a rollback on the selected SnapView session. |
| **snapview -startsession** | Starts a single SnapView session. |
| **snapview -stopsession** | Stops a SnapView session. |
| **Optional status functions (alphabetically)** | |
| **snapview -listrollback** | Displays rollback information such as the rollback status and performance statistics. |
| **snapview -listsessions** | Displays information on all active SnapView sessions. |
| **snapview -listsnapshots** | Displays information on snapshot LUNs. |
| **snapview -listsnapableluns** | Displays the unique IDs of all potential snapshot source LUNs. |
| **snapview -lunpool** | Lists all reserved LUNs that are currently being used with SnapView. |
| **Optional reconfiguration functions (alphabetically)** | |
| **snapview -chgrollback** | Changes the rollback rate. |
| **snapview -chgsnapshot** | Renames a snapshot LUN. |
| **snapview -deactivatesnapshot** | Removes the association of a session with a snapshot LUN. |
| **snapview -rmsnapshot** or **snapview -destroysnapshot** | Removes (deletes) a snapshot LUN. The **-destroysnapshot** switch is supported with Navisphere Express *only*. |

# snapview -activatesnapshot

**Associates a snapshot LUN with a session.**

## Description

The **naviseccli** or **navicli snapview** command with **-activatesnapshot** maps a snapshot LUN to a SnapView session. When you activate a snapshot to a session, the session becomes accessible on the secondary server's storage system.

**Note:** You must issue this command to the SP that owns the source LUN(s); otherwise, the command will fail.

Once a secondary server activates a snapshot to a session, this server can write to the activated snapshot. The software stores all writes made to the snapshot in the reserved LUN pool. If the secondary server deactivates the snapshot from the session, the software destroys all writes made to the session.

**Note:** The **snapview -activatesnapshot** command serves a similar function to the **admsnap activate** command, but the **snapview -activatesnapshot** command does not return snapshot device information or allow you to activate all snapshots using one command.

## Syntax

**snapview -activatesnapshot** is used with **naviseccli** (described on page 118) or **navicli** (described on page 124) as follows:

**snapview -activatesnapshot** *sessionname* **-snapshotid** *uid*
or
**snapview -activatesnapshot** *sessionname* **-snapshotname** *name*

## Prerequisites

For **naviseccli**, you must have a user account on the storage system on which you want to execute the command. For **navicli**, you must have a username/server entry in the Navisphere SP Agent configuration file.

A session must already exist (that is, you must start one using **snapview -startsession**) and a snapshot LUN must exist (**snapview -createsnapshot**) before this command will work.

If you have a VMware ESX Server, you must activate the snapshot before you add it to a storage group.

**Options**

**-activatesnapshot** *sessionname*

Specifies session name (that you assigned when the session was started with **-startsession**).

**-snapshotid** *uid*

Specifies the 32-byte unique ID of the snapshot (assigned by the system when the copy was created with **-createsnapshot**).

**-snapshotname** *name*

Specifies the name of the snapshot (that you optionally assigned when the copy was created with **-createsnapshot**).

**Output**

If the version of software running on the SP does not support this command, a `Command is not supported` error message is printed to **stderr**.

**Examples**

**naviseccli -h ss1_spa snapview -startsession 2pmsession -snapshotname lun20copy**

**naviseccli -h ss1_spa snapview -activatesnapshot 2pmsession -snapshotname lun20copy**

For SP **ss1_spa**, these commands start a session named **2pmsession** with snapshot **LUN lun20copy**, and then associate the session and snapshot. The copy then becomes accessible on the secondary server's storage system.

---

**Note:** To use this command with **navicli**, replace **naviseccli** with **navicli**.

---

# snapview -chgrollback

**Changes the rollback rate.**

## Description

The **naviseccli** or **navicli snapview** command with **-chgrollback** changes the rate at which the data copies back to the source LUN(s).

**Note:** If a session spans across both SPs or if the session is a consistent session, you can issue this command from either SP; otherwise, you must issue this command to the SP that owns the source LUN(s) or the command will fail.

## Syntax

**snapview -chgrollback** is used with **naviseccli** (described on page 118) or **navicli** (described on page 124) as follows:

**snapview -chgrollback** *sessionname* **-rate** [**high** | **medium** | **low**]

## Prerequisites

For **naviseccli**, you must have a user account on the storage system on which you want to execute the command. For **navicli**, you must have a username/server entry in the Navisphere SP Agent configuration file.

## Options

**-chgrollback** *sessionname*

Specifies the name for the SnapView session that is rolling back.

**-rate high** | **medium** | **low**

Specifies the rate at which the software copies the data back to the source LUN(s). Options are **high**, **medium**, or **low**. The default is **medium**.

## Output

If the specified session is not rolling back, the software returns an error message.

If the version of software running on the SP does not support this command, a `Command is not supported` error message is printed to **stderr**.

## Example

**naviseccli -h ss1_spa snapview -chgrollback rollbackTest -rate medium**

For SP **ss1_spa**, this command changes the rollback rate to **medium** for the session named **rollbackTest**. There is no output for this command. You can verify the new rollback rate by issuing the **-listrollback -rate** command. For additional information about the **-listrollback** command, refer to page 144.

**Note:** To use this command with **navicli**, replace **naviseccli** with **navicli**.

# snapview -chgsnapshot

**Renames a snapshot.** This switch is not supported on AX4-5 series storage systems with Navisphere Express.

## Description

The **naviseccli** or **navicli snapview** command with **-chgsnapshot** changes the name of a snapshot.

**Note:** You can issue this command from either SP.

## Syntax

**snapview -chgsnapshot** is used with **naviseccli** (described on page 118) or **navicli** (described on page 124) as follows:

**snapview -chgsnapshot -snapshotid** *uid* **-snapshotnewname** *newname*
or
**snapview -chgsnapshot -snapshotname** *name* **-snapshotnewname** *newname*

## Prerequisites

For **naviseccli**, you must have a user account on the storage system on which you want to execute the command. For **navicli**, you must have a username/server entry in the Navisphere SP Agent configuration file.

## Options

**-snapshotid** *uid*

Specifies the 32-byte LUN unique ID (WWN) of the snapshot, assigned when the snapshot was created. For any letters in the ID, you must specify the same case as the software assigned to the image when the snapshot was created (use the **snapview -listsnapshots** function).

**-snapshotname** *name*

Specifies the existing name for the snapshot.

**-snapshotnewname** *newname*

Specifies the new name for the snapshot.

## Output

If the version of software running on the SP does not support this command, a `Command is not supported` error message is printed to **stderr**.

## Example

**naviseccli -h ss1_spa snapview -chgsnapshot -snapshotname snap11am -snapshotnewname snap3pm**

For SP **ss1_spa**, this command renames the snapshot named **snap11am** to **snap3pm**.

**Note:** To use this command with **navicli**, replace **naviseccli** with **navicli**.

# snapview -createsnapshot

**Creates a snapshot.** This switch is not supported on AX4-5 series storage systems with Navisphere Express.

## Description

The **naviseccli** or **navicli snapview** command with **-createsnapshot** function creates a snapshot. A snapshot is a virtual LUN and when activated (**-activatesnapshot**), it allows a secondary server to view a SnapView session. An active snapshot is a composite of a source LUN and reserved LUN data that lasts until you destroy the snapshot. You can create a snapshot before or after you start a session; however, the snapshot has no use until a secondary server activates it to a session.

**Note:** When you create a snapshot, SnapView automatically allocates and retains additional memory. This additional memory counts against the total replication software memory budget.

If the storage system loses power while the SnapView session is running, both the session and snapshot would survive the storage system power failure, since all sessions run in persistence mode.

You must issue this command to the SP that owns the source LUN(s); otherwise, the command will fail.

**Note:** Unless you have additional software that supports same host access, you must assign the snapshot to a storage group other than the storage group that holds the source LUN(s). You also must assign multiple snapshots, of the same source LUN(s), to different storage groups. For information on software that supports same host access, refer to the "Prerequisites for setting up snapshots" on page 127.

## Syntax

**snapview -createsnapshot** is used with **naviseccli (**described on page 118) or **navicli** (described on page 124) as follows:

**snapview -createsnapshot** *lun* [**-snapshotname** *name*]

## Prerequisites

For **naviseccli**, you must have a user account on the storage system on which you want to execute the command. For **navicli**, you must have a username/server entry in the Navisphere SP Agent configuration file.

## Options

**-createsnapshot** *lun*

Specifies the LUN ID of the source LUN(s).

**-snapshotname** *name*

Lets you assign a meaningful name to the snapshot; for example **2pmsnap**. You can use this name during the session to manage the snapshot. The name can include as many as 32 characters and spaces are allowed.

**Output**

If the version of software running on the SP does not support this command, a `Command is not supported` error message is printed to **stderr**.

**Example**

**naviseccli -h ss1_spa snapview -createsnapshot 20 -snapshotname lun20snap**

For SP **ss1_spa**, this **-createsnapshot** command creates a snapshot for the source LUN(s) that has the LUN ID **20**. The snapshot name is **lun20snap**.

---

**Note:** To use this command with **navicli**, replace **naviseccli** with **navicli**.

---

# snapview -deactivatesnapshot

**Removes the association between a SnapView session and a snapshot LUN.**

## Description

The **naviseccli** or **navicli snapview** command with **-deactivatesnapshot** unmaps a snapshot from a SnapView session and destroys any secondary server writes made to the snapshot. The snapshot becomes inaccessible on the secondary server's storage system, while the session continues.

**Note:** You must issue this command to the SP that owns the source LUN(s); otherwise, the command will fail.

## Syntax

**snapview -deactivatesnapshot** is used with **naviseccli (**described on page 118) or **navicli** (described on page 124) as follows:

**snapview -deactivatesnapshot -snapshotid** *uid* [**-o**]
or
**snapview -deactivatesnapshot -snapshotname** *name* [**-o**]

## Prerequisites

For **naviseccli**, you must have a user account on the storage system on which you want to execute the command. For **navicli**, you must have a username/server entry in the Navisphere SP Agent configuration file.

## Options

**-snapshotid** *uid*

Specifies the 32-byte unique ID of the snapshot (assigned when the copy was created with **-createsnapshot**).

**-snapshotname** *name*

Specifies the name of the snapshot (optionally assigned when the copy was created with **-createsnapshot**).

**-o**

Executes the command without prompting for confirmation.

## Output

If the version of software running on the SP does not support this command, a `Command is not supported` error message is printed to **stderr**.

**Example** **naviseccli -h ss1_spa snapview -deactivatesnapshot -snapshotname lun20copy**

For SP **ss1_spa**, this command removes the existing association between the snapshot **lun20copy** and its session. The snapshot then becomes inaccessible on the secondary server's storage system.

---

**Note:** To use this command with **navicli**, replace **naviseccli** with **navicli**.

---

# snapview -destroysnapshot

**Frees and unallocates the snapshot disk resources for an AX4-5 series system with Navisphere Express.**

## Description

The **naviseccli snapview** command with the **-destroysnapshot** frees and unallocates the snapshot disk resources that were allocated when you prepared the snapshot. This command also removes the snapshot from the storage group and destroys the snapshot.

Before you destroy the snapshot, verify that you have:

- Removed access to the snapshot from the secondary server. If you omit this step, you will have to reboot the secondary server.
- Stopped the snapshot from the source server. If you omit this step, the snapshot will be stopped prior to the destroy.

**Note:** In Navisphere CLI, the point-in-time copy you start and stop, is referred to as a session.

## Syntax

**snapview -destroysnapshot** is used with **naviseccli** (described on page 118) as follows:

**snapview -destroysnapshot -hostname** *name* **[-o]**

## Prerequisites

For **naviseccli**, you must have a user account on the storage system on which you want to execute the command.

## Options

**-hostname** *name*

Specifies the server name assigned to the snapshot. This server is referred to as the secondary server.

**-o**

Destroys the required snapshot disk resources and snapshot without prompting for confirmation.

## Example

**naviseccli snapview -destroysnapshot -hostname mysecondaryserver**

## Output

If the version of software running on the SP does not support this command, a `Command is not supported` error message is printed to **stderr**.

# snapview -listrollback

**Displays rollback information, such as the rollback status and performance statistics.**

## Description

The **naviseccli** or **navicli snapview** command with **-listrollback** and no other switches lists all rollback information. You can obtain more specific information with the function switches described below.

**Note:** If a session spans across both SPs or if the session is a consistent session, you can issue this command from either SP; otherwise, you must issue this command to the SP that owns the source LUN(s) or the command will fail.

## Syntax

**snapview -listrollback** is used with **naviseccli** (described on page 118) or **navicli** (described on page 124) as follows:

**snapview -listrollback** [**-name** *sessionname*] [**-rate** [**-high** | **-medium** | **-low**]] [**-complete**] [**-incomplete**] [**-reads**] [**-writes**] [**-spanreads**] [**-spanwrites**] [**-blocksread**] [**-blockswritten**] [**-rollbackpercent**]

**Note:** If a session spans across both SPs, the function switches that provide statistical information will specify only the statistics for the SP in the **navicli** command.

## Prerequisites

For **naviseccli**, you must have a user account on the storage system on which you want to execute the command. For **navicli**, you must have a username/server entry in the Navisphere SP Agent configuration file.

## Options

**-name** *sessionname*

Specifies the name for the SnapView session that is rolling back. If you do not specify a session name, the software returns the names for all sessions that are rolling back on that SP.

**-rate**

Specifies the rate at which the software copies the data back to the source LUN(s). Values are **high**, **medium**, or **low**.

**-complete**

Displays the number of chunks that have been copied to the source LUN(s). This includes the number of chunks flushed and the number of chunks invalidated.

The number of chunks flushed is the number of reserved LUN chunks that have been copied from the session that is rolling back to the source LUN(s).

The number of chunks invalidated is the number of reserved LUN chunks that will not be copied to the source LUN(s). Server writes made to the source LUN(s) while the rollback is in progress will take precedence over the data chunks being rolled back.

**-incomplete**

Displays the number of copy-on-first-write chunks that have not been copied back to the session's source LUN(s).

**-reads**

Displays the number of server read requests made to the source LUN(s) during the rollback operation.

**-writes**

Displays the number of server write requests made to the source LUN(s) during the rollback operation.

**-spanreads**

Displays the number of server read requests made to the source LUN(s) during the rollback operation, which spanned a partial chunk.

**-spanwrites**

Displays the number of server write requests made to the source LUN(s) during the rollback operation, which spanned a partial chunk.

**-blocksread**

Displays the number of 512-byte disk blocks that were read by the source LUN's server during the rollback operation.

**-blockswritten**

Displays the number of 512-byte disk blocks that were written by the source LUN's server during the rollback operation. Server writes made to the source LUN(s) while the rollback is in progress will overwrite the data being rolled back.

**-rollbackpercent**

Displays the percentage of data chunks that have been copied to the source LUN(s).

## Output

The following is a sample output that shows the devices controlled by one agent. Actual output varies depending on the switches you use with the **-listrollback** command.

```
Name of the session:          rollbackTest
Session RollBack Rate:        low
Number of Chunks Complete:    1024
   Chunks Flushed:            1024
   Chunks Invalidated:        0
Number of Chunks Incomplete:  15293
Host Read Requests:           0
Host Write Requests:          0
Spanning Host Read Requests:  0
Spanning Host Write Requests: 0
Number of Blocks Read:        0
Number of Blocks Written:     0
Percent Rolled Back:      6.27566
```

If no session is rolling back, Navisphere CLI displays the following message:

```
No snap session on this SP is in the process of rolling
back.
```

If the version of software running on the SP does not support this command, a `Command is not supported` error message is printed to **stderr**.

## Examples

You can specify no function switches or only those that you are interested in, as illustrated in the examples below. A sample output is listed on page 146.

**naviseccli -h ss1_spa snapview -listrollback**

Without any function switches specified for SP **ss1_spa**, this command will display the statistics information for all sessions that are rolling back.

**naviseccli -h ss1_spa snapview -listrollback -name**

For SP **ss1_spa**, this command displays only the name of all sessions that are rolling back.

**naviseccli -h ss1_spa snapview -listrollback -name** *sessionName*

For SP **ss1_spa**, this command displays the statistics information for the specified session.

**naviseccli -h ss1_spa snapview -listrollback -name -rate -complete**

For SP **ss1_spa**, this command displays the rollback rate and the number of chunks completed for each session that is rolling back.

---

**Note:** To use this command with **navicli**, replace **naviseccli** with **navicli**.

---

# snapview -listsessions

**Lists session performance statistics.**

## Description

The **naviseccli** or **navicli snapview** command with **-listsessions** and no other switches lists information about all SnapView sessions. With the **-name** *name* function switch, it lists all information on the specified session.

**Note:** If a session spans across both SPs or if the session is a consistent session, you can issue this command from either SP; otherwise, you must issue this command to the SP that owns the source LUN(s) or the command will fail.

## Syntax

**snapview -listsessions** is used with **naviseccli (**described on page 118) or **navicli** (described on page 124) as follows:

**snapview -listsessions [-name** [*sessionname*] ] **[-all] [-cacheusage] [-consistent] [-cowwrites]** [**-creationtime] [-includepeer] [-luns] [-lunwrites] [-persistence] [-readfromcache] [-readslun] [-readssnapshot] [-reserved] [-sessionstate] [-snapshotsname] [-snapshotsuid] [-tlunumber] [-totalwrites] [-writeslargerthansize]**

## Prerequisites

For **naviseccli**, you must have a user account on the storage system on which you want to execute the command. For **navicli**, you must have a username/server entry in the Navisphere SP Agent configuration file.

## Options

**-name** [*sessionname*]

Specifies the name of the session to describe. If you omit *name*, the CLI displays all session names.

**-all**

Lists all the properties of a session without having to type in any other switch.

**-cacheusage**

Displays the storage system's reserved LUN pool capacity in gigabytes and percentage of or total number of chunks used in this SnapView session.

**-consistent**

Lists whether the session is running in consistent mode.

**-cowwrites**

Displays the number of write requests to the reserved LUN pool that triggered a copy on first write.

**-creationtime**

Displays the time and date that the session was created.

**-includepeer**

**Note:** This feature is not supported on AX4-5 or AX series storage systems.

Lists sessions on the peer SP and LUNs from sessions spanning both SPs.

**-luns**

Lists the source LUNs used in the sessions.

**-lunwrites**

Displays the total number of write requests from the source LUN(s).

**-persistence**

Lists whether the session is running in persistence mode. Navisphere CLI version 6.24 or later enables persistence mode as the default.

**-readfromcache**

Displays the number of chunks read from the reserved LUN pool.

**-readslun**

Displays the total number of chunks read from the source LUN(s).

**-readssnapshot**

Displays the number of chunks read from the snapshot.

**-reserved**

Lists all reserved sessions. Reserved sessions are sessions used for another application such as SAN Copy and MirrorView/Asynchronous.

**-sessionstate**

Displays the state of the session: **Stopping**, **Normal,** or **Rolling Back**.

**-snapshotsname**

Lists the names of any active snapshots in this session.

**-snapshotsuid**

Lists the 16-byte LUN unique IDs of the snapshots in this session.

**-tlunumber**

Displays the LUN number of the source LUN(s).

**-totalwrites**

Displays the sum of write requests to the source LUN(s) and to the reserved LUN pool in the SnapView session.

**-writeslargerthansize**

Displays the number of write requests larger than the chunk size.

## Output

The following is a sample output that shows the devices controlled by one agent. Actual output varies depending on the switches you use with the **-listsessions** command.

```
Name of the session: rollbackTest
Number of read requests serviced by the snapview reserved
LUN pool: 65264
Total number of read requests on the snapview logical
unit:  0
Number of reads from the TLU: 4294902032
Number of writes requests in the session: 65264
Number of writes requests to reserved LUN pool that
triggered a COW: 0
Total number of writes requests on the snapview target
logical unit: 65264
Number of writes requests larger than the chunk size: 0
LUN Pool Capacity in GB: 6
Session Usage for This SP: 33.257
List of Target Logical Units: LUN 8 LUN 9
snap Logical Units UID
snap Logical Units Name
Session in persistence:       YES
Session creation time:        03/21/03 06:56:28
Session state:                Normal
```

If the version of FLARE software running on the SP does not support this command, a `Command is not supported` error message is printed to **stderr**.

**Example** **naviseccli -h ss1_spa snapview -listsessions**

For SP **ss1_spa**, this command lists all sessions. A sample output follows.

**Note:** To use this command with **navicli**, replace **naviseccli** with **navicli**.

# snapview -listsnapshots

**Displays information on snapshot copies.**

## Description

The **naviseccli** or **navicli snapview** command with **-listsnapshots** and no other switches lists all information on all snapshot copies.

With **-snapshotname** *name* or **-snapshotid** *uid*, it lists all information about one snapshot. The command works only after a **snapview -createsnapshot** command has created one or more snapshot copies.

**Note:** You can issue this command from either SP.

## Syntax

**snapview -listsnapshots** is used with **naviseccli (**described on page 118) or **navicli** (described on page 124) as follows:

**snapview -listsnapshots [-lun] [-name] [-reserved] [-snapshotname** [*name*]] **[-snapshotid** [*uid*]] **[-state]**

## Prerequisites

For **naviseccli**, you must have a user account on the storage system on which you want to execute the command. For **navicli**, you must have a username/server entry in the Navisphere SP Agent configuration file.

## Options

**-lun**

Lists source LUNs.

**-name**

Lists the SnapView session names.

**-reserved**

Lists all reserved snapshots. Reserved snapshots are snapshots used for another application such as SAN Copy and MirrorView/Asynchronous.

**-snapshotname** [*name*]

Specifies the name of the snapshot assigned when the snapshot was created or the session started. If you omit *name*, it displays all snapshot names.

**-snapshotid** [*uid*]

Specifies the unique ID of the snapshot. If you omit *uid*, it displays the unique IDs of all copies.

**-state**

Lists the state of the copy: **Active** or **Inactive**.

## Output

The following is a sample output that shows the devices controlled by one agent. Actual output varies depending on the switches you use with the **-listsnapshots** command.

```
SnapView logical unit name: snapOnLun4
SnapView logical unit ID:
60:06:01:67:51:09:00:00:CD:2E:76:86:FE:5D:D7:11
Target Logical Unit:        4
State:                      Active
Session Name:               sessionAtNoonOnLun4
```

If the version of software running on the SP does not support this command, a `Command is not supported` error message is printed to **stderr**.

## Example

**naviseccli -h ss1_spa snapview -listsnapshots**

For SP **ss1_spa1**, this command lists all information on snapshot copies. A sample output follows.

---

**Note:** To use this command with **navicli**, replace **naviseccli** with **navicli**.

---

# snapview -listsnapableluns

**Lists all source LUNs that you can create a snapshot on.**

## Description

The **naviseccli** or **navicli snapview** command with **-listsnapableluns** lists all source LUNs that you can snap.

**Note:** You can issue this command from either SP.

## Syntax

**snapview -listsnapableluns** is used with **naviseccli (**described on page 118) or **navicli** (described on page 124) as follows:

**snapview -listsnapableluns**

## Prerequisites

For **naviseccli**, you must have a user account on the storage system on which you want to execute the command. For **navicli**, you must have a username/server entry in the Navisphere SP Agent configuration file.

## Output

The following is a sample output that shows the devices controlled by one agent. Actual output varies depending on the switches you use with the **-listsessions** command.

```
Name LUN 1
Name LUN 2
Name LUN 3
Name LUN 4
Name LUN 5
Name LUN 8
Name LUN 9
Name LUN 10
Name LUN 11
Name metaLun67
```

If the software running in the SP does not support the command, the CLI prints a `Command is not supported` error message to **stderr**.

## Example

**naviseccli -h ss1_spa snapview -listsnapableluns**

For SP **ss1_spa**, this command lists all LUNs ready for snapshot copying. A sample output follows.

**Note:** To use this command with **navicli**, replace **naviseccli** with **navicli**.

# snapview -lunpool

**Lists all reserved LUNs that are currently being used with SnapView.**

## Description

The **naviseccli** or **navicli snapview** command with **-lunpool** lists all information on the reserved LUNs that SnapView is currently using on the source LUNs owned by the directed SP.

## Syntax

**snapview -lunpool** is used with **naviseccli** (described on page 118) or **navicli** (described on page 124) as follows:

**snapview -lunpool**

## Prerequisites

For **naviseccli**, you must have a user account on the storage system on which you want to execute the command. For **navicli**, you must have a username/server entry in the Navisphere SP Agent configuration file.

## Output

The following is a sample output that shows the devices controlled by one agent.

```
Target LUNs  AssociatedLUNPoolLUNs  LUNPoolLUNUsedPercent
4            20                    3.562500
5            21                    3.562500
1            16                    0.347900
0            17                    0.347900
```

If the software running in the SP does not support the command, the CLI prints a `Command is not supported` error message to **stderr**.

## Example

**naviseccli -h ss1_spa snapview -lunpool**

This command lists all reserved LUNs on the storage system being used by SnapView. A sample output follows.

**Note:** To use this command with **navicli**, replace **naviseccli** with **navicli**.

# snapview -preparesnapshot

**Allocates the required snapshot disk resources for an AX4-5 series system with Navisphere Express.**

## Description

The **naviseccli snapview** command with the **-preparesnapshot** allocates the required snapshot disk resources, creates a snapshot, and adds a snapshot to a storage group.

Depending on the size of the virtual disk, this process can take some time to complete. The point-in-time copy of the virtual disk does not exist until you start the snapshot. Once you start the snapshot, and as the source server writes and modifies data on the source virtual disk, the original data (the data being modified after the copy began) is stored in the allocated disk resources for the snapshot, which preserves the point-in-time copy.

---

**Note:** In Navisphere CLI, the point-in-time copy you start and stop, is referred to as a session.

You must issue this command to the SP that owns the source LUN(s); otherwise, the command will fail.

---

## Syntax

**snapview -preparesnapshot** is used with **naviseccli** (described on page 118) as follows:

**snapview -preparesnapshot** *source_virtualdisk* **-hostname** *name* **[-o]**

## Prerequisites

- For **naviseccli**, you must have a user account on the storage system on which you want to execute the command.
- Before you can use the snapshot feature with a storage system with Navisphere Express, you must have at least two servers connected to the storage system. You cannot access the snapshot from the same server as the virtual disk. Doing so may corrupt the data on the source.
- If virtual disks are assigned to the secondary server, and this server is running AIX, HP-UX, Solaris, Linux, or NetWare, we recommend that you use the update function of the Navisphere Server Utility before you access the snapshot. By updating server information, you can determine the current device names for this server. Navisphere Express lists the device names in the **Server Details** page.

- You must have free space on any disk pool. Depending on the size of your source virtual disk, you must have at least 20% of the virtual disk or a minimum of one 10 Gbyte disk.

  Navisphere Express will attempt to allocate 20% of the size of the virtual disk in increments of 10 Gbytes or a minimum of one 10 Gbyte disk. For example, if you prepared a snapshot for a virtual disk that was 60 Gbytes in size (20% is 12 Gbytes), the software would allocate two 10 Gbyte disk resources. If you prepare a snapshot for a virtual disk that is 1 Gbyte, the software would allocate one 10 Gbyte disk resource.

**Options** *source_virtualdisk*

Specifies the source virtual disk number for which you want to create a snapshot.

**-hostname** *name*

Specifies the server name you want to assign to the snapshot. This server is referred to as the secondary server.

**-o**

Allocates the required snapshot disk resources and creates the snapshot without prompting for confirmation.

**Example** **naviseccli snapview -preparesnapshot 11 -hostname mysecondaryserver**

**Output** If the version of software running on the SP does not support this command, a `Command is not supported` error message is printed to **stderr**.

# snapview -rmsnapshot

**Deletes a snapshot.** This switch is not supported on AX4-5 series storage systems with Navisphere Express.

## Description

The **naviseccli** or **navicli snapview** command with the **-rmsnapshot** deletes a snapshot created with the **snapview create** function.

You can specify either the snapshot unique ID or name (if you assigned a name when you created the copy).

**Note:** You must issue this command to the SP that owns the source LUN(s); otherwise, the command will fail.

## Syntax

**snapview -rmsnapshot** is used with **naviseccli** (described on page 118) or **navicli** (described on page 124) as follows:

**snapview -rmsnapshot -snapshotid** *uid* [**-o**]
or
**snapview -rmsnapshot -snapshotname** *name* [**-o**]

## Prerequisites

For **naviseccli**, you must have a user account on the storage system on which you want to execute the command.

For **navicli**, you must have a username/server entry in the Navisphere SP Agent configuration file.

## Options

**-snapshotid** *uid*

Specifies the 32-byte unique ID of the snapshot (assigned when the copy was created with **-createsnapshot**).

**-snapshotname** *name*

Specifies the name of the snapshot (optionally assigned when the copy was created with **-createsnapshot**).

**-o**

Removes the copy without prompting for confirmation.

## Output

If the version of software running on the SP does not support this command, a `Command is not supported` error message is printed to **stderr**.

**Example** **naviseccli -h ss1_spa snapview -rmsnapshot -snapshotname 2pmsnap**

For SP **ss1_spa**, this command starts removing the snapshot with the name **2pmsnap**. The software asks for confirmation:

```
Do you really want to remove the snapshot (y/n)[N]?
```

**y**

---

**Note:** To use this command with **navicli**, replace **naviseccli** with **navicli**.

---

# snapview -startrollback

**Starts a rollback on the selected SnapView session.**

## Description

The **naviseccli** or **navicli snapview** command with **-startrollback** starts a rollback on the selected SnapView session. Rollback lets you restore the point-in-time copy of a session to the source LUN(s).

**IMPORTANT**

**Windows operating systems only - To prevent data corruption during the rollback operation, you should disable the indexing service and recycle bin on the source LUN(s) of the session you will roll back.**

When you confirm the start of a rollback operation, the source LUN can instantly access the session's point-in-time data, while background copying continues (until all the session's point-in-time blocks are copied to the source LUN). You cannot start a new session on any source LUN that is participating in the rollback operation until all its background data copying has completed. The rollback operation itself does not count against the eight-session limit per source LUN.

**Note:** If a session spans across both SPs or if the session is a consistent session, you can issue this command from either SP; otherwise, you must issue this command to the SP that owns the source LUN(s) or the command will fail.

**Rolling back with snapshots**

Each source LUN belonging to a session that is rolling back can have an activated or deactivated snapshot. If the session has an activated snapshot, the software copies any server writes made to this snapshot to the source LUN. However, you must unmount the snapshot *before* you start the rollback operation. When the rollback operation completes, including background copying, you can remount the snapshot.

If you deactivate the snapshot prior to starting the rollback operation, any server writes made to the snapshot are lost.

**Allocating reserved LUN pool space**

The rollback operation itself does not use reserved LUN pool space. However, the rollback may cause copy-on-first-write operations for

sessions you started after the session that is rolling back. This in turn will consume more reserved LUN pool space. These copy-on-first-write operations occur because the original point-in-time data of those sessions is being modified by the rollback operation. Additionally, since you can resume I/O to the source LUN(s) or start another session while background copying continues, you may need additional reserved LUN space.

If you run out of reserved LUNs in the SP's LUN pool while the rollback operation is in progress (during the background copying), the software will terminate the session that is rolling back after it completes all background copying. The software will also terminate any other session that tries to allocate additional reserved LUN pool space.

You *cannot* roll back a SnapView session if the session's source LUN(s) is:

- an unfractured clone. You must fracture the clone before rolling back the session.
- a fractured clone that is unavailable for I/O. For instance, if the clone was fractured while in an out-of-sync state.
- a clone group's source LUN that has any unfractured clone LUNs or clones that are in the process of reverse synchronizing. All clone LUNs for this source LUN must be fractured in a synchronized or consistent state before you can roll back a session to its source LUN.
- a primary image that has one or more unfractured secondary images. You must fracture the secondary image(s) before rolling back the session.
- a secondary mirror image. You must promote the secondary image and then fracture it before rolling back the session.

### Syntax

**snapview -startrollback** is used with **naviseccli** (described on page 118) or **navicli** (described on page 124) as follows:

**snapview -startrollback** *sessionname* **[-rate high | medium | low] [-o]**

**Note:** Once you start a rollback, you *cannot* stop it or the session that is rolling back.

### Prerequisites

For **naviseccli**, you must have a user account on the storage system on which you want to execute the command. For **navicli**, you must

have a username/server entry in the Navisphere SP Agent configuration file.

Before starting a rollback on a SnapView session, verify the following:

- ❑ The session is running in persistence mode. Navisphere CLI version 6.24 or later enables persistence mode as the default.
- ❑ The session is not in the process of being rolled back.
- ❑ The source LUN(s) is not participating in another rollback.
- ❑ If an activated snapshot is mapped to this session, unmount the snapshot.

**Options**

**-startrollback** *sessionname*

Specifies the name for the SnapView session you want to roll back.

**-rate high | medium | low**

Specifies the rate at which the software copies the data back to the source LUN(s). Options are **high**, **medium**, or **low**. The default is **medium**. You can change this rate while the rollback operation is in progress (see **-chgrollback** on page 135). If you do not specify the **-rate** option, the software uses the default value of **medium**.

**-o**

Executes the command without prompting for confirmation.

**Output**

If the version of software running on the SP does not support this command, a `Command is not supported` error message is printed to **stderr**.

**Example**

**naviseccli -h ss1_spa snapview -startrollback rollbackTest -rate low**

For SP **ss1_spa**, this command starts rolling back the session named **rollbackTest**. The specified rollback rate is **low**. The software asks for confirmation.

```
This session has at least one active snapshot associated
with it. Continuing this operation will restore not just
the original data but also any modifications made to the
active snapshot LUNs. If only the original data is
desired for the RollBack, then quit this operation and
deactivate the snapshot first.

Make sure that you have quiesced all host I/O to the
Session's source LUNs and to any of the associated active
```

```
Snapshot LUNs. This command will flush the Session's data
back to the Source LUNs asynchronously in the background.

NOTE: If you want a recovery session for this rollback,
you must create a persistent session NOW before beginning
the rollback. Otherwise you will not be able to recover
the current data once rollback begins.

Do you want to continue? (y/n)?
```

**Note:** To use this command with **navicli**, replace **naviseccli** with **navicli**.

# snapview -startsession

**Starts a SnapView session in persistence mode, and optionally in consistent mode.**

## Description

The **naviseccli** or **navicli snapview** command with **-startsession** starts a SnapView session on the specified source LUN(s). A SnapView session is a point-in-time copy of a source LUN. The session keeps track of how the source LUN looks at a particular point in time. You start a session using the snapshot name, snapshot unique ID (snapshot WWN), or the source LUN's number.

With Navisphere CLI version 6.24 or later, all SnapView sessions run persistence mode, which means that the session will survive SP reboot or failure, software installation, storage-system reboot or power failure, and server I/O trespassing to the peer SP. You can also run a session in consistent mode using the **-consistent** option, which will preserve the point-in-time restartable copy across a set of source LUNs.

---

**Note:** When you start a session, SnapView automatically allocates and retains additional memory. This additional memory counts against the total replication software memory budget.

---

After you start a SnapView session and as the production server writes to the source LUN(s), the software stores a copy of the original data in the reserved LUN pool in chunks. This copy is referred to as copy-on-first-write and occurs only once, which is when the server first modifies a data chunk on the source LUN(s).

---

**Note:** If a session spans across both SPs or if the session is a consistent session, you can issue this command from either SP; otherwise, you must issue this command to the SP that owns the source LUN(s) or the command will fail.

---

## Syntax

**snapview -startsession** is used with **naviseccli** (described on page 118) or **navicli** (described on page 124) as follows:

**snapview -startsession** *sessionname*
**-snapshotid** *uids* | **-snapshotname** *names* | **-lun** *luns* **[-consistent]**
**-persistence**

**Note:** With Navisphere CLI version 6.24 or later, all SnapView sessions run in persistence mode, so the **-persistence** switch is no longer required.

**Prerequisites**

For **naviseccli**, you must have a user account on the storage system on which you want to execute the command. For **navicli**, you must have a username/server entry in the Navisphere SP Agent configuration file.

Before starting a SnapView session, verify the following:

- You configured the reserved LUN pool by using the **reserved -lunpool -addlun** function.
- The source LUNs are currently not being used to start another consistent session. Once the consistent session starts on all the source LUNs, you can start another consistent session using the same source LUNs.
- If MirrorView/Asynchronous is installed, the source LUNs that you will use to start the consistent session cannot be in the process of starting an update to a group. If they are, you may receive an error when you attempt to start the consistent session. Once the mirror or group is updating or has been updated and before the next update occurs, you can start the consistent session using the same source LUNs.

**Post prequisites**

After you start a SnapView session, a secondary server can then activate (map) a snapshot to the SnapView session. The snapshot views the original source LUN data chunks that have been modified since you started the session from the reserved LUN pool and unmodified data chunks from the source LUN(s).

**Options**

**-startsession** *sessionname*

Specifies the name of the session to start. A session name can have as many as 32 characters. A name is required. If you use spaces in the name, enclose the entire name string in quotation marks in this command and in other CLI commands that access the session.

**-snapshotid** *uid*

Specifies one or more snapshot 32-byte unique IDs (WWNs). The snapshot software assigns a unique ID to the snapshot when you create it with **-createsnapshot**. For multiple IDs, separate IDs with blanks. For any letters in the ID, you must specify the same case as the software assigned to the image when the snapshot was created (use the **snapview -listsnapshots** function).

**-snapshotname** *names*

Specifies one or more snapshot names that you assigned when you created the snapshot with **-createsnapshot**. For multiple names, separate the names with commas.

**-lun** *luns*

Specifies one or more source LUN numbers. For multiple LUNs, separate the numbers with blanks.

**[-consistent]**

---

**Note:** For AX4-5 series storage systems, this switch is supported only if both the Navisphere Manager and SnapView enablers are installed. For AX series storage systems, this switch is not supported.

---

Specifies a consistent session. The session will preserve the point-in-time restartable copy across a set of source LUNs. The SnapView driver will delay any I/O requests to the set of source LUNs until the session has started on all LUNs (thus preserving the point-in-time restartable copy on the entire set of LUNs).

---

**Note:** A restartable copy is a data state having dependent write consistency and where all internal database/application control information is consistent with a database management system/application image.

---

If the session spans both SPs you need to specify only one SP and the software will start the session on both SPs. Once you start a consistent session, you cannot add another source LUN to the session. In the event of a failure, the software will not start the session on any source LUN and will display an error message.

You would use consistent mode if any of the following is true:

- You want a consistent replica across a set of LUNs.
- You do *not* want to stop I/O to the source LUN(s) before starting a session.

- You want to prevent other LUNs from being added to the session. If desired, you can start a consistent session on a single source LUN to prevent other LUNs from being added to the session.
- The set of source LUNs that you will use to start the session spans both SPs; however, the source LUNs cannot span storage systems.

The consistent feature is available on a per-session basis (not per snapshot or source LUN) and counts as one of the eight sessions per source LUN limit.

---

**Note:** While a consistent session is in the process of starting on all LUNs, you cannot stop the session (the software may stop the session immediately after the start completes if I/O is queued and there is not enough space in the reserved LUN pool), perform a rollback operation, or activate a snapshot to the session.

---

**-persistence**

---

**Note:** With Navisphere CLI version 6.24 or later, all SnapView sessions run in persistence mode, so the **-persistence** switch is no longer required.

If you have an AX series storage system and are running a CLI version prior to 6.24, you must specify the **-persistence** switch or an error message will be displayed.

---

Specifies a persistent session. The session will survive SP reboot or failure, software installation, storage-system reboot or power failure, and server I/O trespassing to the peer SP.

The persistence feature is available on a per-session basis (not per snapshot or source LUN). In the event of a failure, reserved LUNs along with the source LUNs will trespass to the other SP. Depending on your failover software, once the failed SP is running, you may need to issue a **restore** command in order to restore the proper source LUNs and reserved LUNs back to their original SP. For the appropriate **restore** command, refer to the documentation that shipped with your failover software.

---

**Note:** For information on supported failover software for the storage system you are managing, refer to the release notes for SnapView and admsnap.

---

**Output**

If the version of software running on the SP does not support this command, a `Command is not supported` error message is printed to **stderr**.

**Examples**

**naviseccli -h ss1_spa snapview -startsession 2pmsession**

The **-startsession** command starts a persistent session named **2pmsession**.

**naviseccli -h ss1_spa snapview -startsession 2pmsession -persistence**

The **-startsession** command starts a persistent session named **2pmsession**.

**naviseccli -h ss1_spa snapview -startsession 2pmsession -lun one two three -consistent**

The **-startsession** command starts a persistent and consistent session named **2pmsession** on LUNs one, two, and three. If the session spans both SPs you need to specify only one SP and the software will start the session on both SPs.

---

**Note:** To use this command with **navicli**, replace **naviseccli** with **navicli**.

---

# snapview -stopsession

**Stops a SnapView session.**

## Description

The **naviseccli** or **navicli snapview** command **-stopsession** stops the specified session and frees the reserved LUN pool space used by the session.

If the snapshots participating in the session belong to one or more storage groups and you stop the session, the servers connected to the storage groups will no longer have access to the snapshots in those storage groups. EMC recommends that you do one of the following:

- Flush the I/O on the server viewing the active snapshot before stopping the session.
- Deactivate the snapshot before stopping the session.

Stopping a session with an active snapshot makes the snapshot appear inactive.

---

**Note:** If a session spans across both SPs or if the session is a consistent session, you can issue this command from either SP; otherwise, you must issue this command to the SP that owns the source LUN(s) or the command will fail.

If you started a session on multiple source LUNs, you can select any of the source LUNs to stop the session.

---

## Syntax

**snapview -stopsession** is used with **naviseccli** (described on page 118) or **navicli** (described on page 124) as follows:

**snapview -stopsession** *sessionname* **[-o]**

## Prerequisites

For **naviseccli**, you must have a user account on the storage system on which you want to execute the command. For **navicli**, you must have a username/server entry in the Navisphere SP Agent configuration file.

## Options

**-stopsession** *sessionname*

Specifies the name of the session to stop. Someone must have assigned the name when the snapshot was created or when the session was started.

**-o**

Stops the session without asking for confirmation.

**Output**

If the version of software running on the SP does not support this command, a `Command is not supported` error message is printed to **stderr**. Other errors:

`INVALID_SESSION` Session does not exist.

**Example**

**naviseccli -h ss1_spa snapview -stopsession 2pmsession**

For SP **ss1_spa**, this command stops the session named **2pmsession**. The software asks for confirmation:

```
Stopping of the session will stop data SnapView. Do you
really want to stop the session (y/n)[n]? y
```

**Note:** To use this command with **navicli**, replace **naviseccli** with **navicli**.

## What next?

For a step-by-step overview and examples on setting up and using snapshots with admsnap and the CLI, refer to Chapter 6.

# 5

# Admsnap Commands for Snapshots

This chapter explains how to use the admsnap utility software with SnapView snapshots.

**Note:** CX4 series storage systems do not support Windows 2000 servers as an attached host.

Major topics in this chapter are:

# About admsnap software and snapshots

The admsnap utility is an executable program that you can run interactively or via a script to manage snapshots. The admsnap utility resides on the servers connected to the storage system with the SnapView driver. You must execute the following admsnap commands from one of these attached servers:

- **admsnap activate** – activates a snapshot to a session.
- **admsnap deactivate –** deactivates a snapshot from a session.
- **admsnap flush** – flushes cached data to the LUN.
- **admsnap list** – lists all current snapshots.
- **admsnap start –** starts a session.
- **admsnap stop –** stops a session.

---

**Note:** The admsnap software supports native device names and PowerPath® pseudo-device names. If you use a PowerPath pseudo-device name, you can add this device name to the **etc/fstab** file (for Linux) or to the **etc/vfstab** file (for Solaris).

Admsnap only supports device naming schemes for Veritas in Linux and HP-UX, and device naming schemes and enclosure based naming schemes in Solaris.

---

For a step-by-step overview and examples on setting up and using snapshots with admsnap and Navisphere CLI, refer to Chapter 6.

### Reserved LUN pool (snapshot cache)

With SnapView version 02.03.xxx (or higher), the snapshot cache is referred to as the reserved LUN pool. The reserved LUN pool and the snapshot cache work with SnapView in the same way. However, unlike the snapshot cache, which was used solely for SnapView, the reserved LUN pool shares its LUN resources with other applications such as SAN Copy and MirrorView/Asynchronous. Navisphere CLI snapshot cache commands are still supported; however, EMC recommends that you use the new reserved LUN pool commands. For information on the new reserved LUN pool commands, refer to the latest revision of the *EMC Navisphere Command Line Interface (CLI) Reference*.

For reserved LUN pool information that is specific to SnapView, refer to "SnapView command functions" on page 131.

## Requirements for snapshots

Before you can use admsnap for snapshots, the following must be true:

### General requirements

- A connected server must be running a suitable web browser to run Navisphere CLI (and optionally the Manager). CLI is needed to configure and maintain the reserved LUN pool.
- The storage-system must have the SnapView software enabled.
- The production and secondary servers must have both the Navisphere Host Agent and admsnap software installed. To install the host agent and admsnap, refer to the EMC CLARiiON server support products guide for your operating system.
- A reserved LUN must be available for each source LUN(s) for which you will start a session; otherwise an **admsnap start** command will fail. You can bind a reserved LUN and add it to an SP's reserved LUN pool using Navisphere CLI or Manager.
- You may have to reboot the secondary server the first time you add the snapshot to the server's storage group.

### Platform/server-specific requirements

- For HP-UX, admsnap requires a volume group entry for each path to the storage system. The entry or entries can be contained in a single volume group or multiple volume groups. On the secondary server, the volume group entry or entries must specify a snapshot LUN.
- For UNIX, if a session is unexpectedly stopped (due to SP panic or path failure), the secondary server must unmount the inactive snapshot or reboot the secondary server in order for this server to disregard the sessions point-in-time data.
- If you have a VMware ESX Server, you must activate the snapshot before you add the snapshot to a storage group. Refer to the "Setting up SnapView to use snapshots" on page 127 for additional information.

# admsnap activate

**Allows an attached server to access a snapshot.**

## Description

From the secondary server, the **activate** command scans system buses for storage-system devices and determines if any device is part of a SnapView session.

When a secondary server activates (maps) a snapshot to a session, this server can write to the snapshot. The software stores all writes made to the snapshot in the reserved LUN pool. If the secondary server deactivates the snapshot from the session, the software destroys all writes.

---

**Note:** If you failed to flush the file system buffers before starting the session, the snapshot may not be usable.

---

You can activate only one session to a snapshot. To list current sessions use the CLI command **navicli snapview -listsessions**.

---

**Note:** The Navisphere CLI **snapview -activatesnapshot** command serves a similar function to the **admsnap activate** command, but does not return snapshot device information as admsnap does or allow you to activate all snapshots using one command. You must issue the **admsnap activate** command from the secondary server (the server that will access the snapshot).

---

Depending on your operating system, the **admsnap activate** command will do the following:

| Server OS | Function description |
|---|---|
| **Windows** | Returns the snapshot drive letter. You can use this drive immediately.<br>If the activated snapshot has multiple volumes or partitions, admsnap will assign a drive letter to each volume or partition. If the volume is a dynamic drive, admsnap will activate all the physical drives but will not assign a drive letter. Additional steps will be required to bring the volume online and to assign the drive letter. If the given session is on multiple physical disks and each physical disk has its own volume, then each volume will be assigned a drive letter. If no drive letters are available, then admsnap will activate only one physical disk and an error appears through the command line.<br>**For Windows Terminal Services Edition and Remote Desktop** - If a terminal server is running on the secondary server, you must remove and disable it before using the **activate** command. If you have the terminal server enabled, it prevents the **activate** command from working properly. |
| **AIX, HP-UX, Linux,** and **Solaris** | Returns the snapshot device name. You will need to run **fsck** on this device only if it contains a file system and you did not unmount the source LUN(s) when you started the session. Then, if the source LUN(s) contains a file system, mount the file system on the secondary server using the snapshot device name to make the file system available for use. |
| **NetWare** | Returns the snapshot device name. You will then need to run the **Scan for new devices** or **scan all LUNs** command on the console. After running one of the scan commands, you can mount the volume associated with this device name to make a file system available. |

## Syntax

**admsnap activate** [**-h**] **-s** *session_name* [**-o** *object_name*] [**-t** *object_type*] | [**-d** *driveletter*] [**-r** 0 | **-r** 1] [**-c** *rescan_count*]

## Prerequisites

You must have administrative privileges on the secondary server.

## Options

[**-h**]

Lists the help output for a given operation. If you use this option on the command line, a help message will be displayed for each option you specify. Admsnap will not perform the command; it lists the help output only.

This option is valid for Windows, Solaris, HP-UX, Linux, and AIX systems.

**-s** *sessionname*

Specifies the session name that you assigned when you started the session. This is a string of up to 256 characters, and can be composed of letters, numbers or underscores.

The session name is case sensitive and must be unique on the storage system. If the session name consists of spaces, you must enter the name in double quotation marks (for example, **"June 21st Session"**).

[**-o** *object_name*]

Specifies an object name which you want to activate. You should use this option only when admsnap can determine the snapshot device list for the specified object name.

In UNIX, the objects that are generally available for this option are the raw UNIX device names and the EMC PowerPath device names. In Windows, the objects that generally available for this option are the raw Windows device names.

If you omit this switch, the software searches all devices for the appropriate device, which takes time. You cannot use this option with the **-d** option. If you use the **-t** option, the device name must be a valid name for the type of object specified with the **-t** option.

The table below lists the supported object names for your operating system.

| Operating system | Supported object name(s) |
|---|---|
| **Windows** | • Drive letter[a]<br>• Physical drive name |
| **AIX**, **HP-UX**, **Linux**, **NetWare**, and **Solaris** | • Device name<br>• File system name[b] |

a. The **admsnap activate** command does not allow the use of drive letters A:, B:, C:, or D: with the **-o** option.

b. If the object name is a file system, then admsnap will parse the file system table entry to get the device name and will *attempt* to activate the snapshot for that device.

[**-t** *object_type*]

Specifies the object type. Valid Windows object types are device, file system, volume, and drive letter. Valid UNIX object types are device, volume, and file system. If the object type name contains spaces, enter the name without spaces.

If you do not specify an object type, admsnap will attempt to determine the correct object type among the valid object types and report a failure if it cannot.

You should use the **-t** option when the argument for the **-o** option is not in a standard and recognizable form for your operating system's object or device. This means if you specify a valid argument for the **-o** option, you do not need the **-t** option for any operating system drive letter, file system, Veritas volume, or PowerPath pseudo-device names. For example, on Solaris systems, admsnap determines the **-o** argument of /**dev**/**rdsk**/**c***X***d***X***t***X***s***X* to be a Solaris native device name, so you do not need to specify the **-t** *device*.

You cannot use this option in conjunction with the **–d** option.

[**-d** *driveletter*]

Specifies the drive letter you want to assign to the snapshot device, and if available, admsnap will assign the specified drive letter to the snapshot device. If the operation resulted in more than one device being activated, then all devices are activated but only the first device is assigned the drive letter.

---

**Note:** The **admsnap activate** command does not allow the use of drive letters A:, B:, C:, or D: with the **-d** option.

---

This option is valid on Windows systems *only*. All other operating systems ignore this option. You cannot use this option in conjunction with the **-o** or **-t** options.

[**-r 0** | **-r 1**]

Specifies whether to perform a rescan of the operating system devices before performing the **activate** operation. The default rescanning behavior (when the **-r** option is not present on the command line) depends on whether you specify an object name with the **-o** option. If you do specify the object name, the default is not to perform a rescan. If you do *not* specify the object name, the default is to perform a rescan, which may take some time because it will rescan every device on the operating system.

When you use the **-r [0|1]** option, it overrides any default behavior.

This option is valid on Windows, Solaris, HP-UX, and Linux systems *only*. All other operating systems ignore this option.

[**-c** *rescan_count*]

Specifies the number of rescans for devices on the operating system. A rescan count occurs after a successful **activate** operation to bring the snapshot devices online. If you specify a count of 0, admsnap will not perform a rescan; however, the device could activate successfully on the storage system but will not be online with the server. If you omit the **-c** option, then a default value of 1 is set.

This option is valid on Windows *only*. All other operating systems ignore this option.

**Example** Windows example:

**admsnap activate -s session1 -d e:**

# admsnap deactivate

**Allows the secondary server to remove access to a snapshot and destroys any writes the secondary server made to the snapshot.**

## Description

The **admsnap deactivate** command unmaps a snapshot LUN from a SnapView session by removing access to the snapshot device from the secondary (backup) server. The deactivated snapshot goes offline and thereby becomes inaccessible to the secondary server.

Use this command on the secondary (backup) server after the session has been activated and you no longer require the snapshot device.

---

**Note:** The Navisphere CLI **snapview -deactivatesnapshot** command serves a similar function to the **admsnap deactivate** command, but does not remove the snapshot device information from the server.

You must issue the **admsnap deactivate** command from the secondary server (the server that activated it).

---

When the secondary server deactivates the snapshot from the session, admsnap destroys all writes made to the snapshot.

Depending on your operating system, the **admsnap deactivate** command will do one of the following:

| Server OS | Function description |
|---|---|
| **Windows** | Flushes all buffers and removes the drive letter assigned with admsnap activate.<br>The snapshot goes inactive and continued I/O is not possible. This command lets you avoid rebooting the secondary server between activates.<br>If the deactivated snapshot has multiple volumes or partitions, admsnap will remove a drive letter from each volume or partition. If the volume is a dynamic drive, admsnap will deactivate all the physical drives but will not remove the drive letter. Additional steps will be required to bring the volume offline and to remove the drive letter. If the given session is on multiple physical disks and each physical disk has its own volume, then the drive letter on each volume will be removed.<br>**For Windows Terminal Services Edition and Remote Desktop** - If a terminal server is running on the secondary server, you must remove and disable it before using the **deactivate** command. If you have the terminal server enabled, it prevents the **deactivate** command from working properly. |
| **AIX, HP-UX, Linux,** and **Solaris** | Removes access to the snapshot but does *not* flush buffers.<br>Before you deactivate the snapshot, you must unmount the file system by issuing the **umount** command to flush all buffers. If you are unable to unmount the file system, you can issue the admsnap **flush** command[a]. |
| **NetWare** | Removes access to the snapshot but does *not* flush buffers.<br>Before you deactivate the snapshot, you must use the **dismount** command on the volume to dismount the file system. This will flush all buffers. |

a. The **flush** command is not considered a substitute for unmounting the file system; it only complements unmounting the file system.

## Syntax

**admsnap deactivate** [**-h**] **-s** *session_name* [**-o** *object_name* [**-t** *object_type*]] [**-f**]

## Prerequisites

You must have administrative privileges on the secondary server.

## Options

[**-h**]

Lists the help output for a given operation. If you use this option on the command line, a help message is displayed for each option you specify. Admsnap will not perform the command; it lists the help output only.

This option is valid for Windows, Solaris, HP-UX, Linux, and AIX systems.

**-s** *sessionname*

Specifies the session name that you assigned when you started the session. This is a string of up to 256 characters, and can be composed of letters, numbers or underscores.

The session name is case sensitive and must be unique on the storage system. If the session name consists of spaces, you must enter the name in double quotation marks (for example, **"June 21st Session"**).

The name you specify must be a valid session name for an active SnapView session started from a production server and activated on the secondary (or backup) server that is performing the deactivate operation. All snapshots with the given session name will be deactivated.

On Windows systems, if the activated LUN with the given session name has multiple volumes and the system assigned drive letters, then admsnap will delete each drive letter, dismount the volumes, and then set the volumes offline before deactivating the LUN. If multiple LUNs have the same given session, then admsnap deletes the drive letters for all the volumes and sets the volumes offline before deactivating the LUNs.

On UNIX servers, admsnap only deactivates the devices and no other operations are performed before deactivating the device.

[**-o** *object_name*]

Specifies an object name. When you execute the **admsnap deactivate** command with a device name as an object name, admsnap attempts to deactivate the snapshot on that device with the given session name. If the session name does not exist on the device, admsnap will report an error.

If you specify an object name that contains multiple devices, admsnap will deactivate the session on all the LUNs that make up those devices.

If you use the **-t** option, the device name must be a valid name for the type of object specified with the **-t** option.

The table below lists the supported object names for your operating system.

| Operating system | Supported object name(s) |
|---|---|
| **Windows** | • Drive letter[a]<br>• Physical drive name |
| **AIX**, **HP-UX**, **Linux, Solaris,** or **NetWare** | • Device name<br>• File system name[b] |

a. The **admsnap deactivate** command does not allow the use of drive letters A:, B:, C:, or D: with the **-o** option.

b. If the object name is a file system, then admsnap will parse the file system table entry to get the device name and will *attempt* to deactivate the snapshot for that device.

[**-t** *object_type*]

Specifies the object type. This option is valid only when the **-o** option is given. Valid Windows object types are device, file system, volume, and drive letter. Valid UNIX object types are device, volume, and file system. If the object type name contains spaces, enter the name without spaces.

If you do not specify an object type, admsnap will attempt to determine the correct object type among the valid object types and report a failure if it cannot.

This switch is supported on Windows or Solaris operating systems only.

[**-f**]

If you are running admsnap version 2.0 or lower, you can use this option to force a deactivate operation on a physical device name that does not have a drive letter assigned.

If you use this option on a later version of admsnap, it will ignore the option.

**Examples**

Windows example:

**admsnap deactivate -s session1 -o e:**

Example of a fully qualified pathname to a Veritas volume named **scratch**:

**admsnap deactivate -*o* /dev/xv/dsk/scratchdg/scratch**

Example of a fully qualified pathname to a raw Veritas device name:

**admsnap deactivate -*o* /dev/vx/rdmp/c1t0d0s2**

# admsnap flush

**Flushes all data that has been written to the source LUN.**

## Description

The **admsnap flush** command flushes all data.

Use it on the production server before starting a SnapView session to ensure that all cached data has been written to disk. Also, use the **flush** command on the secondary server before issuing a **deactivate** command to ensure that all data has been written.

Depending on your operating system, the **admsnap flush** command will do the following:

| Server OS[a] | Function description |
|---|---|
| **Windows** | Writes the data from the buffers to the disk. |
| **AIX, HP-UX, Linux,** and **Solaris** | Writes the data from the buffers to the disk. We recommend that you unmount the file system by issuing the **umount** command to ensure that no additional source writes are in process. If you are unable to unmount the file system, you can issue the admsnap **flush** command[b]. |
| **NetWare** | The **flush** command is not supported. Dismount the file system by using the **dismount** command on the volume. |

a. With some operating systems, you may need to shut down the application to flush the data. Specific operating systems have different requirements.

b. Neither the **flush** command nor the **sync** command is a substitute for unmounting the file system. Both commands only complement unmounting the file system.

## Syntax

**admsnap flush** [**-h**] **-o** *object_name* [**-t** *object_type*]

## Prerequisites

You must have administrative privileges on the production server.

## Options

[**-h**]

Lists the help output for a given operation. If you use this option on the command line, a help message is displayed for each option you specify. Admsnap will not perform the command; it lists the help output only.

This option is valid for Windows, Solaris, HP-UX, Linux, and AIX systems.

**-o** *object_name*

Specifies an object name. If you use the **-t** option, the device name must be a valid name for the type of object specified with the **-t** option.

The table below lists the supported object names for your operating system.

| Operating system | Supported object name(s) |
|---|---|
| **Windows** | • Drive letter<br>• Physical drive name |
| **HP-UX**[a]<br>**Solaris** | • Device name<br>• File system name[b]<br>• Veritas volume name[c]<br>• Volume group |
| **Linux** | • Device name<br>• File system name[b]<br>• Veritas volume name[c] |
| **AIX**[a] | • Device name<br>• File system name[b] |

a. If the object name is on a Logical Volume Manager, then admsnap will parse the Logical Volume Manager (LVM) to get the device names and will *attempt* to flush those devices.

b. If the object name is a file system, then admsnap will parse the file system table entry to get the device name and will *attempt* to flush that device.

c. If the object name is a Veritas volume name, admsnap will parse the Veritas volume to determine all physical disks for the volume and will *attempt* to flush each physical disk.

[**-t** *object_type*]

Specifies the object type. This switch is supported on Windows or Solaris operating systems only. Valid Windows object types are device and drive letter. Valid Solaris object types are device, volume, and file system. If the object type name contains spaces, enter the name without spaces.

If you do not provide an object type, admsnap will attempt to determine the correct object type from the supported types and report a failure if it cannot.

You should use the **-t** option when the argument for the **-o** option is not in a standard and recognizable form for your operating system's object or device. This means if you specify a valid argument for the **-o** option, you do not need to specify the **-t** option for any operating system drive letter, file system, Veritas volume, or native or PowerPath pseudo-device names. For

example, on Solaris systems, admsnap determines the **-o** argument of /**dev**/**rdsk**/**c***X***d***X***t***X***s***X* to be a Solaris native device name, so you do not need to specify the **-t** *device*.

**Example** Windows example:

**admsnap flush -o e:**

# admsnap list

**Performs a list operation and provides the World Wide Name (WWN) of all supported devices on the system.**

## Description

The **admsnap list** command will attempt to list devices and their World Wide Names (or WWN) on the operating system. You can use the WWN to distinguish one operating system device from another.

The WWN is a worldwide unique string that the storage system provides for its devices. Use the WWN to uniquely identify those devices on the server. Use this operation to match the server's devices with the devices the server sees on the storage system.

This command is supported on Windows, Solaris, HP-UX, Linux, and AIX operating systems only.

## Syntax

**admsnap list** [**-h**] [**-l** *lun_wwn*] [**-d** *object_name*] [**-a** *output_format*]

## Prerequisites

You must have administrative privileges on the production and secondary server.

## Options

[**-h**]

Lists the help output for a given operation. If you use this option on the command line, a help message is displayed for each option you specify. Admsnap will not perform the command, it lists the help output only.

This command is supported on Windows, Solaris, HP-UX, Linux, and AIX operating systems only.

[**-l** *lun_wwn*]

Lists the WWN of all supported devices on the system. The WWN must use a valid WWN format.

[**-d** *object_name*]

Specifies an object name. On Windows systems, the **-a** option is ignored when used with the **-d** option and the object name argument is a Windows physical device name, for example: **\\.\PhysicalDrive1**.

The table below lists the supported object names for your operating system.

| Operating system | Supported object name(s) |
|---|---|
| **Windows** | • Drive letter<br>• Physical drive name |
| **HP-UX**[a], **Linux, NetWare**, and **Solaris** | • Device name<br>• File system name[b]<br>• Veritas volume name[c] |
| **AIX**[a] | • Device name<br>• File system name[b] |

a. If the object name is on a Logical Volume Manager, then admsnap will parse the Logical Volume Manager (LVM) to get the device names and will *attempt* to list those devices.

b. If the object name is a file system, then admsnap will parse the file system table entry to get the device name and will *attempt* to list those devices.

c. If the object name is a Veritas volume name, admsnap will parse the Veritas volume to determine all physical disks for the volume and will *attempt* to list all physical disks.

[**-a** *output_format*]

Specifies an alternative output style on a Windows server. The default Windows output style is to list the drive letter or physical drive and its WWN. Valid output formats are drive letter (default if **-a** is not specified at all) or physical drive. If the output format contains spaces, enter it without spaces.

Admsnap will ignore the **-a** option when you specify the **-d** option and the object name argument is a Windows device name, for example **\\.\PhysicalDrive1**.

**Example** Windows example:

**admsnap list -d \\.\PhysicalDrive1**

# admsnap start

**Starts a point-in-time copy of a source LUN.**

## Description

The **admsnap start** operation starts a SnapView persistent session with the specified session name. A persistent session survives SP failures, software installation upgrades, and event trespasses.

**Note:** Admsnap version 6.24 or later enables all SnapView sessions to run in persistence mode as the default.

From the production server, the **start** command specifies storage systems that have a LUN or file system you want to copy, then it begins the point-in-time copy of the specified source.

**Note:** Before you start a SnapView session, admsnap requires that you create a snapshot using the **-createsnapshot** command (see page 139).

The **admsnap start** command is valid on Windows, Solaris, HP-UX, Linux, NetWare, and AIX systems.

## Syntax

**admsnap start** [**-h**] **-s** *session-name* **-o** *object-name* [**-t** *object_type*] [**-c**] **-p**

## Prerequisites

You must have administrative privileges on the production server.

## Options

[**-h**]

Lists the help output for a given operation. If you use this option on the command line, a help message is displayed for each option you specify. Admsnap will not perform the command; it lists the help output only.

This option is valid for Windows, Solaris, HP-UX, Linux, and AIX systems.

**-s** *session_name*

Specifies the session name that will be assigned to the session. This is a string of up to 256 characters, and can be composed of letters, numbers or underscores.

The session name is case sensitive and must be unique on the storage system. If the session name consists of spaces, you must enter the name in double quotation marks (for example, **"June 21st Session"**).

**Note:** If you issue multiple start operations from the production server (this includes other production servers) using the same session name, the session will consist of all the LUNs that make up those devices. This operation can be called multiple times (even from different servers or EMC Navisphere) with the same session name and the result is the new objects are added into the existing session.

**-o** *object_name*

Specifies an object name. You must specify an object name to identify the session. If you specify a device name or a drive letter, the LUN(s) you specify may have multiple partitions; if so, the snapshot will consist of all partitions on that LUN(s).

If you use the **-t** option, the device name must be a valid name for the type of object specified with the **-t** option.

**Note: Admsnap version 2.1 with a Windows or Solaris server** - If you specify an object name that spans multiple devices with the **admsnap start** command, and one of those devices fails to start, admsnap will stop the session on all the devices. For example, you start a session on a Solaris file system and that file system spans across two LUNs on the storage system. The first device successfully starts the session but the second does not. The admsnap software will stop the first session started on the device, and then you must re-start the session on the file system after you have corrected the cause of the failure.

The table below lists the supported object names for your operating system.

| Operating system | Supported object name(s) |
|---|---|
| **Windows** | • Drive letter<br>• Physical drive name |
| **HP-UX**[a], **Linux, NetWare,** and **Solaris** | • Device name<br>• File system name[b]<br>• Veritas volume name[c] |
| **AIX**[a] | • Device name<br>• File system name[b] |

a. If the object name is on a Logical Volume Manager, then admsnap will parse the Logical Volume Manager (LVM) to get the device names and will *attempt* to start the session on those devices.

b. If the object name is a file system, then admsnap will parse the file system table entry to get the device name and will *attempt* to start the session on that device.

c. If the object name is a Veritas volume name, admsnap will parse the Veritas volume to determine all physical disks for the volume and will *attempt* to start the session on each physical disk.

[**-t** *object_type*]

Specifies the object type. This switch is valid on Windows and Solaris operating systems only. Valid Windows object types are device and drive letter. Valid Solaris object types are device, volume, and file system. If the object type name contains spaces, enter the name without spaces.

You should use the **-t** option when the argument for the **-o** option is not in a standard and recognizable form for your operating system's object or device.

If you do not specify the **-t** option, admsnap will attempt to determine the correct object type and report a failure if it cannot.

[**-c**]

Specifies a consistent session. The session will preserve the point-in-time copy across a set of source LUNs. The SnapView driver will delay any I/O requests to the set of source LUNs until the session has started on all LUNs (thus preserving the point-in-time on the entire set of LUNs). Once you start a

consistent session, you cannot add another source LUN to the session. In the event of a failure, the software will not start the session on any source LUN and will display an error message.

**Note:** When starting a consistent session, multiple object names must be separated by a comma with no spaces. If desired, you can specify different object names. For examples, refer to page 193.

You would use consistent mode if

- you cannot stop I/O to the source LUN(s) before starting a session,
- you want to prevent other LUNs from being added to the session. If desired, you can start a consistent session on a single source LUN to prevent other LUNs from being added to the session,
- the set of source LUNs that you will use to start the session spans both SPs; however, the source LUNs cannot span storage systems.

The consistent feature is available on a per-session basis (not per snapshot or source LUN) and counts as one of the eight sessions per source LUN limit.

**Note:** While a consistent session is in the process of starting on all LUNs, you cannot stop the session (the software may stop the session immediately after the start completes if I/O is queued and there is not enough space in the SP's reserved LUN pool), perform a rollback operation, or activate a snapshot to the session.

**-p**

Specifies a persistent session. If you use this option, the session will run in persistence mode and will survive SP failures, software installation upgrades, and trespass events. With admsnap version 6.24 or later, all SnapView sessions run in persistence mode, so the **-p** switch is no longer required.

If you have an AX series storage system and are running an admsnap version prior to 6.24, you must specify the **-p** switch or an error message will be displayed.

**Examples**

Windows examples:

**admsnap start -s session1 -o \\.\PhysicalDrive1**

**admsnap start -s session1 -o \\.\PhysicalDrive1, PhysicalDrive2, PhysicalDrive3 -c**

**admsnap start -s session1 -o F:,G:,H:,I -c -p**

**admsnap start -s session1 -o F:,G:,\\.\PhysicalDrive1, PhysicalDrive2 -c**

UNIX examples:

**admsnap start -s session1 -o /mnt/fs1 -p**

**admsnap start -s session1 -o /mnt/fs1,/mnt/fs2,/mnt/fs3 -c**

**admsnap start -s session1 -o /dev/sda,/dev/sdc,/dev/sdd -c -p**

**admsnap start -s session1 -o /dev/sda,/mnt/fs2,vol_name -c**

# admsnap stop

**Stops the point-in-time copy of a source LUN.**

## Description

The **admsnap stop** operation stops a SnapView session under the session name. From the production server that owns the source LUN, the **stop** command stops a SnapView session, freeing the reserved LUN and disk space the session used.

The **admsnap stop** command is valid on Windows, Solaris, HP-UX, Linux, NetWare, and AIX systems.

**Note:** If you started a session on multiple source LUNs, you can select any of the source LUNs to stop the session.

## Syntax

**admsnap stop** [**-h**] **-s** *session-name* **-o** *object_name* [**-t** *object_type*]

## Prerequisites

You must have administrative privileges on the production server.

## Options

[**-h**]

Lists the help output for a given operation. If you use this option on the command line, a help message is displayed for each option you specify. Admsnap will not perform the command, it lists the help output only.

This option is valid for Windows, Solaris, HP-UX, Linux, and AIX systems.

**-s** *session_name*

Specifies the session name that you assigned when you started the session. You must specify the session name you used when you started the session. This is a string of up to 256 characters, and can be composed of letters, numbers or underscores.

The session name is case sensitive and must be unique on the storage system. If the session name consists of spaces, you must enter the name in double quotation marks (for example, **"June 21st Session"**).

**-o** *object_name*

Specifies an object name. You must specify an object name to identify the session.

If you use the **-t** option, the device name must be a valid name for the type of object specified with the **-t** option.

The table below lists the supported object names for your operating system.

| Operating system | Supported object name(s) |
|---|---|
| **Windows** | • Drive letter<br>• Physical drive name |
| **HP-UX**[a], **Linux, NetWare**, and **Solaris** | • Device name<br>• File system name[b]<br>• Veritas volume name[c] |
| **AIX**[a] | • Device name<br>• File system name[b] |

a. If the object name is on a Logical Volume Manager, then admsnap will parse the Logical Volume Manager (LVM) to get the device names and will *attempt* to stop the session on those devices.

b. If the object name is a file system, then admsnap will parse the file system table entry to get the device name and will *attempt* to stop the session on that device.

c. If the object name is a Veritas volume name, admsnap will parse the Veritas volume to determine all physical disks for the volume and will *attempt* to stop the session on each physical disk.

The only time admsnap will not stop the session on all devices within an object is when the device is also in another object that already has the same session name started.

[**-t** *object_type*]

Specifies the object type. This switch is valid on Windows or Solaris operating systems only. Valid types on Windows are device and drive letter. Valid types on Solaris are device, volume, and file system. If the object type name contains spaces, enter the name without spaces.

You should use the **-t** option when the argument for the **-o** option is not in a standard and recognizable form for your operating system's object or device. If you do not specify the **-t** option, admsnap will attempt to determine the correct object type and report a failure if it cannot.

**Example** Windows example:

**admsnap stop -s session1 -o \\.\PhysicalDrive1**

# 6

# SnapView Examples

This chapter contains examples, from setting up clones and snapshots (with Navisphere CLI) to using them (with admsnap and Navisphere CLI). Some examples contain an illustrated overview that shows the main steps outlined in the examples; other examples are specific to a particular platform.

Major sections are:

### Clone examples



### Snapshot examples

## Step-by-step clone overview - all platforms

In the following example, you will use the SnapView clone CLI commands (explained in Chapter 2) in addition to the admsnap clone commands to set up (from the production server) and use a clone (from the secondary server).

1. On the storage system, bind a LUN for each SP to serve as a clone private LUN.

   The clone private LUNs (one for each SP) are shared by all clone groups on a storage system.

   The clone private LUNs store temporary system information used to speed up synchronization of the source LUN and its clone. These structures are called fracture logs. The clone private LUN can be any public LUN that is not part of any storage group. The minimum and standard size for each clone private LUN is 1 Gigabyte. There is no benefit in performance, or otherwise, to using clone private LUNs larger than 1 Gigabyte.

2. On the storage system, bind a LUN to serve as the clone. Each clone should be the same size as the source LUN. The source and clone LUNs can be on the same SP or different SPs.

3. If the source LUN does not exist (for example, because you are creating a new database), you can bind it at the same time as the clone. Then you can add the new source LUN to a storage group.

4. On the storage system, allocate the clone private LUNs. Use the CLI command function **-allocatecpl** for this (see page 61).

5. On the storage system, create the clone group. Use the CLI command function **-createclonegroup** for this (see page 72).

6. If the LUN you choose as your clone is mounted on a secondary server, deactivate the LUN from the server it is mounted on by issuing the appropriate command for your operating system:

   - On a Windows server, use the following admsnap command:

     **admsnap clone_deactivate -o** *clone drive_letter*

   - On a UNIX server, unmount the file system on the LUN you want to use as a clone by issuing the **umount** command.
   - On a Novell NetWare server, use the **dismount** command on the volume to dismount the file system.

7. On the storage system, add the LUN you bound as your clone in step 2 to the clone group. Use the CLI command **-addclone** for this (see page 57).

   By default, when you use the **-addclone** command, the software starts synchronizing the clone (copying source LUN data to the clone). If the source LUN has meaningful data on it, then synchronization is necessary. Depending on the size of the source LUN, a synchronization may take several hours.

   If you do not want the default synchronization to occur when you add the clone to the clone group, then you can tell the CLI that synchronization is not required. To do this use the **-issyncrequired** option in the **-addclone** command. An initial synchronization is not required if your source LUN does not contain any data. If you specify an initial sync with an empty source LUN, resources are used to synchronize the source LUN to the clone LUN.

8. After the clone is synchronized, do the following before fracturing it:

   a. Quiesce I/O to the source LUN.

   b. Flush all cached data to the source LUN by issuing the appropriate command for your operating system.

      – For a Windows server, use the **admsnap flush** command to flush all server buffers.

        **admsnap flush -o E:**

      – For Solaris, HP-UX, AIX, and Linux servers, unmount the file system by issuing the **umount** command. If you are unable to unmount the file system, you can issue the **admsnap flush** command.

        **admsnap flush -o /dev/rdsk/c1t0d2s2**

      – For a Novell NetWare server, use the **dismount** command on the volume to dismount the file system.

   **Note:** Neither the **flush** command nor the **sync** command is a substitute for unmounting the file system. Both commands only complement unmounting the file system.

With some operating systems, additional steps may be required from the secondary server in order to flush all data and clear all buffers on the secondary server. For more information, see the product release notes.

c. Wait for the clone to transition to the synchronized state.

9. Fracture the clone using the CLI **fracture** command. Use the CLI command function **-fractureclone** for this (see page 76).

10. Assign the clone to a storage group. You must assign the clone LUN to a storage group other than the storage group that holds the source LUN. Use the Navisphere CLI command **storagegroup** as described in the *EMC Navisphere Command Line Interface (CLI) Reference*.

11. Activate the clone.

    - For Windows, use the **admsnap clone_activate** command to make the newly fractured clone available to the operating system. After a delay, the **admsnap clone_activate** command finishes rescanning the system and assigns drive letters to newly discovered clone devices.

      **Note:** If the secondary server is running Windows 2000, a reboot is recommended but not required after you activate the fractured clone.

    - For UNIX servers, for all platforms except Linux, **clone_activate** tells the operating system to scan for new LUNs. For Linux, you must either reboot the server or unload and load the HBA driver.
    - On a NetWare server, run the command **list devices** or use the command **scan all LUNs** on the console.

12. If you have a VMware ESX Server**,** do the following:

    a. Rescan the bus at the ESX Server level.

    b. If a virtual machine is already running, power off the virtual machine and use the Service Console of the ESX Server to assign the clone to the virtual machine.

       If a virtual machine is *not* running, create a virtual machine on the ESX Server and assign the clone to the virtual machine.

    c. Power on the virtual machine and scan the bus at the virtual machine level. For virtual machines running Windows, you can use the **admsnap activate** command to rescan the bus.

13. Verify that the contents of the source LUN and clone LUN are in sync.
14. Resume I/O to the source LUN.
15. Use the fractured clone as you wish—for backup, reverse synchronization, or other use.
16. To update the clone with any source LUN modifications made since you fractured it, perform the following steps:
    a. Flush and deactivate the clone.
       – For Windows, use the **admsnap flush** and **admsnap clone_deactivate** commands to flush all server buffers, to dismount, and to remove the drive letter assigned by **clone _activate**. For multi-partitioned clone devices, those having more than one drive letter mounted on it, all other drive letters associated with this physical clone device will also be flushed, dismounted, and removed.

         **admsnap flush -o E:**

         **admsnap clone_deactivate E:**

       – For UNIX, unmount the file system by issuing the **umount** command. If you cannot unmount the file system, you can use the **sync** command to flush buffers. The **sync** command is not considered a substitute for unmounting the file system, but you can use it to reduce the number of incidents of having to **fsck** the file system on your backup server. Refer to your system's man pages for **sync** command usage.
       – For NetWare, use the **dismount** command on the clone volume to dismount the file system.

    b. Remove the clone from the storage group.
    c. Synchronize the clone. Use the CLI command **-syncclone** for this (see page 97).
    d. Repeat steps 8 - 15.
17. When you have finished with this clone, you can remove the clone from its clone group. You can also do the following:
    - Destroy the clone group by using the CLI command **-destroyclonegroup** (see page 75).
    - Remove the clone LUN by using the CLI command **-removeclone** (see page 89).

- Deallocate the clone private LUNs by using the CLI command **-deallocatecpl** (see page 74).

## Clone use illustration

The following figure shows how you might use a clone.

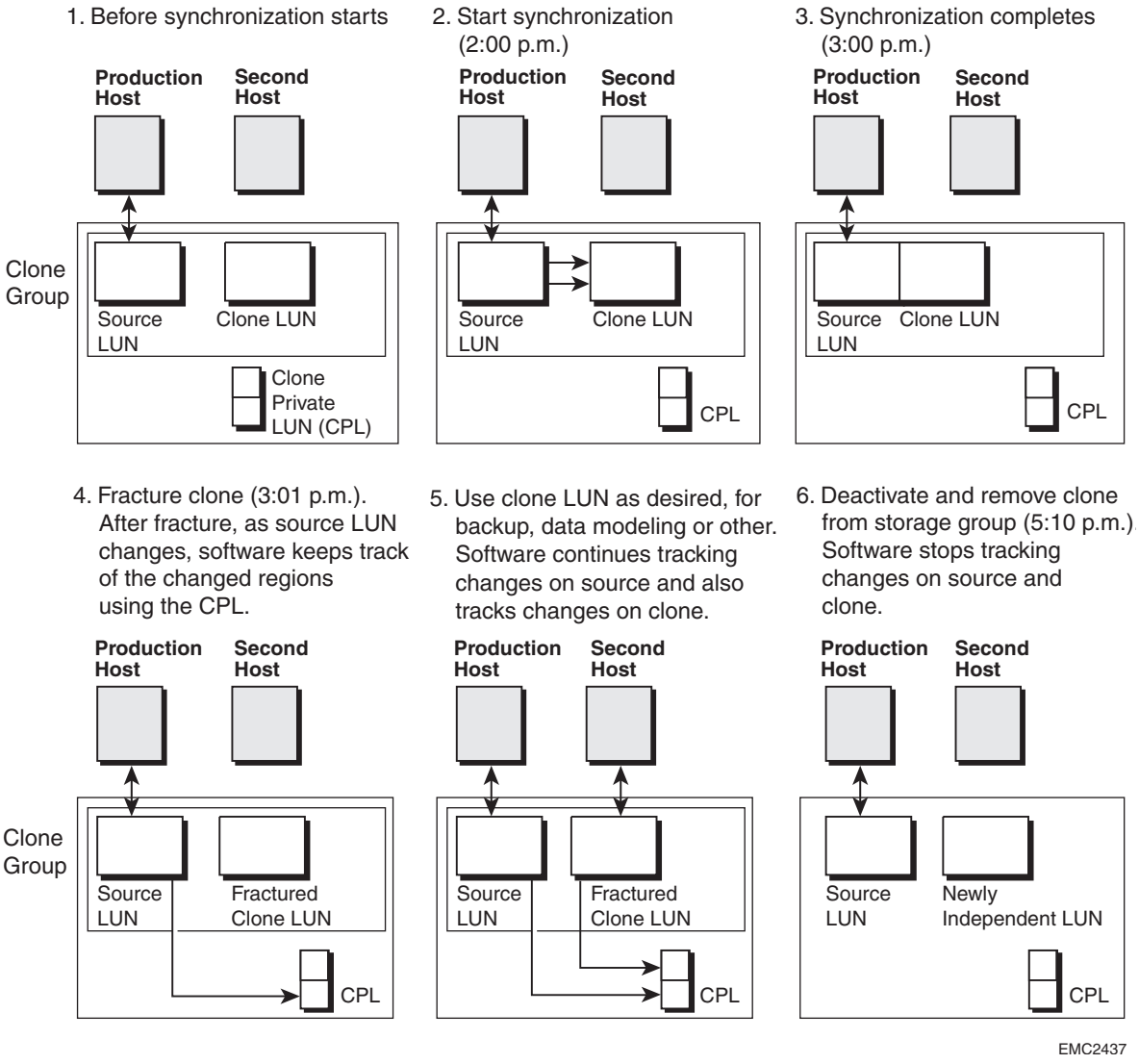


**Figure 4 Using a clone**

## Windows - clone example

The following example shows all the **naviseccli** or **navicli** and **admsnap** commands needed to set up and use a clone on a Windows platform. It includes binding and unbinding the LUNs and RAID groups.

1. Create the source and clone RAID groups and bind the LUNs.

   ```
   naviseccli -h ss_spA createrg 10   1_0  1_1  1_2  1_3  1_4
   naviseccli -h ss_spA createrg 11   1_5  1_6  1_7  1_8  1_9
   naviseccli -h ss_spA  bind r5  20  -rg  10  -sp  A
   naviseccli -h ss_spA bind r5  21  -rg  11  -sp  A
   ```

   **Note:** To use these commands with **navicli**, replace **naviseccli** with **navicli**.

2. Create the clone private LUNs, each 1 Gigabyte long.

   ```
   naviseccli -h ss_spA createrg 100   2_1  2_2  2_3  2_4  2_5
   naviseccli -h ss_spA  bind r5 100 -rg 10 -sp A -sq mb -cp 200
   naviseccli -h ss_spa bind r5 101 -rg 10 -sp A -sq mb -cp 200
   ```

   **Note:** To use these commands with **navicli**, replace **naviseccli** with **navicli**.

3. Wait for all the LUNs to complete binding. Then set up the storage groups.

   ```
   naviseccli -h ss_spa storagegroup -create -gname Production
   naviseccli -h ss_spa storagegroup -create -gname Backup

   naviseccli -h ss_spa storagegroup -connecthost -o -server
   ServerABC -gname Production

   naviseccli -h ss_spa storagegroup -connecthost -o -server
   ServerXYZ  -gname Backup

   naviseccli -h ss_spa storagegroup -addhlu -gname Production
   -hlu 20 -alu 20

   naviseccli -h ss_spa storagegroup -addhlu -gname Backup -hlu
   21 -alu 21
   ```

   **Note:** To use this command with **navicli**, replace **naviseccli** with **navicli**.

4. On both servers, rescan or reboot to let the operating systems see the new LUNs.
5. Allocate the clone private LUNs.

   **naviseccli -User GlobalAdmin -Password mypasssw -Scope 0 -Address ss_spa clone -allocatecpl -spA 100 -spB 101 -o**
6. Create the clone group and add the clone.

   **naviseccli -User GlobalAdmin -Password mypassw -Scope 0 -Address ss_spa clone -createclonegroup -name lun20_clone -luns 20 -description Creatinglun20_clone -o**

   **naviseccli -User GlobalAdmin -Password password -Scope 0 -Address ss_spa clone -addclone -name lun20_clone -luns 20**
7. Run Disk Management on the production server and create an NTFS file system on the source LUN. Copy files to the drive letter assigned to the source LUN on the production server. This example uses **g:** as the driver letter for the source LUN,
8. On the production server, run admsnap to write the buffers.

   **admsnap flush -o g:**

   The clone transitions to the synchronized state.
9. Fracture the clone.

   **naviseccli -User GlobalAdmin -Password password -Scope 0 -Address ss_spa clone -fractureclone -name lun20_clone -cloneid 0100000000000000 -o**
10. On the secondary server, run admsnap to activate the clone.

    **admsnap clone_activate**

    The admsnap software returns a drive letter for the drive assigned to the clone that was just fractured. This example uses **h**: as the drive letter for the clone LUN.
11. Verify that the files that were copied to the source LUN also appear on the clone LUN.
12. If you have a VMware ESX Server, do the following:
    a. Rescan the bus at the ESX Server level.
    b. If a virtual machine is already running, power off the virtual machine and use the Service Console of the ESX Server to assign the clone to the virtual machine.

If a virtual machine is *not* running, create a virtual machine on the ESX Server and assign the clone to the virtual machine.

c. Power on the virtual machine and scan the bus at the virtual machine level. For virtual machines running Windows, you can use the **admsnap activate** command to rescan the bus.

13. On the secondary server, delete the existing files and copy different files to the clone (to **h:**).
14. On the secondary server, run admsnap to deactivate the clone.

    **admsnap clone_deactivate -o h:**
15. On the production server, run admsnap to deactivate the source.

    **admsnap clone_deactivate -o g:**
16. Reverse synchronize to copy the data written to the clone back to the source.

    **naviseccli -User GlobalAdmin -Password password -Scope 0 -Address ss_spa clone -reversesyncclone -name lun20_clone -cloneid 0100000000000000 -o**
17. On the production server, run admsnap to activate the source.

    **admsnap clone_activate**

    Wait for the reverse-sync operation to complete and the clone to transition to the synchronized state.
18. Fracture the clone again to make the source independent.

    **naviseccli -User GlobalAdmin -Password password -Scope 0 -Address ss_spa clone -fractureclone -name lun20_clone -cloneid 0100000000000000 -o**
19. On the production server, verify that the clone (**g**:) contains the files that were written to the clone on the secondary server. It also should not contain the files that were deleted from the clone.
20. On the production server, use admsnap to deactivate the source.

    **admsnap clone_deactivate -o g:**

21. Clean up the storage system by removing and destroying the clone group.

    **naviseccli -User GlobalAdmin -Password password -Scope 0 -Address ss_spa clone -removeclone -name lun20_clone -cloneid 0100000000000000 -o**

    **naviseccli -User GlobalAdmin -Password password -Scope 0 -Address ss_spa clone -destroyclonegroup -name lun20_clone -o**

## Reverse synchronization - all platforms

The following example illustrates the admsnap and Navisphere CLI commands required to reverse synchronize a fractured clone.

1. From the production server, stop I/O to the source LUN.
2. Using admsnap, do the following:
   a. From the production server, deactivate the source LUN by issuing the appropriate command for your operating system.
      – On a Windows server, use the following admsnap command:

        **admsnap clone_deactivate -o** *source-drive-letter*

      – On a UNIX server, unmount the file system by issuing the **umount** command. If you cannot unmount the file system, use the **sync** command to flush buffers. Although the **sync** command is not a substitute for unmounting the file system, you can use it to reduce the number of times you need to issue the **fsck** command on the secondary server's file system. Refer to your system's man pages for **sync** command usage.
      – On a NetWare server, use the **dismount** command on the volume to dismount the file system.
   b. If the clone is mounted on a secondary server, flush all cached data to the clone LUN by issuing the appropriate command for your operating system.
      – For a Windows server, use the **admsnap flush** command.
      – For Solaris, HP-UX, AIX, and Linux servers, unmount the file system by issuing the **umount** command. If you are unable to unmount the file system, issue the **admsnap flush** command. The **flush** command flushes all data and clears all buffers.
      – On a Novell NetWare server, use the **dismount** command on the volume to dismount the file system.

**Note:** Neither the **flush** command nor the **sync** command is a substitute for unmounting the file system. Both commands only complement unmounting the file system.

With some operating systems, additional steps may be required from the secondary server in order to flush all data and clear all buffers on the secondary server. For more information, see the product release notes.

3. Using Navisphere CLI, issue the following command from the SP that owns the source LUN:

   **clone -reversesyncclone -name** *name* | **-clonegroupUid** *uid* **-cloneid** *id* [**-UseProtectedRestore 0** | **1**]

   **Note:** Before you can use the **protected restore** feature, you must globally enable it by issuing the **clone | snapview -changeclonefeature [-AllowProtectedRestore 1]** command.

   **Important** When the reverse synchronization begins, the software automatically fractures all clones in the clone group.

   Depending on whether or not you enabled the **Protected Restore** feature, the following occurs to the clone that initiated the reverse synchronization:

   - ***With* the Protected Restore feature** - the software fractures the clone after the reverse synchronization completes.
   - ***Without* the Protected Restore feature** - the software leaves the clone unfractured.

## Step-by-step snapshots overview - all platforms

In the following procedures, you will use the SnapView snapshot CLI commands in addition to the admsnap snapshot commands to set up (from the production server) and use snapshots (from the secondary server).

1. Choose the LUNs for which you want a snapshot. The size of these LUNs will help you determine an approximate reserved LUN pool size. The LUN(s) in the reserved LUN pool stores the original data when that data is first modified on the source LUN(s).

   **Note:** To manually estimate a suitable LUN pool size, refer to **Managing Storage Systems > Configuring and Monitoring the Reserved LUN Pool** in the Table of Contents for the Navisphere Manager online help and select the **Estimating the Reserved LUN Pool Size** topic.

2. Configure the reserved LUN pool.

   **Note:** You must configure the reserved LUN pool before you start a SnapView session. Use Navisphere Manager to configure the reserved LUN pool (refer to the Navisphere Manager online help topic **Managing Storage Systems > Configuring and Monitoring the Reserved LUN Pool).** Stop I/O and make sure all data cached on the production server is flushed to the source LUN(s) before issuing the **admsnap start** command.

   - For a Windows server, you can use the **admsnap flush** command to flush the data.
   - For Solaris, HP-UX, AIX, and Linux servers, unmount the file system by issuing the **umount** command. If you are unable to unmount the file system, you can issue the **admsnap flush** command.
   - For a Novell NetWare server, use the **dismount** command on the volume to dismount the file system.

   **Note:** Neither the **flush** command nor the **sync** command is a substitute for unmounting the file system. Both commands only complement unmounting the file system.

3. On the production server, log in as **admin** or **root** and issue an **admsnap start** command for the desired data object (drive letter, device name, or file system) and session name. The **admsnap start** command starts the session. You must start a session for each snapshot of a specific LUN(s) you want to access simultaneously.

   You start a session from the production server based on the source LUN(s). You will mount the snapshot on a different server (the secondary server). You can also mount additional snapshots on other servers.

   You can start up to eight sessions per source LUN. This limit includes any reserved sessions that are used for another application such as SAN Copy and MirrorView/Asynchronous. However, only one SnapView session can be active on a secondary server at a time. If you want to access more than one snapshot simultaneously on a secondary server (for example, 2:00 p.m. and 3:00 p.m. snapshots of the same LUN(s), to use for rolling backups), you can create multiple snapshots, activate each one on a different SnapView session and add the snapshots to different storage groups. Or you can activate and deactivate snapshots on a single server.

The SnapView driver will use this moment as the beginning of the session and will make a snapshot of this data available. Sample **start** commands follow.

| |
|---|
| IBM AIX Server (UNIX)<br>**admsnap start -s session1 -o /dev/hdisk21** (for a device name)<br>**admsnap start -s session1 -o /database** (for a file system) |
| HP-UX Server (UNIX)<br>**admsnap start -s session1 -o /dev/rdsk/c0t0d0** (for a device name)<br>**admsnap start -s session1 -o /database** (for a file system)<br><br>Veritas Volume examples:<br><br>Example of a Veritas volume name:<br>**scratch**<br><br>Example of a fully qualified pathname to a Veritas volume:<br>**admsnap start -s session1 -o /dev/vx/dsk/scratchdg/scratch**<br><br>Example of a fully qualified pathname to a raw Veritas device name:<br>**admsnap start -s session1 -o /dev/vx/rdmp/c1t0d0** |
| Linux Server (UNIX)<br>**admsnap start -s session1 -o /dev/sdc** (for a device name)<br>**admsnap start -s session1 -o /database** (for a file system)<br><br>Veritas Volume examples:<br><br>Example of a Veritas volume name:<br>**scratch**<br><br>Example of a fully qualified pathname to a Veritas volume:<br>**admsnap start -s session1 -o /dev/vx/dsk/scratchdg/scratch**<br><br>Example of a fully qualified pathname to a raw Veritas device name:<br>**admsnap start -s session1 -o /dev/vx/rdmp/sdc6** |

| |
|---|
| NetWare Server<br>**load sys:\emc\admsnap\admsnap start -s session1 -o V596-A2-D0:2** (for a device name)<br>(**V596** is the vendor number.) |
| Sun Solaris Server (UNIX)<br>**admsnap start -s session1 -o /dev/rdsk/c0t0d0s7** (for a device name)<br>**admsnap start -s session1 -o /database** (for a file system)<br>Veritas Volume examples:<br>Example of a Solaris Veritas volume name:<br>**scratch**<br>Example of a fully qualified pathname to a Veritas volume:<br>**admsnap start -s session1 -o /dev/vx/dsk/scratchdg/scratch**<br>Example of a fully qualified pathname to a raw Veritas device name:<br>**admsnap start -s session1 -o /dev/vx/rdmp/c1t0d0s2** |
| Windows Server<br>**admsnap start -s session1 \.\\PhysicalDrive1** (for a physical drive name)<br>**admsnap start -s session1 -o H:** (for a drive letter) |

4. Using Navisphere CLI, create a snapshot of the source LUN(s) for the storage system that holds the source LUN(s), as follows. You must create a snapshot for each session you want to access simultaneously.

   Use the **naviseccli** or **navicli snapview** command with **-createsnapshot** to create each snapshot.

   **naviseccli -h** *hostname* **snapview -createsnapshot**

5. **If you do *not* have a VMware ESX Server** - Use the **storagegroup** command to assign each snapshot to a storage group on the secondary server.

   **If you have a VMware ESX Server** - skip to step 7 to activate the snapshot.

6. On the secondary server, use an **admsnap activate** command to make the new session available for use. A sample **admsnap activate** command is:

   **admsnap activate -s session1**

   - On a Windows server, the **admsnap activate** command finishes rescanning the system and assigns drive letters to newly discovered snapshot devices. You can use this drive immediately.
   - On an AIX server, you need to import the snap volume (LUN) by issuing the **chdev** and **importvg** commands as follows:

     **chdev -l hdisk***n* **-a pv=yes** (This command is needed only once for any LUN.)

     **importvg -y** *volume-group-name* **hdisk***n*

     where *n* is the number of the **hdisk** that contains a LUN in the volume group and *volume-group-name* is the volume group name.
   - On a UNIX server, after a delay, the **admsnap activate** command returns the snapshot device name. You will need to run **fsck** on this device only if it contains a file system and you did not unmount the source LUN(s). Then, if the source LUN(s) contains a file system, mount the file system on the secondary server using the snapshot device name to make the file system available for use. If you failed to flush the file system buffers before starting the session, the snapshot may not be usable.

     Depending on your operating system platform, you may need to perform an additional step before **admsnap activate** to rescan the I/O bus. For more information, see the product release notes.

     For UNIX, run **fsck** on the device name returned by the admsnap command, but when you mount that device using the **mount** command, use a device name beginning with **/dev/dsk** instead of the device name **/dev/rdsk** as returned by the admsnap command.
   - On a NetWare server, issue a **list devices** or **Scan All LUNs** command from the server console. After a delay, the system returns the snapshot device name. You can then mount the volume associated with this device name to make a file system

available for use. You may need to perform an additional step to rescan the I/O bus. For more information, see the product release notes.

7. If you have a VMware ESX Server, do the following:

   a. Use the **storagegroup** command to add the snapshot to a storage group connected to the ESX Server that will access the snapshot.

   b. Rescan the bus at the ESX Server level.

   c. If a virtual machine is already running, power off the virtual machine and use the Service Console of the ESX Server to assign the snapshot to the virtual machine.

      If a virtual machine is *not* running, create a virtual machine on the ESX Server and assign the snapshot to the virtual machine.

   d. Power on the virtual machine and scan the bus at the virtual machine level. For virtual machines running Windows, you can use the **admsnap activate** command to rescan the bus.

8. On the secondary server, you can access data on the snapshot(s) for backup, data analysis, modeling, or other use.

9. On the secondary server, when you finish with the snapshot data, release each active snapshot from the operating system as follows:

   - On a Windows server, release each snapshot device you activated, using the **admsnap deactivate** command.
   - On an AIX server, you need to export the snap volume (LUN) by issuing the **varyoff** and **export** commands as follows:

     **varyoffvg** *volume-group-name*

     **exportvg** *volume-group-name*

     Then release each snapshot device you activated, using the **admsnap deactivate** command.
   - On a UNIX server, you need to unmount any file systems that were mounted from the snapshot device by issuing the **umount** command. Then release each snapshot device you activated, using the **admsnap deactivate** command.

- On a NetWare server, use the **dismount** command on the volume to dismount the file system.

A deactivate command is required for each active snapshot. If you do not deactivate a snapshot, the secondary server cannot activate another session using the pertinent source LUN(s). When you issue the **admsnap deactivate** command, any writes made to the snapshot are destroyed.

10. On the production server, stop the session using the **admsnap stop** command. This frees the reserved LUN and SP memory used by the session, making them available for use by other sessions.

    Sample **admsnap stop** commands are identical to the **start** commands shown in step 3. Substitute **stop** for **start**.

11. If you will not need the snapshot of the source LUN(s) again soon, use CLI **snapview -rmsnapshot** command to remove it.

    If you remove the snapshot, then for a future snapshot you must execute all previous steps. If you do not remove the snapshot, then for a future snapshot you can skip steps 4 through 6.

## SnapView session

The following figure shows how a SnapView session starts, runs, and stops.

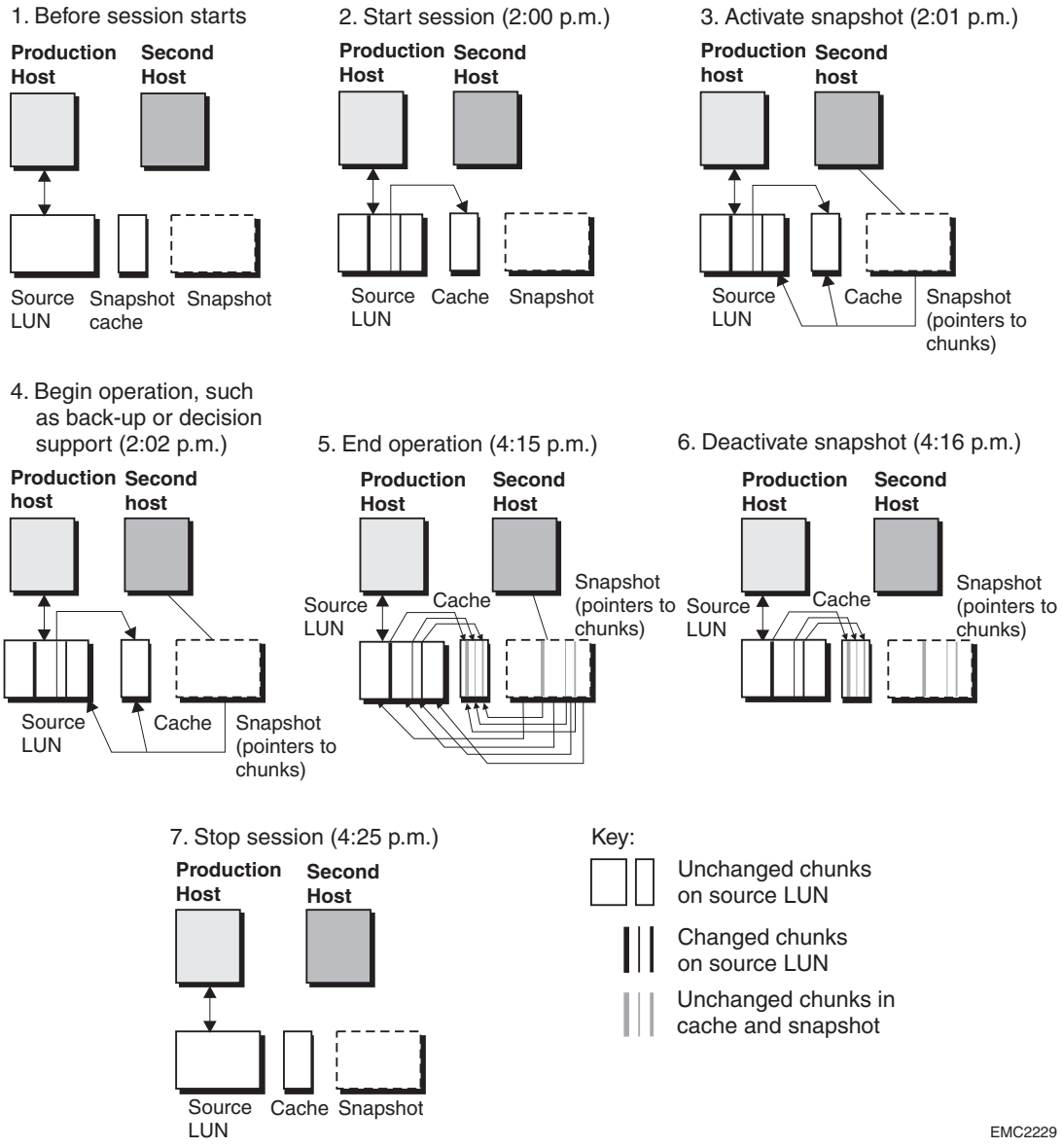


**Figure 5 Using a session and snapshot**

## HP-UX - admsnap snapshot script example

The following example shows how to use admsnap with scripts for copying and accessing data on an HP-UX secondary server.

1. From the production server, create the following script:

   **Script 1**

   a. Quiesce I/O on the source server.

   b. Unmount the file system by issuing the **umount** command. If you are unable to unmount the file system, issue the **admsnap flush** command. The **flush** command flushes all cached data.

   **Note:** The **flush** command is not a substitute for unmounting the file system; the command only complements the unmount operation.

   c. Start the session by issuing the following command:

   **/usr/admsnap/admsnap start -s** *snapsession_name* **-o** *device_name or filesystem_name*

   d. Invoke Script 2 on the secondary server using the **remsh** command.

   e. Stop the session by issuing the following command:

   **/usr/admsnap/admsnap stop -s** *snapsession_name* **-o** *device_name or filesystem_name*

2. From the secondary server, create the following script:

   **Script 2**

   a. Perform any necessary application tasks in preparation for the snap activation (for example, shut down database).

   b. Activate the snapshot by issuing the following command:

   **/usr/admsnap/admsnap activate -s** *snapsession_name*

   c. Create a new volume group directory, by using the following form:

   **mkdir**/**dev**/*volumegroup_name*

   **mknod**/**dev**/*volumegroup_name*/**group c 64 0x X0000**

d. Issue the **vgimport** command, using the following form:

**vgimport** *volumegroup_name*/**dev**/**dsk**/**c***N***t***N***d***N*

e. Activate the volume group for this LUN by entering the following command:

**vgchange -a y** *volumegroup_name*

f. Run **fsck** on the volume group by entering the following:

**fsck -F** *filesystem_type* **/dev/***volumegroup_name*/*logicalvolume_name*

**Note:** This step is not necessary if the secondary server has a different HP-UX operating system revision than the production server.

g. Mount the file system using the following command:

**mount**/**dev**/*volumegroup_name*/*logicalvolume_name*/*filesystem_name*

h. Perform desired tasks with the mounted data (for example, copy contents of mounted file system to another location on the secondary server).

i. Unmount the file system mounted in step g using the following command:

**umount /dev/***volumegroup_name*/*logicalvolume_name*

j. Deactivate and export the volume group for this LUN, by issuing the following commands:

**vgchange -a n** *volumegroup_name*

**vgexport** *volumegroup_name*

k. Unmount the file system by issuing the **umount** command. If you are unable to unmount the file system, issue the **admsnap flush** command. The **flush** command flushes all cached data. If this is not done, the next admsnap session may yield stale data.

l. Deactivate the snapshot by using the following command:

**/usr/admsnap/admsnap deactivate -s** *snapsession_name*

m. Perform any necessary application tasks in preparation for using the data captured in step 6 (for example, start up the database).

n. Exit this script, and return to Script 1.

## UNIX - admsnap single session example

The following commands start, activate, and stop a SnapView session. This example shows UNIX device names.

1. On the production server, make sure all cached data is flushed to the source LUN, by unmounting the file system.

   **umount /dev/dsk/c12d0s4**

   If you are unable to unmount the file system on a Solaris, HP-UX, AIX, or Linux server, you can issue the **admsnap flush** command.

   **admsnap flush -o/dev/rdsk/c12d0s4**

   **Note:** Neither the **flush** command nor the **sync** command is a substitute for unmounting the file system. Both commands only complement unmounting the file system.

2. Start the session:

   **admsnap start -s friday -o /dev/rdsk/c1t2d0s4**

   ```
   Attempting to start session friday on device
            /dev/rdsk/c1t2d0s4
   Attempting to start the session on the entire LUN.
   Started session friday.
   ```

   The start command starts a session named **friday** with the source named **/dev/rdsk/c1t2d0s4**.

3. On the secondary server, activate the session:

   **admsnap activate -s friday**

   ```
   Session friday activated on /dev/rdsk/c1t2d0s4.
   ```

   On the secondary server, the **activate** command makes the snapshot image accessible.

4. On a UNIX secondary server, if the source LUN has a file system, mount the snapshot:

   **mount /dev/dsk/c5t3d2s1 /mnt**

5. On the secondary server, the backup or other software accesses the snapshot as if it were a standard LUN.

6. When the desired operations are complete, from the secondary server, unmount the snapshot. With UNIX, you can use **admsnap deactivate** to do this.

   **admsnap deactivate -s friday -o /dev/dsk/c5t3d2s1**

7. And from the production server, stop the session:

   **admsnap stop -s friday -o /dev/dsk/c1t2d0s4**

   ```
   Stopped session friday on object /dev/rdsk/c1t2d0s4.
   ```

   The **stop** command terminates session **friday**, freeing the reserved LUN used by the session, and making the snapshot inaccessible.

# Windows - admsnap multiple session example

The following example shows three SnapView sessions, started and activated sequentially, using Windows device names. The example shows how each snapshot maintains the data at the time the snapshot was started—here, the data is a listing of files in a directory. The activity shown here is the only activity on this LUN during the sessions.

## Procedural overview

1. Make sure the directory that holds admsnap is on your path.
2. Start sessions **snap1**, **snap2**, **and snap3** on the production server in sequence and activate each session in turn on the secondary server. All sessions run on the same LUN.
3. When session **snap1** starts, four files exist on the LUN. Before starting **snap2**, create four more files in the same directory. Then, on the secondary server, deactivate **snap1.** The deactivate is needed because only one session can be active per server at one time.
4. On the production server start **snap2**, and on the secondary server activate **snap2**. After activating **snap2**, list files, displaying the files created between session starts.
5. Create three more files on the source LUN and start session **snap3.** After deactivating **snap2** and activating **snap3**, verify that you see the files created between the start of sessions **snap2** and **snap3.** The filenames are self-explanatory.

## Detailed procedures with output examples

### Session Snap1

1. On the production server, list files in the **test** directory.

```
F:\> cd test
F:\Test> dir
..
Directory of F:\Test
01/21/2002  09:23a  <DIR> .
01/21/2002  09:23a  <DIR> ..
01/21/2002  09:21a        0 FilesBeforeSession1-a.txt
01/21/2002  09:21a        0 FilesBeforeSession1-b.txt
01/21/2002  09:21a        0 FilesBeforeSession1-c.txt
01/21/2002  09:21a        0 FilesBeforeSession1-d.txt
               4 File(s)              0 bytes
               2 Dir(s)     102,225,920 bytes free
```

2. On the production server, flush data on source LUN, and then start the first session, **snap1**.

```
F:\Test> admsnap flush -o f:
F:\Test> admsnap start -s snap1 -o f:
Attempting to start session snap1 on device
\\.\PhysicalDrive1.
Attempting to start session on the entire LUN.
Started session snap1.
F:\Test>
```

3. On the secondary server, activate the first session, **snap1**.

```
C:\> prompt $t $p
14:57:10.79 C:\> admsnap activate -s snap1
Scanning for new devices.
Activated session snap1 on device F:.
```

4. On the secondary server, list files to show the production files that existed at session 1 start.

```
14:57:13.09 C:\ dir f:\test
...
Directory of F:\Test
01/21/02  09:23a  <DIR>          .
01/21/02  09:23a  <DIR>          ..
01/21/02  09:21a                 0 FilesBeforeSession1-a.txt
01/21/02  09:21a                 0 FilesBeforeSession1-b.txt
01/21/02  09:21a                 0 FilesBeforeSession1-c.txt
01/21/02  09:21a                 0 FilesBeforeSession1-d.txt
               6 File(s)              0 bytes
                            102,225,920 bytes free
```

### Session Snap2

1. On the production server, list files in the **test** directory. The listing shows files created before session 1 started. Notice that we created four additional files.

```
F:\Test> dir
...
Directory of F:\Test
01/22/2002  03:03p  <DIR>          .
01/22/2002  03:03p  <DIR>          ..
01/21/2002  09:21a         0 FilesAfterS1BeforeS2-a.txt
01/21/2002  09:21a         0 FilesAfterS1BeforeS2-b.txt
01/21/2002  09:21a         0 FilesAfterS1BeforeS2-c.txt
01/21/2002  09:21a         0 FilesAfterS1BeforeS2-d.txt
01/21/2002  09:21a          0 FilesBeforeSession1-a.txt
01/21/2002  09:21a          0 FilesBeforeSession1-b.txt
01/21/2002  09:21a          0 FilesBeforeSession1-c.txt
01/21/2002  09:21a          0 FilesBeforeSession1-d.txt
               8 File(s)              0 bytes
               2 Dir(s)     102,225,920 bytes free
```

2. On the production server, start the second session, **snap2**.

```
F:\Test> admsnap flush -o f:
F:\Test> admsnap start -s snap2 -o f:
Attempting to start session snap2 on device
\\.\PhysicalDrive1.
Attempting to start the session on the entire LUN.
Started session snap2.
F:\
```

3. On the secondary server, deactivate the session **snap1**, and activate the second session, **snap2**.

```
15:10:10.52 C:\> admsnap deactivate -s snap1
Deactivated session snap1 on device F:.

15:10:23.89 C:\> admsnap activate -s snap2
Activated session snap2 on device F:.
```

4. On the secondary server, list files to show source LUN files that existed at session 2 start.

```
15:10:48.04 C:\> dir f:\test
...
Directory of F:\Test
01/22/02  03:03p  <DIR>          .
01/22/02  03:03p  <DIR>          ..
01/21/02  09:21a         0 FilesAfterS1BeforeS2-a.txt
01/21/02  09:21a         0 FilesAfterS1BeforeS2-b.txt
01/21/02  09:21a         0 FilesAfterS1BeforeS2-c.txt
01/21/02  09:21a         0 FilesAfterS1BeforeS2-d.txt
01/21/02  09:21a          0 FilesBeforeSession1-a.txt
```

```
01/21/02  09:21a            0 FilesBeforeSession1-b.txt
01/21/02  09:21a            0 FilesBeforeSession1-c.txt
01/21/02  09:21a            0 FilesBeforeSession1-d.txt
              10 File(s)              0 bytes
                          102,225,920 bytes free
```

### Session Snap3

1. On the production server, list the files in the **test** directory. The listing shows files created between the start of sessions 2 and 3.

```
F:\Test> dir
...
Directory of F:\Test
01/22/2002  03:25p  <DIR>          .
01/22/2002  03:25p  <DIR>          ..
01/21/2002  09:21a          0 FilesAfterS1BeforeS2-a.txt
01/21/2002  09:21a          0 FilesAfterS1BeforeS2-b.txt
01/21/2002  09:21a          0 FilesAfterS1BeforeS2-c.txt
01/21/2002  09:21a          0 FilesAfterS1BeforeS2-d.txt
01/21/2002  09:21a          0 FilesAfterS2BeforeS3-a.txt
01/21/2002  09:21a          0 FilesAfterS2BeforeS3-b.txt
01/21/2002  09:21a          0 FilesAfterS2BeforeS3-c.txt
01/21/2002  09:21a          0 FilesBeforeSession1-a.txt
01/21/2002  09:21a          0 FilesBeforeSession1-b.txt
01/21/2002  09:21a          0 FilesBeforeSession1-c.txt
01/21/2002  09:21a          0 FilesBeforeSession1-d.txt
              11 File(s)              0 bytes
               2 Dir(s)     102,225,920 bytes free
```

2. On the production server, flush buffers and start the third session, **snap3**.

```
F:\Test> admsnap flush -o f:
F:\Test> admsnap start -s snap3 - o f:
Attempting to start session snap3 on device
PhysicalDrive1.
Attempting to start the session on the entire LUN.
Started session snap3.
F:\Test>
```

3. On the secondary server, flush buffers, deactivate session **snap2**, and activate the third session, **snap3**.

```
15:28:06.96 C:\>  admsnap flush -o f:
Flushed f:.

15:28:13.32 C:\> admsnap deactivate -s snap2
Deactivated session snap2 on device F:.

15:28:20.26 C:\> admsnap activate -s snap3
Scanning for new devices.
Activated session snap3 on device F:.
```

4. On the secondary server, list files to show production server files that existed at session 3 start.

```
15:28:39.96 C:\>   dir  f:\test
...
Directory of F:\Test
01/22/02  03:25p  <DIR>          .
01/22/02  03:25p  <DIR>          ..
01/21/02  09:21a                 0 FilesAfterS1BeforeS2-a.txt
01/21/02  09:21a                 0 FilesAfterS1BeforeS2-b.txt
01/21/02  09:21a                 0 FilesafterS1BeforeS2-c.txt
01/21/02  09:21a                 0 FilesAfterS1BeforeS2-d.txt
01/21/02  09:21a                 0 FilesAfterS2BeforeS3-a.txt
01/21/02  09:21a                 0 FilesAfterS2BeforeS3-b.txt
01/21/02  09:21a                 0 FilesAfterS2BeforeS3-c.txt
01/21/02  09:21a                 0 FilesBeforeSession1-a.txt
01/21/02  09:21a                 0 FilesBeforeSession1-b.txt
01/21/02  09:21a                 0 FilesBeforeSession1-c.txt
01/21/02  09:21a                 0 FilesBeforeSession1-d.txt
              13 File(s)              0 bytes
                             102,225,920 bytes free
15:28:42.92 C:\Test>
```

5. On the secondary server, deactivate the last session.

```
15:28:45.04 C:\> admsnap deactivate -s snap3
```

6. On the production server, stop all sessions.

```
F:\Test> admsnap stop -s snap1 -o f:
F:\Test> admsnap stop -s snap2 -o f:
F:\Test> admsnap stop -s snap3 -o f:
```

# A

# SnapView Snapshot CLI Error Codes

This appendix lists the SnapView snapshot error codes returned by the CLI. The numbers proceed sequentially.

**Note:** The clone CLI commands do not return any of the error codes in this appendix. Clone CLI commands return 0 if the command is successful and 1 if the command fails.

For reserved LUN pool error codes, refer to revision A09 (or higher) of the *EMC Navisphere Command Line Interface (CLI) Reference*.

# SnapView snapshot CLI error codes

| Error value | Description |
|---|---|
| 123 | Snapshot does not exist. |
| 125 | Invalid SP name. |
| 127 | Package number invalid |
| 128 | This version of FLARE Software does not support nondisruptive software installation (NDU). |
| 129 | Cannot open this file. |
| 130 | File does not exist. |
| 131 | RAID type must be **r1** (RAID1), **r3** (RAID 3), **r5** (RAID 5), or **r1_0** (RAID1/0). |
| 132 | Multiple subcommands specified. Check syntax. |
| 133 | Disk for PSM must be on DPE bus 0. |
| 134 | Configuration does not exist. |
| 135 | Configuration already exists. |
| 136 | Size specified is too small. |
| 137 | Configuration does not exist. Run the navicli **initializearray** command to configure the system. |
| 138 | First option must be a subcommand. |
| 139 | Cannot create RAID group for PSM (Persistent Storage Manager). |
| 140 | Name or UID (unique ID) is required. |
| 141 | Invalid name specified. |
| 143 | Name and LUN are required. |
| 144 | Storage system UID (unique ID) and LUN UID are required. |

| Error value | Description |
|---|---|
| 146 | Image not found. |
| 147 | Synchronized rate cannot be changed since input image is primary. |
| 150 | Invalid session. |
| 151 | Session does not exist. |
| 152 | Session is stopping. |
| 153 | Invalid snapshot. |
| 154 | Snapshot does not exist. |
| 155 | The -o option requires “-all” or “-filename.” |
| 156 | Path to store files is required. |
| 157 | Cannot specify both “-all” and “-filename.” |
| 158 | Enter file index or “quit.” |
| 159 | Invalid input. |
| 160 | Index is out of range. |
| 161 | File not found. |
| 162 | Space not available to retrieve file. |
| 163 | Specified feature not supported. |
| 164 | Feature must be specified. |
| 165 | Cannot specify both '-lun' and '-lunuid'. |
| 166 | Invalid storage processor name. |
| 167 | PSM (Persistent Storage Manager) is not broken. |
| 168 | PSM (Persistent Storage Manager) is broken. Cannot list or create PSM. |
| 169 | LUN cannot be unbound. |
| 170 | Operation not supported on this type of storage system. |
| 171 | Incompatible arguments. Invalid storage system serial number. |
| 172 | Directory not specified. |
| 173 | Invalid number of blocks. |

| Error value | Description |
|---|---|
| 174 | Number of blocks not specified. |
| 175 | Reading of data not supported on this storage system. |
| 176 | Invalid snapshot World Wide Name (WWN). |
| 177 | Invalid storage system serial number. |
| 178 | Navicli '-f' option required to store data in file. |
| 179 | Invalid IP address format. |
| 180 | Storage group cannot be shared. |
| 181 | Invalid HLU number. |
| 182 | Invalid ALU number. |
| 183 | Invalid port ID. |
| 184 | Remote server cannot be managed. |
| 185 | Email response test failed. |
| 186 | Email page response test failed. |
| 187 | Modem page response test failed. |
| 188 | Snmp response test failed. |
| 189 | Phone home response test failed. |
| 190 | Mandatory switch for email/emailpage. |
| 191 | Mandatory switch for modem page. |
| 192 | Mandatory switch for snmp. |
| 193 | Only one message or file can be specified. |
| 194 | Valid dial string contains only digits, parentheses, hyphen. |
| 195 | File does not exist or cannot be opened. |
| 196 | Specified user already exists. |
| 197 | The offset switch is not supported for this storage system. |
| 198 | Valid COM port number is 1, 2, 3, or 4. |
| 199 | Valid dial command is atd, atDp or atD. |

| Error value | Description |
|---|---|
| 200 | Valid message delay contains only ",,," (one or more commas). |
| 202 | Target LUN number is missing. |
| 203 | Session name is missing. |
| 204 | Snapview multiple session feature is not supported. |
| 205 | Cannot specify both snapshot name and snapshot ID. |
| 206 | Cannot specify both -mode and -simulation. |
| 207 | This command is not supported on remote server. |
| 208 | Switch -pathname must be specified. |
| 209 | Get local server attributes failed. |
| 210 | This version of FLARE Software does not support Hi5 RAID type. |
| 211 | Only one of the switches -snapshotid, -snapshotname, or -lun can be specified. |
| 212 | Specified session and snapshot must be based on the same source LUN. |
| 215 | The HLU (server LUN) number you specified cannot be found. |
| 216 | This command must be issued from the SP to which the LUN will trespass. |
| 217 | Invalid bus or enclosure number. |
| 218 | Invalid WWN Seed. |
| 219 | Invalid EMC Part Number. |
| 220 | This RAID group has maximum number of LUNs already. |
| 223 | Allocate log needs to specify a LUN with a valid owner. |
| 224 | This request has been issued through the SP that is not the current owner of the targeted LUN. |
| 225 | Simulated/non-simulated sessions are mixed on the same targeted LUN. |
| 262 | This version of FLARE software does not support rollback. |
| 263 | The switch -rate must be specified. |
| 264 | Invalid rollback rate. Valid values are high, medium and low. |
| 265 | Cannot change the rollback rate. The specified session is not rolling back. |

| Error value | Description |
|---|---|
| 266 | Cannot start a rollback operation. The specified session is not persistent. |
| 267 | Cannot start a rollback operation. The specified session is rolling back. |
| 268 | Cannot start a rollback operation. At least one source LUN is involved in another rollback operation. |
| 269 | Cannot create a recovery session. At least one source LUN has maximum sessions already. |
| 270 | Invalid SnapView session name. |
| 271 | Simulation mode is not supported in this revision. |
| 272 | Snapshot cache chunk size cannot be changed in this revision. |
| 273 | Cannot stop a session when it is rolling back. |
| 274 | Cannot activate a snapshot on a session when the session is rolling back. |
| 275 | Cannot deactivate a snapshot when it is activated on a session that is rolling back. |
| 294 | This request must be issued from the SP that owns the session. |
| 295 | This request must be issued from the SP that owns the session that the snapshot is activated on. |
| 296 | Snapshot is already inactive. |
| 297 | Snapshot name with leading or trailing white space is invalid. |
| 298 | Cannot create a snapshot on private LUNs (hot spare, snap cache, PSM, and so on). |
| 352 | Consistent mode is not supported in this revision. |

# B

# Correcting Bad Blocks

This appendix describes what bad blocks are, how SnapView handles them, and what you can do to correct them

Major sections in this appendix are

## Bad blocks overview

A bad block is an unreadable block on the LUN. The unreadable block is due to an incomplete write to the disk. Since there is an incomplete write to the disk, you cannot read the bad block on the LUN.

Although bad blocks are rare, they can occur anywhere on a LUN. If they occur in data or metadata, most operating systems will detect them and log errors, which causes applications to fail. If a bad block occurs in a file system's free space or in a database's free space, the server does not detect the bad block and it is essentially harmless.

# Bad blocks and clones

If a bad block is encountered on the source LUN during a synchronization, SnapView generates a bad block at the same location on the clone LUN. If a bad block is encountered on the clone LUN during a reverse synchronization, SnapView generates a bad block at the same location on the source LUN. SnapView then proceeds with the synchronization or reverse synchronization operation. The LUNs are then a full copy of the source (if a synchronization was issued) or the clone (if a reverse synchronization was issued), which includes the original bad blocks.

SnapView generates a message in the event log to inform the user of the bad blocks on the LUN. After SnapView generates twenty messages, it stops logging bad blocks in the event log, but continues generating bad blocks on the clone LUN if one is encountered on the source LUN during a synchronization, or it continues generating bad blocks on the source LUN if one is encountered on the clone LUN during a reverse synchronization. If SnapView encounters more than 32,708 bad blocks, it aborts the synchronization or reverse synchronization operation.

If the bad blocks occur in a file system's free space or in a database's free space, SnapView detects them during a full synchronization (the initial synchronization) or reverse synchronization. Subsequent partial synchronizations encounter bad blocks only if they occur in a chunk that the server has written to.

## Correcting bad blocks

Although bad blocks in a database's free space may be harmless, they can cause a synchronization or reverse synchronization operation to take longer than usual, in addition to generating excessive log messages.

You can correct a bad block by successfully writing to it. However, writing to it may be impossible if it is in free space. Instead, you can use a server-based utility to back up the data from the LUN with bad blocks. Then reformat or unbind/bind the LUN and restore the data from backup.

## Bad blocks and rollback

If the software encounters a bad block on a reserved LUN during a rollback operation, SnapView generates a bad block on the source LUN. SnapView generates the bad block at the same location the block was supposed to be rolled back.

SnapView generates a message in the event log to inform you of the bad blocks on the source LUN. After twenty messages, SnapView stops logging bad blocks in the event log, but it continues generating bad blocks on the source LUN until the rollback operation is complete.

### Correcting bad blocks

Although bad blocks in a database's free space may be harmless, they can cause a rollback operation to take longer than usual, in addition to generating excessive log messages.

You can correct a bad block by successfully writing to it. However, writing to it may be impossible if it is in free space. Instead, you can use a server-based utility to back up the data from the LUN with bad blocks. Then reformat or unbind/bind the LUN and restore the data from backup.

# Glossary

This glossary contains terms related to SnapView. Many of these terms are used in this manual.

## A

**Active** A snapshot that is currently participating in a SnapView session and is accessible to secondary servers.

**Activate** An operation on a snapshot that maps it to a SnapView session. This feature is available in Navisphere Manager, admsnap, and the CLI.

**admsnap** Server-based software that provides a command line interface to SnapView software running in a storage-system SP. With admsnap, you can start and stop sessions and activate and deactivate snapshots by typing commands on a secondary server system.

## C

**Chunk** An aggregate of multiple disk blocks that SnapView uses to perform copy-on-first-write operations. The selectable chunk sizes are 16 KB, 32 KB, 64 KB, 128 KB, 256 KB, and 512 KB. The default size is 64 KB (128 blocks in Navisphere). For SnapView version 2.1 or higher, the chunk size is set to 64K (128 blocks). You cannot change this value.

**CLI** EMC Navisphere Command Line Interface. SnapView uses two CLIs: a clone CLI and a snapshot CLI.

**Client** A server (computer or laptop) that has an Internet browser and connects to a storage application server with a network. You use it to manage connected storage systems.

**Clone** A LUN that is an actual copy of a specified source LUN. The state of the clone determines if it is a byte-for-byte copy of its source. You create a clone when you add a clone to the clone group.

**Clone group** A collection of a source LUN and all of its clones. The purpose of creating a clone group is to establish a source LUN that you may want to clone at some time.

**Clone private LUNs** LUNs that record information that identifies areas on the source and clone that have changed since the clone was fractured. A log in the clone private LUN records this information but no actual data is written to the clone private LUN. This log is a bitmap and reduces the time it takes to synchronize and reverse synchronize a clone and its source.

**Clone state** The state of each clone in a clone group. The state of the clone determines whether or not the clone is usable. The possible clone states are consistent, out-of-sync, remote mirror synchronizing, reverse out-of-sync, reverse synchronizing, synchronized, or synchronizing.

**Consistent fracture** Fractures one clone at a time per source LUN across multiple source LUNs. The clones you want to fracture must be within different clone groups. You cannot perform a consistent fracture on clones belonging to different storage systems. After the consistent fracture completes, there is no group association between the clones.

**Consistent mode** A session mode where the point-in-time copy across a set of source LUNs is preserved. The SnapView driver will delay any I/O requests to the set of source LUNs until the session has started on all LUNs (thus preserving the point-in-time on the entire set of LUNs).

**Consistent state** A clone in a synchronized state that receives server I/O to the source (if the clone is unfractured) or to the clone (if the clone is fractured). A consistent clone is usable but may not contain the most up-to-date information since writes made to the source have not been copied to the clone.

**Copy-on-first-write** An algorithm that copies current contents of a source LUN before it is modified (written to). The copy-on-first-write operation is on a chunk: before the first modification of any disk blocks within a chunk, the software reads and stores the original data of the chunk in the reserved LUN pool. This policy applies only to the first modification of the data. Overwrite of any data that has already had a copy-on-first-write does not require any extra processing since the software saved the original data in the reserved LUN pool.

## D

**Deactivate** An operation on a snapshot that unmaps it from a SnapView session to make it invisible to any secondary servers. The software destroys any writes made to the snapshot but the snapshot and session still exist. This feature is available in Navisphere Manager and admsnap, however, the Manager deactivate function does not flush all data and clear all buffers on the secondary server.

## F

**Fracture** The process of breaking off a clone from its source. Once a clone is fractured, it can receive server I/O requests.

## H

**Host agent** EMC Navisphere agent that runs on a server system.

**Host** See server.

## I

**Inactive** A snapshot that is not currently participating in a SnapView session and is invisible to any secondary servers.

## M

**Modified data chunk** A chunk of data that a server changes by writing to the clone, snapshot, or source LUN.

## N

**Navisphere Manager** The EMC Navisphere Manager application.

## O

**Out-of-sync state** A clone that was in the process of synchronizing but failed. An out-of-sync clone is not a byte-for-byte copy of its source LUN and therefore, is unusable.

## P

**Persistence mode** Creates a session that can withstand an SP reboot or failure, a storage system reboot or power failure, or server I/O trespassing to the peer SP. Admsnap and Navisphere CLI version 6.24 or later enable all SnapView sessions to run in persistence mode as the default.

**Point-in-time** The moment a SnapView session starts.

**Private LUN** A LUN that cannot be assigned to a storage group. Once you add a LUN to the reserved LUN pool or allocate a LUN as a clone private LUN, it becomes a private LUN.

**Protected restore** When selected, a process that prevents source writes from being copied to the clone during a reverse synchronization.

## Q

**Quiesce threshold** The time period after which, without I/O from the server, any clone in the consistent state and not fractured is transitioned to a synchronized state. You specify the quiesce threshold when you create a clone group.Valid values are 10 – 3600 seconds. The default is 60 seconds.

## R

**Recovery policy** The policy used to determine how a clone is recovered after a failure. Options are **auto** or **manual**.

**Remote mirror synchronizing state** A clone that is not a point-in-time copy of its source because the source (a MirrorView secondary image) is in the synchronizing or out-of-sync state.

**Reserved LUN** A private LUN (a LUN to which a server cannot perform I/O) assigned to the reserved LUN pool.

**Reserved LUN pool** The disk storage used to store blocks of original data chunks when you first modify that chunk on the source LUN(s) after the start of a

session. Each SP manages its own LUN pool space and assigns a separate reserved LUN (or multiple LUNs) to each source LUN.

**Reserved sessions** Sessions used for another application such as SAN Copy and MirrorView/Asynchronous.

**Reserved snapshots** Snapshots used for another application such as SAN Copy and MirrorView/Asynchronous.

**Restartable copy** A data state having dependent write consistency and where all internal database/application control information is consistent with a database management system/application image.

**Reverse out-of-sync state** A clone that was in the process of reverse synchronizing but failed. Therefore, the source LUN is unusable and another reverse synchronization is recommended.

**Reverse synchronizing state** A clone that is unfractured and in the process of copying its data to its source LUN.

**Rollback** Restores the point-in-time copy of a SnapView session to the source LUN(s).

## S

**Server** In the context of disk-array storage systems, a processor that runs an operating system and uses a disk-array storage system for data storage and retrieval.

**Snapshot** A point-in-time image of a source LUN(s). A snapshot occupies no disk space, but appears like a normal LUN to secondary servers and can serve for backup or another use.

Other, older terms for snapshot, which are no longer used, include **SnapshotCopy LUN (SCLUN)** and **SnapCopy LUN (SLU)**.

**SnapView** Software that allows you to obtain a copy of a LUN by creating a clone or snapshot. The clone or snapshot can serve for backup, decision support scenarios, or as a base for temporary operations on the production data without damaging the original data on the source LUN.

**SnapView session** The period of time that SnapView is managing a reserved LUN pool region. The session begins when you start a session using Navisphere Manager, Navisphere CLI, or admsnap and ends when you stop the

session. You can give each session a name (the session name) when you start the session. The name persists throughout the session and is viewable through Navisphere. You use the name to check session status and to end the session.

**Source LUN** The original LUN from which a clone or snapshot is generated. An older term for source LUN, which is no longer used, is **Target LUN (TLU)**.

**SP agent** The Navisphere agent that runs in an SP (CX4 series, CX3 series, or CX series storage systems).

**Synchronization rate** Specifies a relative value (low, medium, or high) for the priority of completing updates. **High** completes updates faster, but may significantly affect storage-system performance for host I/O requests. **Low** completes updates slower, but also minimizes the impact on other storage-system operations.

**Synchronized state** A clone that is a byte-for-byte copy of its source and, therefore, is usable.

**Synchronizing state** An unfractured clone that is in the process of copying data from its source LUN.

# Index

## D



## E



## F



## I



## L



## M

## N



## O



## P



## Q



## R



## S

## T



## U



## V