Copyright © 2004-2006 EMC Corporation. All rights reserved.
Published August, 2006

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

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

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

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

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

Chapter 1 About EMC MirrorView/A CLI
MirrorView/A overview ................................................................. 1-2
Prerequisites .................................................................................. 1-3
Configuration guidelines ............................................................. 1-4
  Sample configuration ................................................................. 1-5
MirrorView/A connection requirements ..................................... 1-7
  Cable connections between SPs at the MirrorView/A sites ...... 1-7
MirrorView/A terminology .......................................................... 1-11
MirrorView/A features and benefits ........................................... 1-14
  Provision for disaster recovery with minimal overhead .. 1-14
  CLARiiON MirrorView/A environment .................................... 1-14
  Bidirectional mirroring ......................................................... 1-15
  Integration with EMC SnapView snapshot software ....... 1-15
  Integration with EMC SAN Copy software ....................... 1-15
  Replication over long distances .......................................... 1-16
  Application integration ......................................................... 1-16

Chapter 2 MirrorView/A (remote mirroring) commands
MirrorView/A operations overview.............................................. 2-3
About Secure CLI........................................................................... 2-6
Getting started with Secure CLI .................................................. 2-8
  Overview of using the Navisphere 6.X security with Secure CLI......................................................... 2-8
  naviseecli ................................................................................ 2-10
About Java CLI............................................................................... 2-15
Requirements for using Java CLI ................................................ 2-15
Overview of using the Navisphere 6.X security with Java CLI ................................................ 2-15
navicli.jar ......................................................................................... 2-17
MirrorView/A CLI functions ....................................................... 2-20
mirror -async -addimage .............................................................. 2-22
mirror -async -change ................................................................... 2-26
mirror -async -changeimage ......................................................... 2-28
mirror -async -create................................................................. 2-31
mirror -async -destroy.............................................................. 2-33
mirror -disablepath ........................................................................ 2-35
mirror -enablepath ....................................................................... 2-37
mirror -async -fractureimage ...................................................... 2-39
mirror -async -info ......................................................................... 2-41
mirror -async -list ........................................................................... 2-43
mirror -async -promoteimage .................................................... 2-46
mirror -async -removeimage ........................................................ 2-49
mirror -async -setfeature ............................................................. 2-51
mirror -async -syncimage ........................................................... 2-53

Chapter 3  MirrorView/A (consistency group) commands

mirror -async -creategroup ............................................................. 3-2
mirror -async-destroygroup ........................................................... 3-5
mirror -async -addtogroup ............................................................ 3-7
mirror -async -removefromgroup .................................................... 3-9
mirror -async -changegroup .......................................................... 3-11
mirror -async -syncgroup ............................................................ 3-14
mirror -async -fracturegroup ....................................................... 3-16
mirror -async -promotegroup ...................................................... 3-18
mirror -async -listgroups ............................................................. 3-20

Appendix A  Responding to failures

How MirrorView/A handles failures ............................................ A-2
Access to the primary SP fails ................................................. A-2
Primary image fails ................................................................. A-2
Failure of the secondary image .......................................... A-8
Promoting a secondary image when there is no failure ... A-9
Summary of MirrorView/A failures .................................. A-10
Recovering from serious errors ........................................... A-12
How consistency groups handle failures ............................ A-13
Access to the SP fails ............................................................. A-13
Primary storage system fails ................................................. A-13
Appendix B  Navicli command

navicli ....................................................................................................................... B-2
Using navicli with MirrorView/A commands ................................. B-5

Appendix C  MirrorView/A command error codes

MirrorView/A command error codes ............................................... C-2

Index .............................................................................................................................................. i-1
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 manual 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 CLI commands you use to set up, configure, and manage EMC® MirrorView™/Asynchronous software. Each major section includes introductory and format information.

This manual refers to EMC MirrorView/Asynchronous as MirrorView/A.

Audience

This manual is intended for those who will use the naviseclli or navcli.jar command to create and manage remote mirrors on CX3-series and CX-series storage systems with the MirrorView/Asynchronous option. Readers of this guide should be familiar with the following topics:

- The operating system running on the servers you will manage.
- Storage-system components and configurations.
### Preface

#### Organization

This manual contains three chapters and three appendixes, as follows.

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 1</td>
<td>Introduces the MirrorView/Asynchronous software and terminology.</td>
</tr>
<tr>
<td>Chapter 2</td>
<td>Explains the remote mirror CLI commands for managing the MirrorView/A software.</td>
</tr>
<tr>
<td>Chapter 3</td>
<td>Explains the CLI commands for creating and managing consistency groups.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix A</td>
<td>Describes how MirrorView/A handle failures.</td>
</tr>
<tr>
<td>Appendix B</td>
<td>Explains the <code>navicli</code> command.</td>
</tr>
<tr>
<td>Appendix C</td>
<td>Lists the MirrorView error codes you might receive from CLI commands.</td>
</tr>
</tbody>
</table>

#### Related documentation

Related documents include:

- *EMC MirrorView/Asynchronous for Navisphere Administrator’s Guide* (P/N 300-001-333)

For CX3-series and CX-series storage systems, refer to the latest version of the *EMC Navisphere Command Line Interface (CLI) Reference* (P/N 300-003-628).

For the most current management and security content for CX3-series and CX-series storage systems, refer to *EMC Navisphere Manager Administrator’s Guide* (P/N 300-003-511).

#### Conventions used in this guide

EMC uses the following conventions for notes, cautions, warnings, and danger notices.

**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. The caution may apply to hardware or software.

---

**WARNING**

A warning contains information essential to avoid a hazard that can cause severe personal injury, death, or substantial property damage if you ignore the warning.

---
DANGER

A danger notice contains information essential to avoid a hazard that will cause severe personal injury, death, or substantial property damage if you ignore the message.

Typographical conventions
This manual uses the following format conventions:

This typeface Indicates text (including punctuation) that you type verbatim, all commands, pathnames, 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 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.

This typeface Represents a system response (such as a message or prompt), a file or program listing.

[ ] Encloses optional entries.

| Separates alternative parameter values; for example:
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 EMC MirrorView/Asynchronous CLI is posted on the EMC Powerlink™ website. We recommend that you download the latest information before you run the CLI commands.

To access EMC Powerlink, use the following link:

http://Powerlink.EMC.com

After you log in, select Support > Document Library and find the following:

✦ EMC MirrorView/Asynchronous Release Notes
✦ The latest version of this manual that is applicable to your software revision
EMC Installation Roadmap for CX3-Series, CX-Series, AX-Series, and FC-Series 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 connect 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 a message to techpub_comments@EMC.com with your opinions of this guide.
This chapter introduces the EMC® MirrorView™/Asynchronous software and the EMC Navisphere® storage-system management configurations and architecture.

This manual refers to the EMC MirrorView / Asynchronous product as MirrorView/A.

Note: If you are already familiar with MirrorView / A from reading Chapter 1 in the EMC MirrorView/Asynchronous for Navisphere Administrator’s Guide, you can skip to Chapter 2.

Major topics are:

- MirrorView/A overview ................................................................. 1-2
- Prerequisites .................................................................................. 1-3
- Configuration guidelines ............................................................... 1-4
- MirrorView/A connection requirements ........................................ 1-7
- MirrorView/A terminology ............................................................ 1-11
- MirrorView/A features and benefits .............................................. 1-14
MirrorView/A overview

MirrorView/A lets you periodically update a remote copy of production data. It is a software application that keeps a point-in-time copy of a logical unit (LUN) and periodically replicates the copy to a separate location in order to provide disaster recovery, that is, to let one image continue to be active if a serious accident or natural disaster disables the other. It provides data replication over long distances (hundreds to thousands of miles).

The production image (the one mirrored) is called the primary image; the copy image is called the secondary image. MirrorView/A supports one remote image per primary. The primary image receives I/O from a server called the production server; a separate storage system maintains the secondary image. This storage system can optionally have a failover/standby computer connected to it or can be connected to its own computer system. Both storage systems can be in different domains, which you manage with the user interface (UI). The client that is managing the storage system can promote the secondary image if the primary image becomes inaccessible. After initial synchronization, the remote site always has a consistent point-in-time copy of the primary data.

**Note:** As a storage-system-based mirroring product, MirrorView/A does not flush server buffers before replicating the primary data. Therefore, the copy is a crash-consistent image of the primary data. You must verify data integrity of the secondary image before using it for disaster recovery. (The verification process varies by application type.)

MirrorView/A supports MirrorView/A consistency groups, which this manual refers to as consistency groups. A consistency group is a set of asynchronous mirrors whose secondary images need to be kept consistent with each other in order to be useful; that is, the data on the set of secondary images must have existed on the set of primary images previously. This allows an application to use the secondary images if the primary storage system fails.

**Note:** The primary images of mirrors in a MirrorView/A consistency group must reside on a single storage system, and the secondary images of the mirrors must reside on a single (but different) storage system. This contrasts with volumes in a Symmetrix® consistency group, which can reside on multiple storage systems.
Prerequisites

- You must have the MirrorView/A and Access Logix™ software installed and enabled on all CX3-series and CX-series storage systems you want to participate in a mirror. See the chapter on installing software on these storage systems in the *EMC Navisphere Manager Administrator’s Guide*.

- You must have Navisphere Manager installed and enabled.

- Storage area network (SAN) configurations must have qualified switches.

- Wide area network (WAN) configurations must have qualified FC-to-IP devices.
Configuration guidelines

The following are configuration rules for MirrorView/A:

- Each mirror must have one primary image and zero or one secondary images. Any single storage system can have only one image of a mirror.

- A storage system can have concurrent mirroring connections to a maximum of four other storage systems. (Mirroring connections are common between synchronous and asynchronous mirrors.)

- You can configure a maximum of 50 primary and secondary images on CX400, CX500, and CX3-20 storage systems and a maximum of 100 primary and secondary images on CX600, CX700, CX3-40, and CX3-80 storage systems. The total number of primary and secondary images on the storage system make up this maximum number.

  **Note:** A metaLUN is a single entity; therefore, it is counted as one of your images. For example, if a mirrored metaLUN is composed of five components, it is counted as one of your images, not five.

  These limits are independent of the limits for synchronous mirrors. (See the *EMC MirrorView/Synchronous for Navisphere Administrator’s Guide*).

- To manage remote mirror configurations, the Navisphere management workstation must have an IP connection to both the local and remote storage systems. The connection to the remote storage system should have an effective bandwidth of at least 128 Kb/second. The storage systems can be in different Navisphere domains, which you manage with the UI (see the *EMC Navisphere Manager Administrator’s Guide*).

- MirrorView/A supports the new, larger LUNs that the FLARE™ operating environment supports. (See the FLARE release notes.)
Sample configuration

Figure 1-1 shows a sample remote mirror configuration with two sites and a primary and secondary image that includes the database of four LUNs.

Figure 1-1  Sample remote mirror configuration

In Figure 1-1, database server 1, the production server, executes customer applications. These applications access data on storage system 1, in the database server storage group. Storage system 2 mirrors the data on the database server storage group.
**Note:** Storage groups are different than consistency groups, which are described in Chapter 3. For more information about storage groups, see the EMC Navisphere Manager Administrator’s Guide.

We recommend that you attach a secondary server to storage system 2, so that if a complete site failure occurs where storage system 1 and database server 1 are located, you can completely fail over to the secondary site, and thus minimize the outage window. The server at the standby site is not required; but because it is recommended, this example includes it in the overall configuration.

Each server has a path to each SP through each fabric — to each storage system. If a failure occurs in a path, software installed on the server (for example, EMC PowerPath® software) can switch to the path through the other SP and continue accessing the data, transparent to the applications on the server.

The production server sends a write request to an SP in storage system 1, which then writes data to the local LUN. The change to the primary LUN is recorded, and at an interval that you define, all changes are copied to the secondary storage system.

If a failure occurs in storage system 1, an administrator can use the client that is managing the storage system to promote the image on storage system 2 to the role of primary image.

**Note:** The mirrored data is inaccessible until the secondary image is promoted to a primary image.

Then the appropriate applications can start on any connected server (here, database server 2) with full access to the data. The mirror can be accessible in minutes, although the time needed for applications to recover will vary.
MirrorView/A connection requirements

MirrorView/A requires the following:
- One server, connected to one of the storage systems (a second server, connected to the other storage system, is optional).
- A Fibre Channel connection between the two storage systems (direct, switch, or IP connection, shown on the following pages).

Cable connections between SPs at the MirrorView/A sites

MirrorView/A use a front-end port on each storage processor (SP) as a communication channel between the storage systems in a remote mirror configuration. This port is called the mirror port in this document. The mirror ports for the different storage systems are:
- Port 5 for a CX3-20c or CX3-40c SP
- Port 3 for a CX600, CX700, or CX3-80 SP
- Port 1 for a CX400, CX500, CX3-20, or CX3-40 SP

Although server I/O can share the front-end port with MirrorView/A, for performance reasons, we strongly recommend that server I/O use the front-end ports that MirrorView/A is not using.

CAUTION
Currently, MirrorView and SAN Copy software cannot share the same SP port. Before installing the MirrorView enabler, you must deselect any MirrorView ports that a SAN Copy session is using. Otherwise, any SAN Copy sessions using the MirrorView port will fail.

For MirrorView/A to work correctly, the SP A mirror port at one site must be connected to the SP A mirror port on the other site and the SP B mirror port at one site must be connected to the SP B mirror port at the other site. The connections can be either direct or through a switch fabric.
**Direct remote mirror connections**

A direct mirror configuration consists of one primary storage system and one secondary storage system. The remote mirror connections must be between:

- SP A mirror ports on the primary and secondary storage systems
- SP B mirror ports on the primary and secondary storage systems

Figure 1-2 shows a sample direct remote mirror configuration.
Fabric remote mirror connections

A fabric mirror configuration consists of one primary storage system and up to four secondary storage systems. The fabric connections must be as follows:

- SP A mirror port on the primary storage system must be connected to the same switch fabric as the SP A mirror port on the secondary storage system.
- SP B mirror port on the primary storage system must be connected to the same switch fabric as the SP B mirror port on the secondary storage system.

Note: The fabric to which SP A mirror ports are connected can be the same fabric or a different fabric than the one to which the SP B mirror ports are connected.

You must zone the mirror port switch connections as follows:

- A separate zone for the SP A mirror port on the primary storage system and the SP A mirror port on each secondary storage system.
- A separate zone for the SP B mirror port on the primary storage system and the SP B mirror port on each secondary storage system.

For example, if you have primary storage system 1 and secondary storage systems 2 and 3, you need the following two zones:

- Zone 1 - SP A mirror port on storage system 1 and SP A mirror ports on storage systems 2 and 3.
- Zone 2 - SP B mirror port on storage system 1 and SP B mirror ports on storage systems 2 and 3.

You can use the same SP port for server data and MirrorView/A. Be careful when an IP distance connection is used because using the same SP port may cause a degradation in both replication and server application performance.

**CAUTION**

MirrorView and SAN Copy software cannot share the same SP port. Before installing the MirrorView enabler, you must deselect any MirrorView ports that a SAN Copy session is using. Otherwise, any SAN Copy sessions using the MirrorView port will fail.
Figure 1-3 shows a sample remote mirror fabric configuration.

**Figure 1-3** Sample remote mirror connection through a fabric
## MirrorView/A terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active state</td>
<td>State in which a remote mirror is running normally.</td>
</tr>
<tr>
<td>Attention state</td>
<td>The mirror’s secondary image is fractured, and the mirror is configured to generate an alert in this case. The mirror continues to accept server I/O in this state.</td>
</tr>
<tr>
<td>Auto recovery</td>
<td>Option to have synchronization start as soon as a system-fractured secondary image is determined to be reachable.</td>
</tr>
<tr>
<td>Consistency group</td>
<td>A set of asynchronous mirrors that are managed as a single entity and whose secondary images always remain in a consistent and recoverable state with respect to their primary image and each other.</td>
</tr>
<tr>
<td>Consistency group condition</td>
<td>Displays more detailed information about the consistency group, including whether the group is normal, initializing, updating, admin fractured, system fractured, or waiting on admin.</td>
</tr>
<tr>
<td>Consistency group state</td>
<td>Indicates the current state of the consistency group: Synchronized, Consistent, Quasi-Consistent, Synchronizing, Out-of-Sync, Scrambled, Rolling Back, or Empty.</td>
</tr>
<tr>
<td>Consistent state (of image)</td>
<td>State in which a secondary image is identical to either the current primary image or to some previous (consistent point-in-time) instance of the primary image.</td>
</tr>
<tr>
<td>Fracture</td>
<td>A condition in which I/O does not propagate to the secondary image and can result when you initiate the fracture (Admin Fracture) or when the system determines that the secondary image is unreachable (System Fracture). An admin fracture may also occur if the MirrorView/A software detects an error condition that requires administrative intervention to correct.</td>
</tr>
<tr>
<td>Image condition</td>
<td>The condition of a secondary image provides additional information about the status of updates for the image. Values include normal, administratively fractured, system fractured, updating, synchronizing, or waiting on admin.</td>
</tr>
<tr>
<td>Image state</td>
<td>Indication of the relationship between a secondary image and the primary image of a mirror. The image states are: Synchronized, Consistent, Synchronizing, Out-of-Sync, and Rolling Back. See States.</td>
</tr>
</tbody>
</table>
### MirrorView/A mirroring

A feature that provides disaster recovery by maintaining mirrors of LUNs on other storage system(s). MirrorView/A can work in conjunction with, but is independent of, the other major CLARiiON® software options such as PowerPath® software. MirrorView/A works with LUNs in SAN storage systems; thus you can use it to mirror one or more LUNs.

### Out-of-Sync state

A remote mirror state in which the software does not know how the primary and secondary images differ; therefore a full synchronization is required to make the secondary image(s) usable for recovery. Also see Image state.

### Primary image

The LUN on the production storage system that contains user data and is the source for data copied to the secondary image. There is one primary image and zero or one secondary image. A remote mirror is ineffective for recovery unless it has a secondary image. This manual also refers to primary image as primary.

### Promote (to primary)

The operation by which the administrator changes an image’s role from secondary to primary. As part of this operation, the previous primary image becomes a secondary image. If the previous primary image is unavailable when you promote the secondary image (perhaps because the primary site suffered a disaster), the software does not include it as a secondary image in the new mirror. A secondary image can be promoted if it is in either the Synchronized state or the Consistent state and an update is not currently transferring data.

### Recovery time objective

A time interval you set for MirrorView/A to automatically update the secondary image. You can specify it as the time interval between the start of two consecutive updates or between the end of the last update and the beginning of the next update.

### Remote mirror

A remote mirror is the combination of a LUN on one storage system, called the primary image, and another LUN on a different storage system, called the secondary image. The software maintains the secondary image as an exact copy of the primary image at some (possibly previous) point in time. If the server and/or storage system at the primary site fails, you can promote the secondary image to take over the role of the primary, thus allowing continued access to your production data.
**Remote mirror image**
The LUN on one storage system that participates in a remote mirror. The image can be either the primary or a secondary image. This manual refers to remote mirror image also as *image*.

**Remote mirror state**
Active and attention. In the active state, the remote mirror is running normally. In the attention state, a site or communications failure has caused several accessible images to fall below the required minimum. Administrator action is required to correct the issue with a secondary image.

**Rolling back state**
The state that occurs after a successful promotion where there was an unfinished update to the secondary image. This state persists until the rollback operation completes.

**Secondary image**
A LUN that contains a mirror of the primary image LUN. There can be zero or one secondary image. This manual also refers to *secondary image* as *secondary*.

**States**
Remote mirror states and image states. The remote mirror states are: Active and Attention. The image states are: Synchronized, Consistent, Synchronizing, Out-of-Sync, and Rolling Back.

**Synchronized state**
The state in which the data in the secondary image is identical to that in the primary. On the next write to the primary, the image state will change to Consistent. Also see *States*.

**Synchronizing state**
A secondary image in the process of initial synchronization. The data in the secondary image is not usable for recovery until the synchronization operation completes. Thus, you cannot promote an image in the Synchronizing state to the primary image. Once initial synchronization completes, a MirrorView/A mirror should not return to this state, but stay consistent or synchronized. Also, see *States*.

**Unfinished update**
Cases when the secondary image is fractured and data is not currently being transferred, as well as when data is being transferred.

**Update in progress**
Data is currently being transferred to the secondary storage system.
About EMC MirrorView/A CLI

**MirrorView/A features and benefits**

MirrorView/A mirroring has the following features:
- Provision for disaster recovery with minimal overhead
- CLARiiON environment
- Bidirectional mirroring
- Integration with EMC SnapView™ LUN copy software
- Integration with EMC SAN Copy™ software
- Replication over long distances
- Application integration

**Provision for disaster recovery with minimal overhead**

Provision for disaster recovery is the major benefit of MirrorView/A mirroring. Destruction of the data at the primary site would cripple or ruin many organizations. After a disaster, MirrorView/A lets data processing operations resume with minimal overhead. MirrorView/A enables a quicker recovery by creating and maintaining a copy of the data on another storage system.

MirrorView/A is transparent to servers and their applications. Server applications do not know that a LUN is mirrored and the effect on performance is minimal.

With MirrorView/A, secondary systems are periodically updated. MirrorView/A is not server-based; therefore it uses no server I/O or CPU resources. The processing for mirroring is performed on the storage system.

**CLARiiON MirrorView/A environment**

MirrorView/A operates in a highly available environment, leveraging the dual-SP design of CLARiiON systems. If one SP fails, MirrorView/A running on the other SP will control and maintain the mirrored LUNs. If the server is able to fail over I/O to the remaining SP, then periodic updates will continue. The high-availability features of RAID protect against disk failure, and mirrors are resilient to an SP failure in the primary or secondary storage system.
Bidirectional mirroring

A single storage system may be primary (that is, hold the primary image) for some mirrors and secondary (that is, hold the secondary image) for others. This enables bidirectional mirroring.

**Note:** A storage system can never hold more than one image of a single mirror.

Integration with EMC SnapView snapshot software

SnapView software lets you create a snapshot of an active LUN at any point in time; however, do this only when the mirror is not updating to the secondary image. Since the secondary image is not viewable to any servers, you can use SnapView in conjunction with MirrorView/A to create a snapshot of a secondary image on a secondary storage system to perform data verification and run parallel processes, for example backup.

**Note:** Before starting a SnapView session, make sure that the secondary image is in the synchronized or consistent state. Starting a SnapView session of a secondary LUN when MirrorView/A is updating the secondary will not give consistent data. Also note that data cached on the production server, as well as data written to the primary image but waiting to be transferred to the secondary on the next update, will not be included in the session of the secondary image.

You can create a snapshot of a mirrored LUN, but you cannot create a clone (BCV) of a mirrored LUN.

For more information about SnapView, see the **EMC SnapView for Navisphere Administrator’s Guide**.

Integration with EMC SAN Copy software

SAN Copy™ software lets you create an intra- or inter-storage system copy of a LUN at any point in time; however, do this only when the image state is either Synchronized or Consistent and the mirror is not updating to the secondary image. The copy is a consistent image that can serve for other application purposes while I/O continues to the source LUN. The MirrorView secondary image is not viewable to any servers, but you can use SAN Copy to create a copy of the secondary
image on a secondary storage system to perform data verification and run parallel processes.

**Note:** Related to the process of making the MirrorView secondary unavailable for server I/O, you cannot run SAN Copy full copy sessions on MirrorView secondary images. You can, however, run SAN Copy incremental sessions on MirrorView secondary images.

For more information about SAN Copy, see the *EMC SAN Copy for Navisphere Administrator’s Guide*.

---

**Replication over long distances**

MirrorView/A uses FC-to-IP devices to provide replication over long distances (hundreds to thousands of miles).

---

**Application integration**

Using the Navisphere commands, you can script application integration. You can do this by setting the mirror to the manual update type and issuing a `syncimage` command from the CLI. To ensure that the application data is in a consistent state, put the application (for example, a database) in backup mode, make sure any data buffered on the server is flushed to the storage system, and issue the MirrorView/A `syncimage` command. After the update has started, you can resume the application activities. These application integration processes allow MirrorView/A to generate a secondary that has a consistent state image of the primary and, thus you can readily use it for backup or application testing.
This chapter explains the `navisecli` and `navicli.jar` commands and the commands for creating and managing remote mirrors on CX3-series and CX-series storage systems with the MirrorView/A option. These commands let you use MirrorView/A software to create a byte-for-byte copy of one or more local LUNs connected to a distant storage-system server.

**Note:** The commands in this chapter function only with a storage system that has the optional MirrorView/A software installed.

**Topics are:**
- MirrorView/A operations overview .................................................. 2-3
- About Secure CLI ................................................................................. 2-6
- Getting started with Secure CLI ....................................................... 2-8
- navisecli ........................................................................................... 2-10
- About Java CLI ................................................................................. 2-15
- Requirements for using Java CLI ................................................... 2-15
- navicli.jar ........................................................................................... 2-17
- MirrorView/A CLI functions ......................................................... 2-20
- mirror -async -addimage ................................................................. 2-22
- mirror -async -change ...................................................................... 2-26
- mirror -async -changeimage ........................................................... 2-28
- mirror -async -create ...................................................................... 2-31
- mirror -async-destroy .................................................................... 2-33
- mirror -disablepath .......................................................................... 2-35
- mirror -enablepath ........................................................................... 2-37
- mirror -async -fractureimage .......................................................... 2-39
- mirror -async -info ........................................................................... 2-41
- mirror -async -list ............................................................................. 2-43
- mirror -async -promoteimage .......................................................... 2-46
MirrorView/A (remote mirroring) commands

- mirror -async -removeimage ....................................................... 2-49
- mirror -async -setfeature ......................................................... 2-51
- mirror -async -syncimage ......................................................... 2-53
MirrorView/A operations overview

1. Connect the same Navisphere client that is managing the storage system to both storage systems and configure this client, so that you can manage both storage systems.

   **Note:** Optionally, you can perform this step at the secondary site as well.

2. Estimate the size of the reserved LUN pool and the reserved LUNs (see the *EMC Navisphere Command Line Interface (CLI) Reference*).

3. Establish a usable, two-way connection between the MirrorView/A storage systems (see “mirror-enablepath” on page 2-37).

4. If the primary LUN does not exist, bind it on its server’s storage system. Wait for the LUN to finish binding and add it to the storage group.

5. If the secondary LUN does not exist, create a secondary image LUN.

   **Note:** The secondary LUN can be a different RAID type from the primary, as long as the block size matches.

6. Wait for the secondary LUN to finish binding.

7. From the management workstation, create the remote mirror (see “mirror-async-create” on page 2-31).

   Next add a secondary image. To add a secondary image, see “mirror-async-addimage” on page 2-22.

   Normally, when you add a secondary image to a mirror, the software synchronizes the secondary image with the primary. The software will initially copy all the data from the primary LUN to the secondary LUN, in order to ensure that the two LUNs are identical.

   At any time in the previous sequence of steps, you can get remote mirror status with the `mirror-async-list` command.
8. If a primary failure occurs, Navisphere reports the failure.

If the primary failure is minor, have the primary fixed and resume mirroring.

If the primary failure is catastrophic, the original client that is managing the storage system may be unusable and thus unable to report the failure. For such a failure, the administrator at the secondary site must set up a client to manage the storage system (if not already done), and then promote the secondary to primary and take other recovery action needed. This includes assigning the newly promoted LUN to a storage group, if it is not already the appropriate one.

When you use MirrorView/A on a VMware ESX Server, after you promote the secondary image to a primary, perform the following steps:

a. If not already assigned, assign the newly promoted primary image to a storage group of the same or standby ESX Server.

b. Rescan the bus at the ESX Server level.

c. If not already created, create a Virtual Machine (VM) on the same or standby ESX Server. The VM is not powered up when you create it.

d. Assign the newly promoted primary to the VM.

e. Power up the VM.

If the VM is created and running and you have not already assigned the newly promoted primary to the VM, perform these steps:

a. Perform steps a and b above.

b. If you are running ESX Server 2.5.x, power it down.

   ________________
   Note: If you are running ESX Server 3.x, you do not need to power down the VM.

   ________________
   c. To assign the newly promoted primary to the VM, use the Virtual Center interface for ESX Server 3.x and 2.5.x or the Management User Interface for ESX Server 2.5.x.
d. If you are running ESX Server 2.5.x, power up the VM.

**Note:** If you are running ESX Server 3.x, you do not need to power up the VM.

The primary image (which is now the secondary image) will not be accessible to the primary ESX Server.

**Note:** For configuration restrictions when using VMFS volumes, go to the E-Lab Interoperability Navigator on the EMC Powerlink website, and under the **PDFs and Guides** tab, open the **VMware ESX server.pdf** file.

9. If access to the secondary storage image fails, the primary storage system will fracture the remote mirror. If the problem with the secondary is minor (for example, replacing a cable), then the administrator can fix it. Mirroring will recover and resynchronize the image, if the image recovery policy is Automatic.

10. Create a consistency group and add the primary image to it (optional). See “mirror -async -creategroup” on page 3-2.

Whenever you want to stop mirroring, you can first fracture and remove the secondary images, and then destroy the mirror. This does not affect any data on either image LUN, and access to the primary LUN is also unaffected. The LUN that held the secondary image now becomes accessible as a regular LUN.
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 combines key features of the existing Classic (see “Using navicli with MirrorView/A commands” on page B-5) and Java CLI functionalities (see “About Java CLI” on page 2-15). Similar to Classic CLI, you do not need to install a JRE to run Secure CLI. Secure CLI implements the security features of Java CLI, providing role-based authentication, audit trails of CLI events, and SSL-based data encryption.

Note: Secure CLI is not supported on all operating systems. Refer to the Agent/CLI 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 navisecli command (and switches) together with another subcommand (and its switches). The navisecli command replaces navicli (Classic CLI) and java-jar navicli.jar (Java CLI) in the command line and, in general, preserves the original command syntax and output for script compatibility.

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.

Secure CLI does not support commands targeted to Host Agents, such as -lunmapinfo. For commands that can be issued 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.

Note: Secure CLI does not distinguish case of characters, so regardless of the server operating system, you can use 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. To retain consistency with previous commands, if the command fails, Secure CLI returns 1 for commands that originated in Java CLI and 1 or greater for commands that originated in Classic CLI.
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 account using Navisphere CLI, refer to the *EMC Navisphere Command Line Interface (CLI)*. For details on using Navisphere 6.X security, refer to the *EMC Navisphere Security Administrator’s Guide*.

You can also choose to configure a Navisphere 6.X security file (see the next section, “Overview of using the Navisphere 6.X security with Secure CLI”) on the server. 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 the 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 or local) for each command you issue, or, more conveniently, you can create a Navisphere security file.

**Note:** If you have an existing security file to issue Java CLI commands, you must create a new security file for Secure CLI commands. The security files can co-exist, however you cannot use a security file implemented for Java CLI commands to issue Secure CLI commands. The security file you create using Secure CLI, implements a different architecture and additional enhancements, such as the -user and -seccfilepath options.

The Navisphere security file is an encrypted file stored for each user on each server. 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
server. You must issue the `-AddUserSecurity` function on the server 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 server 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 server, given the alternative username `altusername`, the password `mypass` and the scope 0 (global scope), type:

```
naviseccli -AddUserSecurity -password mypass -scope 0 -user altusername
```

Then, on this server, 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\n```

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.

**User access**
Anyone that can log in to the server running Navisphere CLI 6.X or later.

**Format**
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]
[-timeout | -t timeout]
[-user username]
[-v]
[-xml]
CMD [optional_command_switches]
```

where the `naviseccli` switches are:

- **-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*  
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 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

Specifies whether the user account on the storage system you want to log in to is local or global. A 0 (default) indicates global; a 1 indicates local.
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.

-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").
About Java CLI

Java CLI is implemented in Java and its commands run in a command window. Each Java CLI command consists of the command `java -jar navicli.jar` and switches together with functions and function switches.

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

**Important** - You can issue this command to either SP; however, we recommend that you issue it from the SP that owns the source LUN(s) to prevent any timeouts from occurring.

If a Java CLI command fails and the CLI does not generate its own error message, it displays an error message from the Host Agent or SP Agent. Java CLI generates errors about command line syntax for commands and options and their values. Java CLI commands return 0 if the command is successful and 1 if the command fails.

Requirements for using Java CLI

Before you can use the MirrorView/A CLI commands, you must install a Java runtime environment (JRE). This is required for the `navicli.jar` command. For the latest JRE supported, see the EMC Navisphere Manager and Manager Base Release Notes.

Overview of using the Navisphere 6.X security with Java CLI

A storage system will not accept a command from the Java 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 or local) in each command line, or more conveniently, you can add password and scope information to the Navisphere security file and omit the password and scope from the command line. The Navisphere security file is an encrypted file stored for each user on each server. You can add or remove a password or scope from the security file using the `-AddUserSecurity` or `-RemoveUserSecurity` functions as arguments to the `navicli.jar` command.
You can create a user account on the storage system using the CLI. Refer to *EMC Navisphere Command Line Interface (CLI)* for more information. The username on the storage system must match the username you used to log in to this server. The CLI will always pass your current username to the storage system; you cannot specify a username in an `-AddUserSecurity` request.

For example, to add yourself under your current username to the security file on the current server, given the password `mypass` and the scope 0 (global scope), type:

```
java -jar navicli.jar -AddUserSecurity -password mypass -scope 0
```

Then, on this server, you can enter CLI commands to any storage system on which you have an account that matches your current username with password `mypass` and global scope (`scope 0`).

---

**Note:** The password is passed to the storage system just as you typed it. It is case-sensitive.
navicli.jar

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

**Note:** All switches for the `java -jar navicli.jar` command are case-insensitive.

**Description**
The `java -jar navicli.jar` command sends storage-system management and configuration requests to a storage system via the Internet.

**User access**
You must be able to log in to the server running Navisphere CLI 6.X or later.

**Format**
The `navicli.jar` command is used as follows:

```
java -jar navicli.jar  -help
```

or

```
java -jar navicli.jar  [navicli switches] CMD [optional switches]
```

or for NetWare

```
java -jar [-ns | -nsac] navicli.jar [navicli switches] CMD [optional switches]
```

where the `navicli` switches are:

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

- **-address**  *address* or **-h**  *address*
  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 **-password**  *password* and **-scope** switches.
-f filename

Specifies a file to redirect any output that would otherwise appear after the command itself.

-nopoll

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.

-ns | -nsac

For NetWare systems, creates a new screen for the Java application. One of these switches is needed with NetWare for CLI command output to appear on the screen. -ns creates a new screen for the Java application. -nsac also creates a new screen for the Java application, and automatically closes the screen when the application terminates.

-parse

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 case sensitive. You can omit this if you are using the security file.

-port port

Sets the port number of the storage system. The default is 80.

-q

Prevents error and help messages from displaying. ("Quiet" operation.)

-RemoveUserSecurity

Directs the CLI to remove security information about the current user from the security file on this server.
-scope 0|1

Specifies whether the user account on the storage system you want to log in to is local or global. A 0 (default) indicates global; a 1 indicates local.

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/she has a local account.

-timeout seconds

Sets the time-out value in seconds. The default is 600 seconds.

-user username

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

-xm1

Specifies command output in XML format. Use the -o (override switch) when specifying -xm1 on commands that require confirmation. Otherwise, the XML output will contain your confirmation string.

CMD

The MirrorView/A command (CMD) is mirror.

The CLI will set the exit code before terminating. It sets the exit code to zero when the command executes successfully or to one when the command fails. Failures include command line parsing errors as well as errors encountered during the execution of the 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").
MirrorView/A (remote mirroring) commands

MirrorView/A CLI functions

The Navisphere CLI provides one of the two interfaces to the MirrorView/A mirroring software. The Navisphere Manager UI provides the other. You can use the naviseccli mirror or navicli.jar mirror CLI command to set up and manage remote mirror sessions.

The naviseccli or navicli.jar mirror command functions are summarized in Table 2-1.

<table>
<thead>
<tr>
<th>Table 2-1 naviseccli or navicli.jar MirrorView/A command functions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Essential functions (in order performed)</strong></td>
</tr>
<tr>
<td>mirror -enablepath</td>
</tr>
<tr>
<td>Opens a path between the local and remote storage system.</td>
</tr>
<tr>
<td>Required regardless of the type of connection between</td>
</tr>
<tr>
<td>storage systems.</td>
</tr>
<tr>
<td>mirror -async -create</td>
</tr>
<tr>
<td>Creates a new remote mirror using an existing LUN. The LUN</td>
</tr>
<tr>
<td>on which the -create command is executed becomes the</td>
</tr>
<tr>
<td>primary image.</td>
</tr>
<tr>
<td>mirror -async -addimage</td>
</tr>
<tr>
<td>Adds a secondary image to a mirror using a LUN on a remote</td>
</tr>
<tr>
<td>storage system. The default is to synchronize unless you</td>
</tr>
<tr>
<td>specify otherwise.</td>
</tr>
<tr>
<td>mirror -async -changeimage</td>
</tr>
<tr>
<td>Changes secondary image properties such as auto recovery</td>
</tr>
<tr>
<td>setting and recovery policy.</td>
</tr>
<tr>
<td>mirror -async -fractureimage</td>
</tr>
<tr>
<td>Fractures a secondary image, suspending mirroring of data to</td>
</tr>
<tr>
<td>that image. A fracture log is maintained, so that the image</td>
</tr>
<tr>
<td>can be quickly resynchronized later.</td>
</tr>
<tr>
<td>mirror -async -removeimage</td>
</tr>
<tr>
<td>Removes a secondary image from a mirror.</td>
</tr>
<tr>
<td>mirror -async -promoteimage</td>
</tr>
<tr>
<td>Promotes a secondary image to primary, while simultaneously</td>
</tr>
<tr>
<td>demoting the primary (if still accessible) to secondary.</td>
</tr>
<tr>
<td>Required in order to access the data in the secondary image</td>
</tr>
<tr>
<td>in the event of a failure of the primary.</td>
</tr>
<tr>
<td>mirror -async -syncimage</td>
</tr>
<tr>
<td>Starts synchronization of the secondary image with the</td>
</tr>
<tr>
<td>primary. Needed if automatic recovery is not on. I/O is</td>
</tr>
<tr>
<td>allowed with primary while synchronization occurs. You can</td>
</tr>
<tr>
<td>specify a synchronization rate to avoid serious performance</td>
</tr>
<tr>
<td>impact.</td>
</tr>
<tr>
<td>mirror -async -info</td>
</tr>
<tr>
<td>Displays MirrorView/A information about a storage system.</td>
</tr>
<tr>
<td>mirror -async -list</td>
</tr>
<tr>
<td>Displays information on existing mirrors.</td>
</tr>
</tbody>
</table>
### Table 2-1: navisecccli or navicli.jar MirrorView/A command functions (continued)

<table>
<thead>
<tr>
<th>Optional reconfiguration functions (alphabetically)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>mirror -async -change</td>
<td>Changes remote mirror properties.</td>
</tr>
<tr>
<td>mirror -async -destroy</td>
<td>Terminates remote mirroring of a LUN. This does not affect the data in the LUN, and the server continues to access it uninterrupted.</td>
</tr>
<tr>
<td>mirror -disablepath</td>
<td>Severs all paths between the local storage-system SPs and the remote storage-system SPs.</td>
</tr>
</tbody>
</table>
mirror -async -addimage

Description
The navisecl or navicli.jar mirror -async command with -addimage adds a secondary image to a mirror using a LUN on a remote storage system. The -addimage function is required if you want mirroring to occur. You must direct the command to the storage system that holds the primary image.

You can use the -info function to display the UIDs of storage systems that have images that can be mirrored.

Note: To add a secondary image, you must have a working connection to the remote storage system, and the remote LUN must be exactly the same size (number of blocks) as the primary image.

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

Format
mirror -async -addimage is used with navisecl (described on page 2-10) or navicli.jar (described on page 2-17) as follows:

mirror -async -addimage -name name | -mirroruid mirroruid
-arrayhost sp-hostname | sp-IP-address -lun lun-number | -lunuid lun-uid

or

-arrayuid storage-system-uid -lunuid lun-uid
[-recoverypolicy auto | manual]
[-syncrate high | medium | low]
[-manualupdate | -delaystart time | -enddelay time]
[-nosyncrequired]

Note: In this format "or" means that you can specify the switch, -arrayhost sp-hostname | sp-IP-address -lun lun-number | -lunuid lun-uid or you can specify the switch, -arrayuid storage-system-uid -lunuid lun-uid. The rest of the format is common to both.

where:

-name name

Specifies the name that you gave to the mirror with the -create command.
-mirroruid  uid

Specifies the 16-byte primary image unique ID (World Wide Name) the software assigned when the mirror was created. For any letters in the ID, you must specify the same case as the software assigned when the mirror was created (use `mirror -list` function, page 2-43).

-arrayhost  sp-hostname | sp-IP-address

Specifies the hostname or network address of the remote SP that will hold the secondary image. The secondary image will be added to the remote mirror connected to this host’s SP. Use this switch or `-arrayuid`, next.

-arrayuid  storage-system-uid

Specifies the 8-byte storage system unique ID of the remote SP. You can determine this by using the `mirror -info` command (page 2-41) or the `getarrayuid` command (see EMC Navisphere CLI Reference). For any letters in the ID, use the same case that the software assigned when the mirror was created. Use this switch or `-arrayhost`, above.

-lun  lun-number

Specifies the LUN number that will serve as the secondary image of the mirror.

-lunuid  lun-uid

Specifies the 16-byte unique identification number (World Wide Name) of the LUN to be added as a secondary image. If you specified `-arrayuid`, then `-lunuid` is required. If you specified `-arrayhost`, then you can use either `-lun` or `-lunuid`.

**Note:** When you specify `-arrayuid` and `-lunuid`, you must add the mirror driver on the remote LUN using the `-setfeature` command before executing the `-addimage` command.

-recoverypolicy  manual | auto

Specifies the policy for recovering the secondary mirror image after a system fracture. Values are:

- **manual** The administrator must explicitly start a synchronization operation to recover the secondary mirror image.
auto  Recovery automatically resumes as soon as the primary image determines that the secondary mirror image is once again accessible. This is the default.

-syncrate  high | medium | low

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 server I/O requests, particularly where there are multiple concurrent synchronizations occurring. **Low** completes updates much slower, but also minimizes impact on other storage system operations.

-manualupdate

Indicates that you must explicitly update the image.

-delaystart  time

Specifies the time (in minutes) from the beginning of the previous update to the start of the next update. The current update must complete before the next one can start. If the update is still in process when the time period expires, the next update will start immediately once the current update completes.

-enddelay  time

Specifies the time (in minutes) from the end of the previous update to the next update. A value of 0 causes updates to occur as fast as possible.

-nosyncrequired

Specifies that the image will be added in a synchronized state; thus, no initial synchronization will occur.

**Conventions and recommendations**

None.

**Example**

Any of the following command lines adds an image to a mirror:

```
```

**Note:** If you specify the remote storage system using the **-arrayuid** switch, before issuing the command line above, you must explicitly add drivers to the LUN stack using the **-setfeature** command.
or

```
naviseccli -h ss1_spa mirror -async -addimage -name db_mirror -arrayhost cpc426 -lun 9
```

or

```
naviseccli -h ss1_spa mirror -async -addimage -name db_mirror -arrayhost cpc426 -lunuid 60:06:06:5A:62:02:00:00:92:95:DE:7C:E4:71:D4:11
```

For SP `ss1_spa`, any of the above commands adds to the remote mirror `db_mirror` the LUN in storage system `50:06:06:10:00:FD:A1:69` (or a LUN on the remote storage system `cpc426`) whose LUN unique ID (world wide name, WWN) is `60:06:06:5A:62:02:00:00:92:95:DE:7C:E4:71:D4:11` or whose LUN number is 9.

**Note:** To use this command with `navicli.jar`, replace `naviseccli` with `java -jar navicli.jar`.

See also the example for `mirror -async -create` on page 2-32.

**Output**

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 `stdout`. You may also receive other error messages.
MirrorView/A (remote mirroring) commands

**mirror -async -change**

Changes mirror properties

**Description**
The `naviseccli` or `navicli.jar` `mirror -async` command with `-change` changes one or more parameters of a previously created mirror.

You must direct the command to the storage system that holds the primary image. You must specify at least one of the optional switches with this command. If you omit the override option, then the CLI displays a message for each parameter to confirm the change.

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

**Format**
`mirror -async -change` is used with `naviseccli` (described on page 2-10) or `navicli.jar` (described on page 2-17) as follows:

```
mirror -async -change -name name | -mirroruid uid
[-description description]
[-requiredimages num_of_images] [-o]
```

where:

- **-name name**
  Specifies the name the `-create` command gave the mirror. You can specify either the mirror name or unique ID (next).

- **-mirroruid uid**
  Specifies the 16-byte remote mirror unique ID assigned when the mirror was created. For any letters in the ID, you must specify the same case as the software assigned when the mirror was created (use `mirror -list` command).

- **-description description**
  Provides a meaningful title for the mirror (for example, Employee and Benefits Table for Human Resources). The software does not use the description to identify the mirror. You can specify up to 256 characters, but we suggest you not exceed the line length defined by the operating system.

- **-requiredimages num_of_images**
  Specifies the minimum number of secondary images that should be accessible to the mirror. The default value is 0. Allowed values are 0, 1, and all. If the number of accessible images goes below the
specified value, the mirror goes into the Attention state and generates a log message. You can configure this to generate an appropriate warning (for example, e-mail message or page) to the administrator to enable the administrator to correct the problem.

-o

Executes the command without prompting for confirmation.

Conventions and recommendations
None.

Example

```
navisecll.jar -h ss0_spa mirror -async -change -name dbmirror
-requiredimages 0
```

For the remote mirror `dbmirror` on SP `ss0_spa`, this command reduces the minimum number of accessible secondary images to 0.

Note: To use this command with `navicli.jar`, replace `navisecll` with `java -jar navicli.jar`.

Output
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 stdout. You may also receive other error messages. Use the -list command to confirm that the changes were made.
mirror -async -changeimage

Changes secondary image properties

**Description**
The `naviseccli` or `navicli.jar` mirror -async command with -changeimage changes one or more parameters of a mirror image.

You must direct the command to the storage system that holds the primary image. You must specify at least one of the optional switches. If you omit the override option, then it displays a message similar to the following for each parameter to confirm the change.

Do you want to change the sync rate from Medium to High (y/n)?

To change a parameter, enter y.

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

**Format**
`mirror -async -changeimage` is used with `naviseccli` (described on page 2-10) or `navicli.jar` (described on page 2-17) as follows:

```
mirror -async -changeimage -name name | -mirroruid uid
-imagemuid image-uid [-recoverypolicy manual | auto]
([-manualupdate] | [-delaystart time] | [-enddelay time]) [-syncrate rate] [-o]
```

where:

- **-name name**
  Specifies the name you gave to the remote mirror in the -create command. You can specify either the mirror name or uid (next).

- **mirroruid uid**
  Specifies the 16-byte unique ID assigned when the mirror was created. For any letters in the ID, you must specify the same case as the software assigned when the mirror was created (use `mirror -list` command).

- **imagemuid image-uid**
  Specifies the 8-byte unique ID of the secondary image you want to change. For any letters in the ID, you must specify the same case that the software assigned to the image when the mirror was created (use `mirror -list` command).
-recoverypolicy manual | auto

Specifies the policy for recovering the secondary mirror image after a system fracture. Values are

- **manual** The administrator must explicitly start a synchronization operation to recover the secondary mirror image.

- **auto** Recovery automatically resumes as soon as the primary image determines that the secondary mirror image is once again accessible. This is the default.

-syncrate high | medium | low

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 server I/O requests, particularly where there are multiple concurrent synchronizations occurring. **Low** completes updates much slower, but also minimizes impact on other storage system operations.

-manualupdate

Indicates that you must explicitly update the image.

-delaystart time

Specifies the time (in minutes) from the beginning of the previous update to the start of the next update. The current update must complete before the next one can start. If the update is still in process when the time period expires, the next update will start immediately once the current update completes.

-enddelay time

Specifies the time (in minutes) from the end of the previous update to the next update. A value of 0 causes updates to occur as fast as possible.

-o

Executes the command without prompting for confirmation.

**Conventions and recommendations**

None.
Example

```
naviseclli -h server1 mirror -async -changeimage -name db_mirror -imageuid 50:06:06:10:00:FD:A1:6 -syncrate high
```

For storage system `server1`, mirror `db_mirror`, image `50:06:06:10:00:FD:A1:6`, this command changes the synchronization rate to high.

**Note:** To use this command with `navicli.jar`, replace `naviseclli` with `java -jar navicli.jar`.

Output

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 `stdout`. You may also receive other error messages. Use the `-list` command to confirm that the changes were made.
mirror -async -create

Creates a new asynchronous mirror using an available LUN for mirroring.

Description

The naviseccli or navicli.jar mirror -async command with -create creates a new mirror using an existing LUN. The LUN on which you run the command becomes the primary LUN. But, for mirroring to occur, you must add an image with the -addimage function. Later, you can change some of the parameters you specify here with the -change function.

If you specify a minimum number of required images (other than the default value zero) in the -create command, the mirror will start in the attention state, since you have not yet added the required secondary image(s). To avoid this problem, do not specify a minimum number of images with -create; instead, create the mirror, add the image, and then if you want a minimum number of required images, change the mirror with the -change command to require it.

User access

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

Format

mirror -async -create is used with naviseccli (described on page 2-10) or navicli.jar (described on page 2-17) as follows:

```
mirror -async -create -name name -lun lun_number
[-description description] [-requiredimages num_of_images] [-o]
```

where:

- **-name name**

  Lets you assign a meaningful name to the mirror; for example "Employee Benefits." To include spaces in the name, enclose the entire string in quotation marks as in the example. You can use this name to manage the mirror. The name can include as many as 64 characters.

- **-lun lun_number**

  Specifies the LUN number that will serve as the primary image of the mirror you are creating.
-description description

Lets you assign a description for the mirror. The software does not use the description to identify the mirror. You can specify up to 256 characters, but we suggest you not exceed the line length defined by the operating system.

-requiredimages num_of_images

Specifies the minimum number of secondary images that should be accessible to this mirror. The default value is 0. Allowed values are 0, 1, and all. If the number of accessible images goes below the specified value, the mirror goes into the Attention state and generates a log message. You can configure this to generate an appropriate warning (for example, e-mail message or page) to the administrator to enable the administrator to correct the problem.

-o

Executes the command without prompting for confirmation.

Conventions and recommendations

You can use the -info command to display the lun numbers that can be mirrored.

Example

```
naviseccli -h ss1_spa mirror -async -create -name db_mirror -lun 112
```

The mirror -async -create command creates the mirror named db_mirror on LUN 112. The mirror automatically activates.

Note: To use this command with navicli.jar, replace naviseccli with java -jar navicli.jar.

Output

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 stdout. You may also receive other error messages.
MirrorView/A (remote mirroring) commands

**mirror -async-destroy**

Destroys a mirror

**Description**

The `naviseccli` or `navicli.jar mirror -async` command with `-destroy` destroys a mirror. It is the counterpart of `-create`. It ends mirroring of the LUN that is part of the mirror. This does not affect the data in the LUN, and the server continues to access it uninterrupted.

---

**Note:** You cannot destroy a mirror while it is a member of a consistency group.

You must remove the secondary image in the mirror before issuing the command to the storage system holding the primary image. If you cannot remove the secondary image or access the primary image, you can use the `-force` option. You must direct the command to the storage system that holds the primary image unless the `-force` option is specified, in which case, you can direct it to either storage system.

---

**CAUTION**

Use the `-force` option only if all other options to remove an image and the mirror have failed.

---

If you omit the override option, a message to confirm the destruction of the mirror appears.

Do you really want to destroy the mirror <mirror-name> (y/n)?

To destroy the mirror, answer y; to take no action, enter n.

**User access**

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

**Format**

`mirror -async -destroy` is used with `naviseccli` (described on page 2-10) or `navicli.jar` (described on page 2-17) as follows:

```bash
mirror -async -destroy -name name | -mirroruid uid [-force] [-o]
```

where:

- `-name name`

  Specifies the name you gave to the mirror with the `-create` command. You can also use the remote mirror ID.
**MirrorView/A (remote mirroring) commands**

- **mirroruid** *uid*
  
  Specifies the 16-byte remote mirror unique ID assigned when the mirror was created. For any letters in the ID, you must specify the same case as the software assigned when the mirror was created (use `mirror -list` command).

- **force**
  
  Forcibly destroys the mirror. This option affects only the local storage system.

- **-o**
  
  Executes the command without prompting for confirmation.

**Conventions and recommendations**

None.

**Example**

```
navisecli -h ss1_spa mirror -async -destroy db_mirror
```

Do you really want to destroy the mirror `db_mirror`? (y/n)? y

For SP `ss1_spa`, this command destroys the remote mirror that was created with the name `db_mirror`.

**Note:** To use this command with `navicli.jar`, replace `navisecli` with `java -jar navicli.jar`.

**Output**

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 `stdout`. You may also receive other error messages if the command fails for some reason.
**mirror -disablepath**

Disables the communication paths created by -enablepath to remote mirror SPs

**Description**

The navisecli or navicli.jar mirror command with -disablepath severs the path established by -enablepath in the remote storage system.

When issued to any SP, the command severs paths for both SPs in both the primary and secondary storage systems; that is, it severs these paths if they exist:

- SP A of the local storage system to SP A of the remote storage system
- SP A of the remote storage system to SP A of the local storage system
- SP B of the local storage system to SP B of the remote storage system
- SP B of the remote storage system to SP B of the local storage system

This means you can issue the command to either SP of either storage system to sever the paths between the storage systems you are targeting.

If you issue the command while a mirror exists, the CLI issues a warning message; if you continue, any mirror connected over the specified path will be fractured.

**Note:** If both asynchronous and synchronous mirrors are installed, you need to run the command only once. It will take effect for both mirrors.

If you omit the override option, the CLI displays a message to confirm the disabling of the path. To disable the path, answer y; to take no action, enter n.

**User access**

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

`mirror -disablepath` is used with `naviseccli` (described on page 2-10) or `navicli.jar` (described on page 2-17) as follows:

```
mirror -disablepath connected-SP-hostname | connected-SP-IP-address [-o]
```

where:

- `connected-SP-hostname | connected-SP-IP-address`
  
  Specifies the hostname or IP address of SP A or SP B of a connected storage system to disconnect.

- `-o`
  
  Executes the command without prompting for confirmation. Without this switch, the CLI displays a confirmation message for each SP.

### Conventions and recommendations

None.

### Example

```
naviseccli -h ss1_spa mirror -disablepath ss27_spa
```

Disabling of the path will stop mirroring of data to all images if any among these arrays. A system fracture of all mirrors between two arrays on following storage processors will occur due to this action.

```
128.221.39.29
10.14.20.154
```

Do you want to continue (y/n)?  y

If primary and secondary mirrors exist, a warning message appears.

For SP `ss1_spa`, this command and confirmation remove the paths between the local SPs and the remote mirror SPs.

```
naviseccli -h ss1_spa mirror -async -info
```

**Note:** To use this command with `navicli.jar`, replace `naviseccli` with `java -jar navicli.jar`.

The `mirror -info` command displays the updated status.

### Output

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 `stdout`. 
MirrorView/A (remote mirroring) commands

**mirror -enablepath**

Establishes a path for remote mirroring between a local and remote storage system

**Description**

The `naviseccli` or `navicli.jar` mirror command with **-enablepath** creates a path between the specified SP and its peer SP to a remote SP and its peer. These paths will serve for remote mirroring. A remote mirror requires a path between the primary and secondary storage systems.

When issued to any SP, the command creates paths for both SPs in both storage systems; that is, it creates these paths:

- SP A of the local storage system to SP A of the other storage system
- SP A of the other storage system to SP A of the local storage system
- SP B of the local storage system to SP B of the other storage system
- SP B of the other storage system to SP B of the local storage system

The SP As or the SP Bs in both storage systems must be physically connected.

If you cannot enable paths on one SP due to some error, an informational message appears and the paths are enabled for the other SP. To obtain status information, run the `mirror -info -systems` command.

**Note:** If both asynchronous and synchronous mirrors are installed, you will need to run the command only once. It will take effect for both mirrors.

**User access**

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

**Format**

`mirror -enablepath` is used with `naviseccli` (described on page 2-10) or `navicli.jar` (described on page 2-17) as follows:

`mirror -enablepath  SP-hostname \ SP-IP-address`
where:

*SP-hostname | SP-IP-address*

Specifies the hostname or IP address of SP A or SP B of the remote storage system on which you want to enable remote mirroring.

**Conventions and recommendations**

None.

**Example**

`naviseccli -h ss1_spa mirror -enablepath ss27_spa`

For SP *ss1_spa*, this command enables the paths between *ss1_spa* and its peer SP and *ss27_spa* and its peer SP.

`naviseccli -h ss1_spa mirror -async -info`

The `mirror -info` command displays the updated status.

---

**Note:** To use this command with `navicli.jar`, replace `naviseccli` with `java -jar navicli.jar`.

**Output**

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 `stdout`. 
mirror -async -fractureimage

Fractures an image in a mirror

**Description**

The `naviseccli` or `navicli.jar mirror -async` command with `-fractureimage` starts an administrative fracture of a secondary image. It also prevents scheduled updates from starting in the future. You must direct the command to the storage system that holds the primary image.

(An automatic fracture called a system fracture occurs if a secondary image becomes inaccessible to the primary image. After a system fracture, the system periodically tries to communicate with the secondary image and — if it succeeds — starts synchronization based on the fracture log. If you issue the `-fractureimage` while the mirror is system fractured, the system will not try to communicate or resynchronize until you issue a `-syncimage` command.)

The software displays a message to confirm the fracture:

Do you want to fracture the image *image UID* (y/n)?

To fracture the mirror, answer **y**; to take no action, enter **n**.

**User access**

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

**Format**

`mirror -async -fractureimage` is used with `naviseccli` (described on page 2-10) or `navicli.jar` (described on page 2-17) as follows:

`mirror -async -fractureimage -name name | -mirroruid uid -imageuid uid [-o]`

where:

- **-name name**
  
  Specifies the name you gave to the mirror with the `-create` command.

- **-mirroruid uid**
  
  Specifies the unique 16-byte mirror ID (World Wide Name) that the software assigned when it created the mirror.

- **-imageuid uid**
  
  Specifies the 8-byte image unique ID of the secondary image to be fractured.
**-o**

Executes the command without prompting for confirmation.

### Conventions and recommendations

None.

### Example

```bash
navisecli -h ss1_spa mirror -async -fractureimage -name db_mirror -imageuid 50:06:06:10:00:FD:A1:6
```

Do you really want to fracture the image (y/n)? y

For SP ss1_spa of the mirror named db_mirror, this command fractures the secondary image with the unique ID shown.

---

**Note:** To use this command with navicli.jar, replace navisecli with java -jar navicli.jar.

---

### Output

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 `stdout`. You may also receive other error messages.
**mirror -async -info**

Displays general information on asynchronous mirroring in a storage system.

**Description**
The `naviseccli` or `navicli.jar mirror -async` command with the `-info` function and no other switches lists all general information for asynchronous mirror support. Issuing this command with one or more options displays the information based on the option issued. See the example for sample output.

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

**Format**
`mirror -async -info` is used with `naviseccli` (described on page 2-10) or `navicli.jar` (described on page 2-17) as follows:

```
mirror -async -info [-all] [-cancreate] [-maxmirrors]
[-maxremoteimages] [-mirroredluns] [-mirrorableluns] [-systems]
```

where:

- **-all**

  Returns all general information for asynchronous mirror support.

- **-cancreate**

  Indicates whether an asynchronous mirror can be created in this storage system: yes or no.

- **-maxmirrors**

  Displays the maximum number of mirrors that can be created in this storage system.

- **-maxremoteimages**

  Specifies the maximum number of secondary images allowed. A maximum of one image is allowed.

- **-mirroredluns**

  Lists the LUN numbers of LUNs that are mirrored in this storage system. The list includes LUNs used as both secondary and primary images.
MirrorView/A (remote mirroring) commands

-mirrableluns

Lists LUNs that can be mirrored in this storage system but are not mirrored at present.

-systems

Lists remote storage systems that can be used for a secondary image.

Conventions and recommendations

None.

Example

naviseclli -h ss1_spa mirror -async -info

Can a mirror be created on this system: YES
Maximum number of remote images: 1
Logical Units that are mirrored in Primary Images: 128 5
Logical Units that are mirrored in Secondary Images: 0
Logical Units that can be mirrored: 1 2 3 4 10 11 12 13 14 21 22 58 59
Remote systems that can be enabled for mirroring:
Remote systems that are enabled for mirroring:
Array UID: 50:06:01:60:90:20:27:47
Status: Enabled on both SPs
Maximum number of possible Mirrors: 100

For SP ss1_spa, this command displays potential and current remote mirror information.

Note: To use this command with navicli.jar, replace naviseclli with java -jar navicli.jar.

Output

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 stdout.
mirror -async -list

Displays information about existing asynchronous mirrors

Description

The naviseccli or navicli.jar -async mirror command with -list and no other switches lists all information for all mirrors on the storage system. Issuing this command with one or more options displays the information based on the option(s) issued. You can direct the command to any storage system; however, only the storage system holding the primary image of a mirror will always have the latest information about the mirror. Status obtained from other storage systems may be stale; do not use it unless the primary storage system is unavailable. See the example for sample output.

User access

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

Format

mirror -async -list is used with naviseccli (described on page 2-10) or navicli.jar (described on page 2-17) as follows:

mirror -async -list [-name [name]] [-mirroruid] [uid]
[ -ownergroupid] [-ownergroupname] [-requiredimages] [-state]
[ -timestamp] [-transition]

where:

-name name

Specifies the name you gave the mirror with the -create function. You can use the mirror name or unique ID to obtain information on a mirror.

-mirroruid uid

Specifies the 16-byte remote mirror unique ID (World Wide Name) the software assigned when it created the mirror.

-description

Lists the meaningful title for the mirror, if any. The software does not use the description to identify the mirror.

-faulted

Lists whether the remote mirror is faulted: yes or no. If it is faulted, the mirror is either fractured or in the attention state.
**MirrorView/A (remote mirroring) commands**

- **-images**
  Displays image information, such as the role of the image (primary or secondary), image ID, WWN of the LUN comprising the image, recovery policy, and sync rate, image state, image condition, and image transitioning. Also displays update type (Manual, Time elapsed since start of last update, and Time elapsed since end of last update), update period in minutes (displays N/A where not applicable), time in minutes since last update, time in minutes until next update, last image error (why the image is faulted; displays Not Available if no error).

- **-imagecount**
  Lists the current number of images that compose the mirror. This count includes the primary image.

- **-imagesize**
  Lists the size of the image (capacity of the LUN containing the image) in user blocks.

- **-lun**
  Lists the LUN number of the local image.

- **-ownergroupid**
  Lists the unique ID value of the consistency group of which the mirror is a member.

- **-ownergroupname**
  Lists the unique name of the consistency group of which the mirror is a member.

- **-requiredimages**
  Lists the minimum number of secondary images required for the mirror.

- **-state**
  Describes the operational state of the mirror. Values are active and attention.

- **-timestamp**
  Returns the clock time when the last update started.
-transition

Specifies whether the mirror is changing state.

Conventions and recommendations

None.

Example

navisecli -h ss1_spa mirror -async -list

MirrorView Name:  pbc async ml 80
MirrorView Description:
MirrorView UID:  54:33:20:90:60:01:06:50:03:00:00:00:00:00:00:00
Logical Unit Numbers:  80
Remote Mirror Status:  Mirrored
MirrorView State:  Active
MirrorView Faulted:  NO
MirrorView Transitioning:  NO
Minimum number of images required:  0
Image Size:  409600
Image Count:  2
Images:
Image UID:  50:06:01:60:90:20:33:54
Is Image Primary:  YES
Logical Unit UID:  60:06:01:60:2F:E7:0C:00:3C:5B:04:A2:50:4C:D8:11
Image Condition:  Unknown Image Condition

Is Image Primary:  NO
Logical Unit UID:  60:06:01:60:3F:E1:0D:00:A4:C9:01:63:B6:4C:D8:11
Image State:  Synchronized
Image Condition:  Normal
Recovery Policy:  Automatic
Synchronization Rate:  Medium
Image Faulted:  NO
Image Transitioning:  NO
Synchronizing Progress(%):  0
Update Type:  Manual
Time in secs since previous update:  12550
Time in secs until next update:  Not Available

Note: To use this command with navicli.jar, replace navisecli with java -jar navicli.jar.

Output

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 stdout. You may also receive other error messages.
mirror -async -promoteimage

Promotes a secondary image to primary

**Description**
The `navisecli` or `navicli.jar -async mirror` command with `-promoteimage` promotes a secondary image to the role of the primary image. The command is successful if it can promote the secondary to a primary that is server-accessible. You must direct this command to a storage system that holds the secondary image — never to the storage system that holds the primary. The command will fail if you issue it to the storage system that holds the primary image.

You can promote a secondary image if the secondary is in a synchronized or consistent state. There must be no current update, in addition to the state restrictions listed.

When you promote a secondary image to primary, the software assigns a new mirror UID to the promoted image to distinguish it from the old mirror. The new status of the old primary image depends on whether the old primary image is accessible when promotion occurs.

After promoting an image, you must explicitly add the newly promoted image to an appropriate storage group (and incidentally, you should also remove the old primary from its storage group, if possible).

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

**Format**
`mirror -async -promoteimage` is used with `navisecli` (described on page 2-10) or `navicli.jar` (described on page 2-17) as follows:

```
mirror -async -promoteimage -name name | -mirroruid uid
-imageuid imageuid [-type normal | oos | local] [-o]
```

*where:*

```
-name name
```

Specifies the name you gave the mirror with the `-create` command. You can use the mirror name or unique ID to promote a secondary image.
-mirroruid  uid

Specifies the unique 16-byte mirror ID (World Wide Name) that was assigned when the mirror was created. For any letters in the ID, see comments under -imageuid below.

-imageuid  uid

Specifies the unique 8-byte image ID of the secondary image you want to promote. For any letters in the ID, you must specify the same case as the software assigned to the image when it created the mirror (use mirror -list function).

-type  normal | oos | local

Checks the status of the mirror images. Normal (the default) directs the driver to perform internal safety checks and return the appropriate error if the promotion would cause an out-of-sync state for the old primary or remote image if the mirror is unreachable. oos indicates that no safety check is performed, and the promote will occur even if it results in an out-of-sync state for the old primary. local results in promoting the local image only to the primary. The driver attempts to contact the old primary, and if it reaches it, the old primary removes its secondary image.

-o

Executes the command without prompting for confirmation.

Conventions and recommendations

In normal operation when there is no failure, you can (but probably would not want to) promote a secondary image. If you do so, the primary image is demoted to secondary. The new primary image inherits the mirror properties of the demoted primary image. Before issuing this command consider the following:

- If possible, the secondary image should be synchronized (synchronized state) with the primary image. If the secondary image is in the consistent state (not synchronized state) when you promote it to primary image, data will probably be lost during the promotion process.

The secondary image to be promoted must be in a consistent or synchronized state. If the secondary image is out-of-sync or synchronizing, then you cannot promote it; an error will result.

- You must stop I/O to an accessible primary image before you issue this command. If I/O continues to the original primary, data may be lost as software demotes the primary to secondary and/or an SP may reboot unexpectedly during the attempted promotion.
MirrorView/A (remote mirroring) commands

Example

`naviseccli -h ss1_spa mirror -async -promoteimage -name db_mirror -imageuid 50:06:06:10:00:FD:A1:6`

For SP `ss1_spa`, owner of the remote mirror named `db_mirror`, this command promotes the secondary image with the unique ID shown.

Note: To use this command with `navicli.jar`, replace `naviseccli` with `java -jar navicli.jar`.

Output

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 `stdout`. You may also receive other error messages.
**mirror -async -removeimage**

Removes a secondary image from a mirror

**Description**

The `naviseccli` or `navicli.jar mirror -async -removeimage` command removes a secondary image from the remote mirror. You must issue this command to a storage system that holds the primary image.

**Note:** If an update is in progress, this command will not execute. We recommend that you fracture the secondary image first, and then you can issue the `mirror -async -removeimage` command.

If you omit the `-override` option, the software displays a message to confirm the remove operation:

Removing of the image will stop mirroring of data to the specified image. It may cause the minimum number of images required parameter to be violated that will cause the mirror to go into attention state. Do you really want to remove the image(y/n)?

Answering yes will remove the image.

**User access**

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

**Format**

`mirror -async -removeimage` is used with `naviseccli` (described on page 2-10) or `navicli.jar` (described on page 2-17) as follows:

```
mirror -async -removeimage -name name | -mirroruid mirroruid
-arrayhost sp-hostname | sp-IP-address | -imageuid imageuid [-o]
```

where:

- **-name name**
  
  Specifies the name you gave the mirror with the `-create` command. You can use the mirror name or unique ID to remove a secondary image.

- **-mirroruid uid**
  
  Specifies the 16-byte remote mirror unique ID (World Wide Name) assigned when the software created the mirror. For any letters in the ID, see comments under `-imageuid` below.
-arrayhost  *sp-hostname | sp-IP-address*

Specifies the hostname or network address of the remote SP that holds the secondary image you want to remove. If you specify this switch, the driver(s) from the secondary LUN stack is removed.

-imageuid  *image-uid*

Specifies the 8-byte image unique ID of the secondary image you want to remove. For any letters in the ID, you must specify the same case as the software assigned to the image when it created the mirror (use mirror -list command). If you specify this switch, you must explicitly remove the driver(s) from the secondary LUN stack using the -setfeature command.

-o

Executes the command without prompting for confirmation.

**Conventions and recommendations**

None.

**Example**

```bash
navisecli -h ss1_spa mirror -async -removeimage -name db_mirror -imageuid 50:06:06:10:00:FD:A1:6
```

Removing the image will stop mirroring of data to the specified image. It may cause the minimum number of images required parameter to be violated that will cause the mirror to go into attention state. Do you really want to remove the image (y/n)?  

```
y
```

For IP address ss1_spa, owner of the mirror named db_mirror, this command removes the image with the unique ID shown.

**Note:** To use this command with navicli.jar, replace navisecli with java -jar navicli.jar.

**Output**

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 stdout. You may also receive other error messages.
mirror -async -setfeature

Adds or removes an asynchronous mirror driver to or from the LUN stack

**Description**

The `naviseccli` or `navicli.jar -async mirror -setfeature` command with the `-setfeature` adds the asynchronous mirror driver to or removes it from the LUN stack depending on whether you specify the `-on` or `-off` option. You must issue this command to the storage system where the LUN resides.

**Note:** `-setfeature` is needed only if you want to use the `-arrayuid` switch to specify the remote storage system for the `-addimage` command.

**User access**

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

**Format**

`mirror -async -setfeature` is used with `naviseccli` (described on page 2-10) or `navicli.jar` (described on page 2-17) as follows:

`mirror -async -setfeature -on | -off -lun number | lunuid uid`

where:

- `-on`

  Adds the driver to the LUN stack.

- `-off`

  Removes the driver from the LUN stack.

- `-lun number`

  Specifies the LUN number to which you want to add or from which you need to remove the driver.

- `-lun uid`

  Specifies the unique 16-byte identification number (World Wide Name) of the LUN to which you will add the driver or from which you will remove the driver.

**Conventions and recommendations**

None.
MirrorView/A (remote mirroring) commands

Example  naviseccli -h ss1_spa mirror -async -setfeature -on -lun 112
For SP ss1_spa, this command starts adding the driver to LUN 112.

Note: To use this command with navicli.jar, replace naviseccli with java -jar navicli.jar.

Output  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 stdout. You may also receive other error messages.
**mirror -async -syncimage**

Starts an update of the secondary image

**Description**

The navisecli or navicli.jar mirror -async command with -syncimage begins the synchronization operation on a secondary image that was previously administratively fractured or on one that the mirror image condition is normal. You must issue this command to the storage system that holds the primary image. The secondary image should be fractured when performing this operation. If you specify a secondary image that is not fractured, an internal error message will be returned.

The software displays a message to confirm:

*During synchronization, array performance will be impacted. Do you want to sync the image (y/n)?*

Answering **y** will start synchronization.

**User access**

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

**Format**

`mirror -async -syncimage` is used with navisecli (described on page 2-10) or navicli.jar (described on page 2-17) as follows:

`mirror -async -syncimage -name name | -mirroruid mirroruid -imageuid imageuid [ -o ]`

where:

- **-name name**
  
  Specifies the name given to the mirror with the **-create** command. You can use the mirror name or unique ID to update a secondary image.

- **-mirroruid uid**
  
  Specifies the 16-byte remote mirror unique ID (World Wide Name) assigned when the software created the mirror. For any letters in the ID, see comments under **-imageuid** below.

- **-imageuid imageuid**
  
  Specifies the 8-byte image unique ID of the secondary image you will synchronize. For any letters in the ID, you must specify the same case as the software assigned to the image when the software created the mirror (use **mirror -list** command).
**MirrorView/A (remote mirroring) commands**

- **-o**
  
  Executes the command without prompting for confirmation.

**Conventions and recommendations**

None.

**Example**

```
navisecli -h ss1_spa mirror -async -syncimage
-name db_mirror -imageuid 50:06:10:00:FD:A1:6
```

During synchronization, array performance will be impacted. Do you want to sync the image (y/n)?

```
y
```

For SP **ss1_spa**, owner of the mirror named **db_mirror**, this command starts synchronizing the image with the unique ID shown.

**Note:** To use this command with **navicli.jar**, replace **navisecli** with **java -jar navicli.jar**.

**Output**

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 **stdout**. You may also receive other error messages.
This chapter explains the commands for creating and managing groups of mirrors on CX3-series and CX-series storage systems with the EMC MirrorView/A option. This manual refers to groups of mirrors as **consistency groups**.

**Note:** The LUNs in a MirrorView/A consistency group must reside on the same storage system, unlike the volumes in a Symmetrix consistency group, which can reside on multiple storage systems.

**Topics are:**
- `mirror -async -creategroup` ...............................................................3-2
- `mirror -async-destroygroup` .............................................................3-5
- `mirror -async -addtogroup` .............................................................3-7
- `mirror -async -removefromgroup` ....................................................3-9
- `mirror -async -changegroup` ............................................................3-11
- `mirror -async -syncgroup` .............................................................3-14
- `mirror -async -fracturegroup` ..........................................................3-16
- `mirror -async -promotegroup` .......................................................3-18
- `mirror -async -listgroups` ..............................................................3-20
MirrorView/A (consistency group) commands

mirror -async -creategroup

Establishes a consistency group to which you can later add mirrors

**Description**
The naviseccli or navicli.jar mirror command with **-creategroup** creates a consistency group to which you can add mirrors with subsequent commands. You must direct the command to the storage system on which you are creating the MirrorView/A group.

**Note:** On a CX400, CX500, and CX3-20 storage system, you can have 8 consistency groups, and each consistency group can have up to 8 mirrors. On a CX600, CX700, CX3-40, and CX3-80 storage system, you can have 16 consistency groups, and each consistency group can have up to 16 mirrors.

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

**Format**
mirror -async -creategroup is used with naviseccli (described on page 2-10) or navicli.jar (described on page 2-17) as follows:

```
mirror -async -creategroup -name name [-description description] [-manualupdate | -delaystart minutes | -enddelay minutes] [-syncrate rate] [-recoverypolicy policy] [-o]
```

where:

- **-name name**
  Specifies the name of the consistency group. Use only alphanumeric characters for the consistency group name. The name can be up to 32 characters.

- **-description description**
  Specifies a textual description of the consistency group.

- **-manualupdate**
  Specifies that you must explicitly start each update of the group. If you specify neither **-manualupdate**, **-delaystart**, nor **-enddelay**, the default is **-enddelay 60**.

**Note:** If an update is not running, you can manually start an update even if an automatic update interval is set.
-delaystart minutes

Specifies the time (in minutes) from the start of the previous update until the start of the next one. If the previous update has not completed, the next update is delayed until the update in progress has completed. If you specify neither -manualupdate, -delaystart, nor -enddelay, the default is -enddelay 60.

-enddelay minutes

Specifies the time (in minutes) from the end of the previous update until the start of the next one. Specifying a value of 0 results in the group being updated as fast as possible. If you specify neither -manualupdate, -delaystart, nor -enddelay, the default is -enddelay 60.

-syncrate rate

Specifies a relative value (low, medium, or high) for the priority of completing updates. Values for rate are low, medium, or high. High completes updates faster, but may significantly affect storage system performance for server I/O requests. Low completes updates slower, but also minimizes the impact on other storage system operations.

-recoverypolicy policy

 Specifies the policy for recovering the mirrors in a group after a failure causes a system fracture. Values are:

 manual Update of the secondary image does not begin until you explicitly issue a Synchronize command.

 auto Update of the secondary image automatically begins as soon as the connection between the two storage systems is again operational.

-o

Executes the command without prompting for confirmation.

Note: All the switches, except -name and -o are meaningful only if the storage system receiving the command hosts the primary images for the mirrors in the consistency group.

Conventions and recommendations

None.
### MirrorView/A (consistency group) commands

#### Example

```bash
navisecli -h ss1_spa mirror -async -creategroup -name dbgroup
```

For SP `ss1_spa`, this command creates a consistency group with the name `dbgroup`.

**Note:** To use this command with `navicli.jar`, replace `navisecli` with `java -jar navicli.jar`.

#### Output

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 `stdout`. You may also receive other error messages.
mirror -async-destroygroup

**Description**

The naviseccli or navicli.jar mirror -async command with -destroygroup destroys a consistency group. It is the counterpart of -creategroup.

*Note:* Make sure that the consistency group is empty before destroying it.

If you omit the override option, it displays a message to confirm the destruction of the consistency group:

Do you really want to destroy the group <group-name>? (y/n)?

To destroy the consistency group, answer y; to take no action, enter n.

**User access**

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

**Format**

mirror -async -destroygroup is used with naviseccli (described on page 2-10) or navicli.jar (described on page 2-17) as follows:

mirror -async -destroygroup -name name | -groupid id [-force] [-o]

where:

- **-name name**
  
  Specifies the name you gave the mirror with the -creategroup command. You can also use the consistency group ID.

- **-groupid id**
  
  Specifies the consistency group unique ID assigned when the consistency group was created.

- **-force**
  
  Destroys the consistency group even if it is not empty.

  *Note:* Particularly in a failure situation you may use the -force switch to destroy a group that still has members.

- **-o**
  
  Executes the command without prompting for confirmation.
<table>
<thead>
<tr>
<th><strong>Conventions and recommendations</strong></th>
<th>None.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example</strong> naviseccli -h ss1_spa mirror -async -destroygroup dbgroup</td>
<td>Do you really want to destroy the group dbgroup? (y/n)? y</td>
</tr>
<tr>
<td></td>
<td>For SP ss1_spa, this command destroys the consistency group that was created with the name dbgroup.</td>
</tr>
<tr>
<td><strong>Note:</strong> To use this command with navicli.jar, replace naviseccli with java -jar navicli.jar.</td>
<td></td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td>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 stdout. You may also receive other error messages if the command fails for some reason.</td>
</tr>
</tbody>
</table>
**mirror -async -addtogroup**

Adds a mirror to a previously created consistency group

**Description**

The `naviseccli` or `navicli.jar mirror -async` command with `-addtogroup` adds one mirror at a time to a consistency group.

**Note:** The mirror you specify must have its primary image on the storage system.

No two secondary images of mirrors in the consistency group can be on different storage systems.

**User access**

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

**Format**

`mirror -async -addtogroup` is used with `naviseccli` (described on page 2-10) or `navicli.jar` (described on page 2-17) as follows:

```
mirror -async -addtogroup -name name | -groupid id
-mirrorname name | -mirroruid id
```

where:

- **-name name**
  
  Specifies the name of the consistency group to which you want to add the mirror.

- **-groupid id**
  
  Specifies the unique ID of the consistency group to which you want to add the mirror.

- **-mirrorname name**
  
  Specifies the name of the mirror you want to add to the consistency group.

- **-mirroruid id**
  
  Specifies the World Wide Name of the mirror you want to add to the consistency group.

**Conventions and recommendations**

None.
Example

`naviseccli -h ss1_spa mirror -async -addtogroup -name dbgroup -mirrorname "Sales Accounts Mirror to Europe"`

For SP `ss1_spa`, this command adds the mirror, *Sales Accounts Mirror to Europe* to the consistency group, `dbgroup`.

Note: To use this command with `navicli.jar`, replace `naviseccli` with `java -jar navicli.jar`.

Output

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 `stdout`. You may also receive other error messages.
mirror -async -removefromgroup

Removes a mirror from a consistency group

Description
The navisecli or navicli.jar mirror -async command with -removefromgroup removes one mirror at a time from the consistency group.

Note: The mirror you specify must have its primary image on the storage system. This does not apply if you use the -force switch.

If you omit the -o option, the software displays a message to confirm the remove operation. Answering yes will remove the mirror.

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

Format
mirror -async -removefromgroup is used with navisecli (described on page 2-10) or navicli.jar (described on page 2-17) as follows:

mirror -async -removefromgroup -name name | -groupid id
-mirrorname name | -mirroruid id [-force] [-o]

where:
-name name
   Specifies the name of the consistency group from which you want to remove the mirror.

-groupid id
   Specifies the unique ID of the consistency group to which you want to add the mirror.

-mirrorname name
   Specifies the name of the mirror you want to remove from the consistency group.

-mirroruid id
   Specifies the World Wide Name of the mirror you want to remove from the consistency group.

-force
   Removes the mirror even if the connection with the remote storage system is not functioning.
### MirrorView/A (consistency group) commands

- `-o`  
  Executes the command without prompting for confirmation.

#### Conventions and recommendations

None.

#### Example

```
naviseccli -h ss1_spa mirror -async -removefromgroup -name dbgroup
-mirroruid 83:5A:20:90:01:06:50:01:00:00:00:00:00:00:00:00
```

For SP `ss1_spa`, owner of the consistency group named `dbgroup`, this command removes the mirror with the unique ID shown.

---

**Note:** To use this command with `navicli.jar`, replace `naviseccli` with `java -jar navicli.jar`.

#### Output

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 `stdout`. You may also receive other error messages.
mirror -async -change group

Changes consistency group properties

**Description**
The `naviseccli` or `navicli.jar mirror -async` command with `-change group` changes one or more parameters of a consistency group.

You must specify at least one of the optional switches. If you omit the override option, then it displays a message similar to the following for each parameter to confirm the change.

Do you want to change the group synchronization rate from Medium to High (y/n)?

To change a parameter, enter y.

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

**Format**
`mirror -async -change group` is used with `naviseccli` (described on page 2-10) or `navicli.jar` (described on page 2-17) as follows:

```
mirror -async -change group -name name | -groupid id
[-description description] [-manualupdate | -delay start minutes | -end delay minutes] [-sync rate] [-recovery policy policy] [-o]
```

where:

- **-name name**
  Specifies the name of the consistency group you want to change. You can specify either the consistency group name or ID (next).

- **-group id**
  Specifies unique ID of the consistency group you want to change.

- **-description description**
  Specifies a textual description for the consistency group.

- **-manual update**
  Specifies that you must explicitly start each update of the group.
-**delaystart** minutes

Specifies the time (in minutes) from the start of the previous update until the start of the next one. If the previous update has not completed, the next update is delayed until the update in progress has completed.

-**endedelay** minutes

Specifies the time (in minutes) from the end of the previous update until the start of the next one. Specifying a value of 0 results in the group being updated as fast as possible.

-**syncrate** rate

Specifies a relative value (low, medium, or high) for the priority of completing updates. Values for rate are low, medium, or high. High completes updates faster, but may significantly affect storage system performance for server I/O requests. Low completes updates slower, but also minimizes the impact on other storage system operations.

-**recoverypolicy** policy

Specifies the policy for recovering the mirrors in a group after a failure causes a system fracture. Values are:

manual Update of the secondary image does not begin until you explicitly issue a Synchronize command.

auto Update of the secondary image automatically begins as soon as the connection between the two storage systems is again operational.

-o

Executes the command without prompting for confirmation.

**Conventions and recommendations**

None.

**Example**

```
naviseccli -h server1 mirror -async -changegroup -name dbgroup -syncrate high
```

For storage system server1, consistency group dbgroup, this command changes the synchronization rate to high.

**Note:** To use this command with navicli.jar, replace naviseccli with java -jar navicli.jar.
Output  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 stdout. You may also receive other error messages.

Use the -listgroups command on the mirror to confirm that the changes were made.
**mirror -async -syncgroup**

Starts an update of the consistency group

**Description**
The `naviseccli` or `navicli.jar` `mirror -async` command with `-syncgroup` begins the synchronization operation on a consistency group. All the images in the consistency group must be primary.

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

**Format**
`mirror -async -syncgroup` is used with `naviseccli` (described on page 2-10) or `navicli.jar` (described on page 2-17) as follows:

```
mirror -async -syncgroup -name name| -groupid id
[-synctype startnow] [-o]
```

where:

- **-name name**
  Specifies the name of the consistency group you want to update or synchronize. You can specify either the consistency group name or id (next).

- **-groupid id**
  Specifies the unique ID of the consistency group you want to update or synchronize.

- **-synctype startnow**
  Starts synchronization immediately. `startnow` is the only currently supported option.

- **-o**
  Executes the command without prompting for confirmation.

**Conventions and recommendations**
None.
Example  

naviseccli -h ss1_spa mirror -async -syncgroup -name dbgroup -synctype startnow

For SP ss1_spa, owner of the consistency group named dbgroup, this command starts synchronizing the consistency group immediately.

Note: To use this command with navicli.jar, replace naviseccli with java -jar navicli.jar.

Output  

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 stdout. You may also receive other error messages.
mirror -async -fracturegroup

Fractures all image(s) in a consistency group

Description
The naviseclli or navicli.jar mirror -async command with -fracturegroup starts an administrative fracture of all images in the consistency group, even if they are already system fractured. The consistency group cannot be in a scrambled state, and all images must be primary. If the consistency group is in the process of updating, it stops.

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

Format
mirror -async -fracturegroup is used with naviseclli (described on page 2-10) or navicli.jar (described on page 2-17) as follows:

mirror -async -fracturegroup -name name | -groupid id
[-fracturetype abort] [-o]

where:

-name name
Specifies the name of the consistency group you want to fracture. You can specify either the consistency group name or ID (next).

-groupid id
Specifies the unique ID of the consistency group you want to fracture.

-fracturetype abort
Specifies that you want to abort an update if one is in progress. The secondary retains any extents that have been copied to the secondary, and therefore do not need to be resent on the next synchronization.

-o
Executes the command without prompting for confirmation.

Conventions and recommendations
None.
Example

```
navisecli -h ss1_spa mirror -async -fracturegroup
-name dbgroup
```

For SP `ss1_spa`, with the consistency group named `dbgroup`, this command fractures all the images in the consistency group.

---

**Note:** To use this command with `navicli.jar`, replace `navisecli` with `java -jar navicli.jar`.

---

Output

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 `stdout`. You may also receive other error messages.
mirror -async -promotegroup

Promotes all secondary images in a consistency group to primary images

Description

The naviseccli or navicli.jar mirror -async command with -promotegroup promotes all secondary images to the role of the primary images.

User access

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

Format

mirror -async -promotegroup is used with naviseccli (described on page 2-10) or navicli.jar (described on page 2-17) as follows:

`mirror -async -promotegroup -name name | -groupid id [-type type] [-o]`

where:

- **-name name**
  
  Specifies the name of the consistency group you want to promote. You can specify either the consistency group name or ID (next).

- **-groupid id**
  
  Specifies the unique ID of the consistency group you want to promote.

- **-type type**
  
  Specifies the type of promotion. Values are:

  **normal** (default), which causes a failure if the connection to the primary storage system is not functioning or if at least one set of the images to promote are not synchronized with the primaries. Otherwise, it will destroy the original mirrors and create new ones with the targeted secondaries as primaries and the original primaries as secondaries.

  **local**, which does not attempt to add back the original primaries as secondaries, but always creates new mirrors with only primary images consisting of the targeted secondaries.

  **oos** (out-of-sync), which attempts a normal promotion, reversing the original roles of primaries and secondaries even if they are not all synchronized. If the connection to the primary storage system is not functioning, a local promotion will occur.
-o

Executes the command without prompting for confirmation.

**Conventions and recommendations**

None

**Example**

```bash
naviseccli -h ss1_spa mirror -async -promotegroup -name dbgroup
```

For SP `ss1_spa`, owner of the consistency group named `dbgroup`, this command promotes all secondary images.

**Note:** To use this command with `navicli.jar`, replace `naviseccli` with `java -jar navicli.jar`.

**Output**

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 `stdout`. You may also receive other error messages.
MirrorView/A (consistency group) commands

**mirror -async -listgroups**

Displays information about existing consistency groups

**Description**
The `naviseccli` or `navicli.jar mirror -async` command with `-listgroups` and no other switches lists all information for all asynchronous consistency groups on the storage system. Issuing this command with one or more options displays the information based on the option(s) issued.

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

**Format**
`mirror -async -listgroups` is used with `naviseccli` (described on page 2-10) or `navicli.jar` (described on page 2-17) as follows:

```
```

where:

- **-name [name]**
  Specifies the name of the consistency group whose information you want to display. If you do not specify the name of the consistency group with this switch, that piece of information will be provided for the consistency group(s) being displayed.

- **-groupid [id]**
  Specifies the unique ID of the consistency group whose information you want to display.

- **-feature**
  Displays the maximum number of consistency groups allowed on the storage system and the maximum number of members per consistency group.

- **-update**
  Displays the type of update as either manual, **Time elapsed since start of last update** (in minutes), or **Time elapsed since end of last update** (in minutes). On the next line it also displays the Update Period in minutes. If Update Period is not applicable, it displays **N/A**.
-**nextupdate**

Displays the number of minutes until the next update.

-**syncrate**

Displays the rate at which data is transferred during updates and initial synchronizations.

-**state**

Displays Synchronized, Out-of-Sync, Synchronizing, Consistent, Rollingback, Quasi-Consistent, Scrambled, Incomplete, Local Only, or Empty.

-**role**

Displays Primary, Secondary, or Unknown.

-**mirrors**

Displays the mirror name, mirror uid, the IDs of both images in the pair, and the storage system ID of the paired image.

-**condition**

Displays the consistency group condition as Normal, Initializing, Updating, Admin Fractured, System Fractured, Waiting on Admin.

-**recoverypolicy**

Displays either manual or automatic to describe whether the system will wait for you to restart updates in the event of a system fracture, after the problem is corrected.

-**description**

Displays the description associated with the consistency group.

---

**Conventions and recommendations**

None.
Example

naviseccli -h 10.14.5.200 mirror -async -listgroups

Group Name: taraFARGroup
Group ID: 50:06:01:60:90:20:5A:83:00:00:00:00
Description:
Maximum Number of Groups Allowed: 16
Maximum Number of Mirrors per Group: 16
Update Type: Manual
Time to next update: 0
State: Quasi Consistent
Role: Secondary
Mirror Name: taraFAR202
Mirror WWN: C6:05:20:88:60:01:06:50:08:00:00:00:00:00:00:00
Primary WWN: 50:06:01:60:88:20:05:C6
Array hosting remote image: 50:06:01:60:88:20:05:C6
Mirror Name: taraFAR201
Mirror WWN: C6:05:20:88:60:01:06:50:07:00:00:00:00:00:00:00
Primary WWN: 50:06:01:60:88:20:05:C6
Array hosting remote image: 50:06:01:60:88:20:05:C6
Mirror Name: taraFAR200
Mirror WWN: C6:05:20:88:60:01:06:50:06:00:00:00:00:00:00:00
Primary WWN: 50:06:01:60:88:20:05:C6
Array hosting remote image: 50:06:01:60:88:20:05:C6
Condition: Admin Fractured - Intervention required
Sync Rate: Medium
Recovery Policy: Manual

Group Name: HWLocalPromoteTestGroup
Group ID: 50:06:01:60:90:20:5A:83:01:00:00:00
Description:
Maximum Number of Groups Allowed: 16
Maximum Number of Mirrors per Group: 16
Update Type: Manual
Time to next update: 0
State: Synchronized
Role: Secondary
Mirror Name: HWLocalPromoteTestMirror
Mirror WWN: C6:05:20:88:60:01:06:50:09:00:00:00:00:00:00:00
Primary WWN: 50:06:01:60:88:20:05:C6
Array hosting remote image: 50:06:01:60:88:20:05:C6
Condition: Normal
Sync Rate: Medium
Recovery Policy: Manual
MirrorView/A (consistency group) commands

Note: To use this command with navicli.jar, replace naviseccli with java -jar navicli.jar.

Output 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 stdout. You may also receive other error messages.
MirrorView/A (consistency group) commands
This appendix describes how MirrorView/A and consistency groups handle failures.

Topics include:
- How MirrorView/A handles failures ............................................A-2
- How consistency groups handle failures ....................................A-13


How MirrorView/A handles failures

When a failure occurs during normal operations, MirrorView/A lets you perform several actions to recover. In recovering from failures, MirrorView/A achieves two goals:

◆ Preserves data integrity
◆ Minimizes the amount of time that data is unavailable to the user

Access to the primary SP fails

If an SP that owns mirrored LUNs on the primary system fails, MirrorView/A on the other SP takes ownership of those mirrored LUNs by trespassing them when something on the server (like PowerPath) initiates the trespass. This allows mirroring to continue, provided the server is set up properly to handle the failover (for example, a Windows server with PowerPath). When the primary LUN is trespassed, MirrorView/A sends a trespass request to any secondary images when the next update starts. Therefore, you may notice that the mirrored LUNs on the secondary system have moved from SP A to SP B, or vice versa. MirrorView/A keeps the SP ownership the same on the primary and secondary systems during updates. If the primary image is on SP A, then the secondary image will be on SP A. This may not occur until the start of the next update.

Primary image fails

If the storage system controlling the primary image fails, access to the mirror stops until you either repair the storage system or promote a secondary image of the mirror to primary.

Promoting a secondary image to a primary image

In situations where you must replace the primary storage system due to a catastrophic failure, you can use a promotion to access data on the secondary storage system. To recover and restore I/O access, you must promote a secondary mirror image to the role of primary mirror image, so that a server can access it.

Note: You can also promote a secondary image even if there has not been a catastrophic failure.
If the primary image and secondary image can communicate with each other, then when the secondary image is promoted, the former primary image is demoted to a secondary image.

To promote a secondary image, the following conditions must be true:

- You must direct the `navicli.jar mirror` commands to the storage system holding the secondary image.
- The state of the secondary image you will promote must be either Consistent or Synchronized.
- An update is not currently transferring data for this mirror.

**CAUTION**

Promoting a secondary image will cause loss of data written to the primary after the start of the last completed update. If any updates have been made to the primary image since that time, a full resynchronization of the mirror will be required after the promotion. Also, if an update is currently active (that is, transferring data), the promotion will not be allowed; allow the update to complete and the image to transition into the Synchronized state; then perform the promotion. An alternative to allowing the update to complete is to fracture the image.

In a failure situation, before promoting a secondary image to a primary image:

1. If the existing primary image is accessible, remove the primary image from any storage groups before promoting the secondary image to avoid I/O and therefore inconsistent data.
2. Ensure that no I/O, either generated from a server or by an update in progress, is occurring in the asynchronous mirror.
3. If the existing primary is available, make sure that it lists the secondary image that is to be promoted as "synchronized."

To promote a secondary image to a primary image:

1. Issue the `mirror -async -promoteimage` command. See “`mirror -async -promoteimage`” on page 2-46.

**Note:** If you do not specify the `-type` switch, the command performs a normal promote.
2. If the original primary storage system failed, remove the primary storage system from the domain.

3. Add the newly promoted image to a storage group if necessary.

At some point later, you can also perform the following steps:

1. Verify that the failed storage system is not the master of the domain.
   
   If it is, assign another storage system to be the master. See the *EMC Navisphere Command Line Interface (CLI) Reference*.

2. Verify that the failed storage system is not a portal.
   
   If it is a portal, remove the portal and configure a different storage system as a portal. See the *EMC Navisphere Command Line Interface (CLI) Reference*.

**Example promote scenarios**

The following scenarios illustrate examples of promoting a mirror.

**Scenario 1**

You attempt to promote a mirror that has a secondary image, but the connection between the storage system is not working. The secondary image indicates that it is synchronized, when it is actually system fractured and consistent. An error, *Existing image unreachable*, appears. You can investigate the reason for the loss of connectivity and correct the problem before continuing with the promotion, or you can select the **Local Only Promote** option to complete the promotion.

If you select **Local Only Promote**, the software promotes the local mirror and attempts to contact the original primary image and remove the promoted image from the mirror. In the case described here, it cannot contact the other storage system, so it converts the local image to a primary image in a mirror with no secondary images.

---

**Note:** In this scenario, a **Force Promote** has exactly the same effect as the **Local Only Promote** operation.

Since the software cannot contact the remote storage system, the original mirror still exists on the storage system originally hosting the primary image. However, even if connectivity is restored, any attempt to start an update will fail (since the secondary has been promoted), and the secondary image will remain administratively
fractured forever. You should use **Force Destroy** to remove this original mirror.

**Scenario 2**
You attempt to promote a mirror whose secondary image is in the **Consistent state.** An error, **Existing primary will be out-of-sync,** appears. If possible, allow the secondary to go to the **Synchronized state** (for example, stop application I/O to the primary image, flush data from the server, start an update and wait for it to complete). You can then promote the secondary without requiring a full resynchronization. Otherwise, you can select either the **Force Promote** or the **Local Only Promote** option to continue the promotion. In either case, you must perform a full resynchronization before you again have the mirror providing protection for your data.

**Running MirrorView/A on a VMware ESX Server**
When you use MirrorView/A on a VMware ESX Server, after you promote the secondary image to a primary, perform the following steps:

1. Assign the newly promoted primary image to a storage group of the same or standby ESX Server.
2. Rescan the bus at the ESX Server level.
3. Create a Virtual Machine (VM) on the same or standby ESX Server.
4. Assign the newly promoted primary to the VM.
   Assign it to a different VM unless you remove the failed primary, in which case you can assign it to the same VM.
5. Power up the VM.

If the VM is created and running, perform these steps:

1. Power it down.
2. Use the Service Console on the ESX Server to assign the newly promoted primary to the powered-down VM.
3. Power up the VM.

The primary image (which is now the secondary image) will not be accessible to the primary ESX Server.
Responding to failures

Recovering by promoting a secondary image

When you promote the secondary image, the software assigns a new mirror ID to the promoted image to distinguish it from the old mirror, even though the mirrors have the same name. The new image condition of the old primary image depends on whether the old primary image is accessible at the time of promotion.

If the existing primary image is accessible when you promote, the software attempts to add the old primary image as a secondary image of the promoted mirror; that is, the images swap roles.

If the primary image is *not* accessible when you promote, the software creates a new mirror with the former secondary image as the new primary image and no secondary image, as shown in the example below. The mirror on the original primary storage system does not change, and so continues to have a stale record of the former secondary. You must remove the original mirror with the `-mirror -async -destroy -force` command once the original primary storage system is available again.

<table>
<thead>
<tr>
<th>Mirror before promotion</th>
<th>Mirror after promotion</th>
</tr>
</thead>
<tbody>
<tr>
<td>mirror ID = aaa</td>
<td>mirror ID = bbb</td>
</tr>
<tr>
<td>primary image = LUN xxxx</td>
<td>primary image = LUN yyyy</td>
</tr>
<tr>
<td>secondary image = LUN yyyy</td>
<td>secondary image = none</td>
</tr>
</tbody>
</table>

Restoring the original mirror configuration after recovery of a failed primary image

If the original mirror becomes accessible following a failure and the mirror’s secondary image has been promoted, the old mirror will be unable to communicate with the new one. To restore your mirrors to their original configuration, do the following:

1. If you want to retain any data on the original primary LUN, copy it to another LUN before continuing, or alternatively, you can create a LUN that will become the primary LUN. The following process overwrites the contents of the original primary LUN.

2. Remove the original primary LUN from any storage groups of which it is a member.
3. Destroy the original mirror using the `-mirror -async -destroy -force` command.

<table>
<thead>
<tr>
<th>Original mirror</th>
<th>New mirror</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old mirror is destroyed. Original LUN used for primary image remains (LUN xxxx)</td>
<td>primary image = LUN yyyy secondary image = none</td>
</tr>
</tbody>
</table>

4. Add a secondary image to the new mirror using the LUN that was the primary image for the original mirror (LUN xxxx).

**CAUTION**

Data from the promoted LUN will overwrite all the data in the secondary image (original primary) LUN.

The secondary image synchronizes automatically. Allow the synchronization to complete.

<table>
<thead>
<tr>
<th>New mirror</th>
</tr>
</thead>
<tbody>
<tr>
<td>primary image = LUN yyyy secondary image = LUN xxxx</td>
</tr>
</tbody>
</table>

**CAUTION**

Allow the image to transition to the Synchronized state following the synchronization. If the image is in the Consistent state when you promote it, another full resynchronization is required, and data may be lost.

5. Promote the secondary image (LUN xxxx) in the new mirror to primary.

If you attempt promotion and the system indicates that the resulting mirror would be out-of-sync, do not complete the promotion. Instead, determine why the images are potentially different. If necessary, start an update of the mirror, wait for it to complete, and then for the secondary image to transition to the Synchronized state. Then you can retry the promotion.
The new mirror has the same configuration as the original mirror.

<table>
<thead>
<tr>
<th>New mirror</th>
</tr>
</thead>
<tbody>
<tr>
<td>primary image = LUN xxxx</td>
</tr>
<tr>
<td>secondary image = LUN yyyy</td>
</tr>
</tbody>
</table>

During a promotion, the recovery policy for a secondary image is always set to manual recovery. This prevents a full synchronization from starting until you want it to.

6. If required, reset the recovery policy back to automatic.

**Recovering without promoting a secondary image**

If the primary storage system fails, but can be readily repaired, recovery is simpler. MirrorView/A records any writes that had completed before the failure and transfers them to the remote image when the next update occurs. Any writes that were sent to the storage system but not yet acknowledged may be lost, but application-specific recovery techniques, such as *chkdsk* or *fsck* for filesystems, can usually correct any issues.

To recover without promoting a secondary image, follow these steps:

1. Repair the primary storage system and/or server.
2. Fracture the asynchronous mirror(s).
3. Complete any necessary application-specific recovery of the data on the primary image.
4. Make sure that the data is flushed from the server to the storage system.
5. Synchronize the asynchronous mirror(s).

**Failure of the secondary image**

When a primary image cannot communicate with a secondary image, it marks the secondary as *unreachable* and stops updating the secondary image. The secondary image is marked System Fractured. The loss of communication may be due to a link between storage systems, an SP failure on the secondary storage system, or some other failure on the secondary storage system. In the event of the communication failure, the secondary image remains a member of the mirror.
If the mirror is set for automatic recovery, an update automatically starts once the secondary storage system is again accessible. Otherwise, you must manually start the update.

---

**Promoting a secondary image when there is no failure**

You may want to promote your secondary image even if no failure occurs on the storage systems. For example, you may want to test your disaster recovery procedure before a real failure occurs, or perhaps the server attached to the primary storage system failed, and you must resume operations using the server attached to the secondary storage system.

If the original primary is accessible when you promote the secondary, the software verifies whether the images are identical. If possible, the secondary image should be in the Synchronized state (stop application I/O, flush data from the servers, start and update and wait for it to complete). If the images are identical, they swap roles, resynchronization is not necessary, and the promotion is complete. If the images are potentially different (that is, the secondary image is not in the Synchronized state), then you must specify the type of promotion. (See page 2-46.) As part of a promotion, any secondary images in the new mirror are set to manual recovery.

<table>
<thead>
<tr>
<th>Mirror before promotion</th>
<th>Mirror after promotion</th>
</tr>
</thead>
<tbody>
<tr>
<td>mirror ID = aaa</td>
<td>mirror ID = bbb</td>
</tr>
<tr>
<td>primary image = LUN xxxx</td>
<td>primary image = LUN yyyy</td>
</tr>
<tr>
<td>secondary image = LUN yyy</td>
<td>secondary image = LUN xxxx</td>
</tr>
</tbody>
</table>

If the images are not synchronized, you can specify to forcibly promote (oos), local promote, or to not promote. oos promote and local promote require a full resynchronization of the data before mirrored protection is again in effect.
Summary of MirrorView/A failures

Table A-1 shows how MirrorView/A might help you recover from system failure at the primary and secondary sites. It assumes that the secondary image of the mirror is in either the Synchronized or Consistent state.

Table A-1  Recovery from system failures

<table>
<thead>
<tr>
<th>Event</th>
<th>Result and recovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss of access to primary image LUN</td>
<td>Check connections between server and storage system, including zoning and correct operation of any switches. Check for an SP reboot.</td>
</tr>
<tr>
<td>Secondary SP is rebooted</td>
<td>If the secondary SP reboots, for example, due to a software failure, an explicit command or the SP is physically pulled and reseated, you may see the secondary image become system fractured. It is also possible for the secondary to become administratively fractured, in which case simply synchronize the image.</td>
</tr>
<tr>
<td>Server accessing primary image fails</td>
<td>Catastrophic failure. I/O stops. After two more update periods, if the secondary image is in the Consistent state, it will transition to the Synchronized state. Nothing more happens until the server is repaired or replaced, or a secondary image is promoted.</td>
</tr>
</tbody>
</table>
| Storage system running primary image fails | Option 1 - Catastrophic failure. The mirror is left in the state it was already in. If the secondary image is in either the Consistent or Synchronized state, it may be promoted to provide access to your data. **Note:** Any writes to the primary since the start of the last completed update are not propagated to the secondary.  
Option 2 - Noncatastrophic failure, repair is feasible.  
The administrator has the problem fixed, and then normal production operation can resume.  
Option 3 - Only one SP fails.  
If the SP that controls the mirror fails, software on the server (for example, PowerPath) can detect the failure. This software can then cause control of the mirror to be transferred to the surviving SP, and normal operations can continue. If such software is not running on the server, then you must either manually transfer control using Navisphere, or access to the mirror stops until the SP is back in service.  
If the SP that does not control the LUN fails, then mirroring continues as normal. |
Responding to failures

Storage system running secondary image fails
- If the SP that does not control the secondary image fails, nothing happens with respect to this mirror.
- If the SP that controls the mirror fails (or both SPs fail or a catastrophic failure of the entire system occurs), the secondary image of the mirror will become system fractured at the start of the next update. If the mirror is appropriately configured, this will cause the mirror to go into the Attention state, although I/O will continue as normal to the primary image.

The administrator has a choice: If the secondary can easily be fixed (for example, if someone pulled out a cable), then the administrator can have it fixed and let things resume. Otherwise, to regain protection of your data and you have another storage system available, you can force destroy the existing mirror, recreate it, and add a secondary image on another working storage system. Protection is not established until the full synchronization of the secondary image completes.

Loss of connection between storage systems (indicated by system fractures)
Check the cables, make sure that all SPs are still working, and make sure the MirrorView path between the storage systems is still enabled and active. Check correct zoning and other function of any switches.

Failures when adding secondary images
Make sure that:
- The connection between storage systems works.
- You are managing both storage systems (which may require managing two domains).
- The secondary LUN is available and the same size as the primary image.
- The secondary image LUN is not in the storage group.
- The secondary image LUN is not already a secondary image, of either a synchronous or asynchronous mirror.
- The secondary LUN is not part of a clone group.
- The reserved LUN pools on both primary and secondary storage systems are adequately configured.
- The secondary LUN is not set up as a destination for SAN Copy.

When the secondary image does not synchronize
Make sure that:
- The connection between the storage systems is still good.
- The recovery policy is set to auto, not manual.
- The secondary SP is working.
Try manually fracturing and then manually synchronizing the image.

---

**Table A-1 Recovery from system failures (continued)**

<table>
<thead>
<tr>
<th>Event</th>
<th>Result and recovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage system running secondary image fails</td>
<td>• If the SP that does not control the secondary image fails, nothing happens with</td>
</tr>
<tr>
<td></td>
<td>respect to this mirror.</td>
</tr>
<tr>
<td></td>
<td>• If the SP that controls the mirror fails (or both SPs fail or a catastrophic</td>
</tr>
<tr>
<td></td>
<td>failure of the entire system occurs), the secondary image of the mirror will</td>
</tr>
<tr>
<td></td>
<td>become system fractured at the start of the next update. If the mirror is</td>
</tr>
<tr>
<td></td>
<td>appropriately configured, this will cause the mirror to go into the Attention</td>
</tr>
<tr>
<td></td>
<td>state, although I/O will continue as normal to the primary image.</td>
</tr>
<tr>
<td></td>
<td>The administrator has a choice: If the secondary can easily be fixed (for example,</td>
</tr>
<tr>
<td></td>
<td>if someone pulled out a cable), then the administrator can have it fixed and let</td>
</tr>
<tr>
<td></td>
<td>things resume. Otherwise, to regain protection of your data and you have another</td>
</tr>
<tr>
<td></td>
<td>storage system available, you can force destroy the existing mirror, recreate it,</td>
</tr>
<tr>
<td></td>
<td>and add a secondary image on another working storage system. Protection is not</td>
</tr>
<tr>
<td></td>
<td>established until the full synchronization of the secondary image completes.</td>
</tr>
<tr>
<td>Loss of connection between storage systems</td>
<td>Check the cables, make sure that all SPs are still working, and make sure the</td>
</tr>
<tr>
<td>(indicated by system fractures)</td>
<td>MirrorView path between the storage systems is still enabled and active. Check</td>
</tr>
<tr>
<td></td>
<td>correct zoning and other function of any switches.</td>
</tr>
<tr>
<td>Failures when adding secondary images</td>
<td>Make sure that:</td>
</tr>
<tr>
<td></td>
<td>• The connection between storage systems works.</td>
</tr>
<tr>
<td></td>
<td>• You are managing both storage systems (which may require managing two domains).</td>
</tr>
<tr>
<td></td>
<td>• The secondary LUN is available and the same size as the primary image.</td>
</tr>
<tr>
<td></td>
<td>• The secondary image LUN is not in the storage group.</td>
</tr>
<tr>
<td></td>
<td>• The secondary image LUN is not already a secondary image, of either a synchronous</td>
</tr>
<tr>
<td></td>
<td>or asynchronous mirror.</td>
</tr>
<tr>
<td></td>
<td>• The secondary LUN is not part of a clone group.</td>
</tr>
<tr>
<td></td>
<td>• The reserved LUN pools on both primary and secondary storage systems are</td>
</tr>
<tr>
<td></td>
<td>adequately configured.</td>
</tr>
<tr>
<td></td>
<td>• The secondary LUN is not set up as a destination for SAN Copy.</td>
</tr>
<tr>
<td>When the secondary image does not synchronize</td>
<td>Make sure that:</td>
</tr>
<tr>
<td></td>
<td>• The connection between the storage systems is still good.</td>
</tr>
<tr>
<td></td>
<td>• The recovery policy is set to auto, not manual.</td>
</tr>
<tr>
<td></td>
<td>• The secondary SP is working.</td>
</tr>
<tr>
<td></td>
<td>Try manually fracturing and then manually synchronizing the image.</td>
</tr>
</tbody>
</table>
Responding to failures

Recovering from serious errors

In the unlikely event that the mechanism for tracking changes made to the primary image fails (for example, insufficient memory available on the SP), the secondary image is marked as permanently fractured. To recover from this situation, you must remove the secondary image from the mirror, and then re-add it (which does a full resynchronization). This failure may indicate that you are using close to the storage system’s capacity for layered features.

Some other serious failures will transition MirrorView/A into a degraded mode of operation, where administrative requests will be rejected and no further updates run. Degraded mode affects only a single SP; the other SP in the storage system may continue to run normally (depending on the nature of the failure).

When an SP enters degraded mode, the system logs an event that indicates why MirrorView/A is in the degraded mode. Usually you can recover from the degraded mode by simply rebooting the affected SP, but some specific cases require you to check other components that MirrorView/A uses before rebooting the SP. Table A-2 lists various scenarios in which MirrorView/A goes to the degraded mode and the recovery options you can take.

<table>
<thead>
<tr>
<th>Event</th>
<th>Result and recovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal memory corruption</td>
<td>Mirror data does not match the expected value. Reboot the SP.</td>
</tr>
<tr>
<td>Serious, unexpected errors</td>
<td>MirrorView/A receives unexpected errors from its underlying components during operation. Check the event log for a record of errors, and take steps to correct them. For example, if the reserved LUN pool LUNs are faulted, recover the reserved LUN pool LUNs. Then reboot the SP.</td>
</tr>
<tr>
<td>Internal fracture failure</td>
<td>A fracture operation fails due to reasons other than an error you made. Check the event log for the appropriate failure reason. Reboot the SP to fix the problem.</td>
</tr>
</tbody>
</table>
How consistency groups handle failures

When a failure occurs during normal operations for consistency groups, MirrorView/A lets you perform several actions to recover. When recovering from failures, MirrorView/A achieves three goals:

- Preserves data integrity
- Minimizes the amount of time that data is unavailable to the user
- Ensures that the consistency of the consistency group is maintained

Access to the SP fails

Consider a consistency group that has member mirrors, some of which SP A controls and some of which SP B controls. If SP A on the primary storage system fails, then software on the attached server, for example, PowerPath, moves control of the mirrors that were controlled by SP A to SP B. This allows applications on the server, as well as the mirroring of data to the secondary storage system, to continue uninterrupted. However, as part of the transfer of control, the consistency group becomes system fractured. If the recovery policy is set to automatic, an update automatically starts on the surviving SP (SP B in this example). However, if the recovery policy is manual, you must manually start an update.

Primary storage system fails

If the storage system running the primary consistency group fails, access to the data in the group’s member LUNs is lost. You can either repair the failed storage system and then continue operations, or you can promote the secondary consistency group, so as to access the data from the secondary storage system.

Recovering by promoting a secondary consistency group

As part of the consistency group promotion each of the mirror members is promoted. This section describes three types of group promotions, which are based on the connectivity status between the primary and the secondary and the condition of the individual members. For more information about promoting a consistency group, see Chapter 3.
Responding to failures

Normal promotion
When you request promotion for a secondary image, the software determines if connectivity exists between the storage systems hosting the primary and secondary consistency groups. If connectivity is working, it tests the members of the group to determine if the result of promotion will be an out-of-sync group or a synchronized. The promote operation will fail if the primary is unreachable or the secondary group will be out-of-sync after promotion. You then can do a **local only** promote or a **force** promote, described below.

Force promote
A **force promote** promotes each member of the group and places the newly promoted mirrors in the group (removing the original members). If the original primary storage system is available, the original primary images will become secondary images of the promoted mirrors. The promoted group is marked as Out-of-Sync and its recovery policy is set to **manual**. You must initiate an update for the group in order to start the full update, which is required for the group to be once again protecting your data. If the original primary storage system is unavailable, **Force Promote** has the same effect as **Local Only Promote**, described below.

**Important:** You must perform a full update on the new secondary image, which will overwrite all existing data on that image

Local only promote
A **local only** promote promotes the secondary image of each consistency group member to a primary image, but does not attempt to add the original primary image or any other existing secondary images to the promoting mirror. If a connection exists between the primary and the secondary, for each member of the primary, the software attempts to remove the image being promoted on the secondary. Thus, the old primary consistency group will have all primary images and no secondary images. If no connection exists, the promote will still continue on the secondary, and the operation will not fail. The original primary consistency group cannot communicate with the promoted secondary consistency group even if the MirrorView/A connection between the storage systems is restored (since the secondary consistency group was promoted to a primary consistency group).

If a failure occurs during promoting (for example, an SP reboots), the consistency group may be left in an inconsistent state. It is possible that some members have only primary images or some have been
promoted or not promoted at all. Check the state of the promoted consistency group to detect any problems during promotion.

A consistency group is in the **scrambled** state if at least one of its member’s primary images is missing its corresponding secondary images. Note: Table A-3 lists the configurations in which the scrambled state can occur.

**Note:** Either the **Local Only Promote** or the **Force Promote** operation can result in a consistency group that contains mirrors that have no secondary images at all. In this case, the consistency group is no longer performing its function. The best way to correct this is to remove the mirrors from the consistency group, add secondary images as required, and add the mirrors back to the group.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Ways to reach this state</th>
<th>Recovery options</th>
<th>Is the consistency group promotable in this state?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consistency group members consist of only primary images. Individual members do not have any secondary images associated with them.</td>
<td>After a local promotion.</td>
<td>Force the removal of each member from the consistency group, add secondary images to each mirror, and add the mirrors in the consistency group again.</td>
<td>No, because there are no secondary images associated with the consistency group.</td>
</tr>
<tr>
<td>If a normal promotion or a force promotion fails in mid-operation, some members may consist of only primary images. The remaining members are successfully promoted and have secondaries associated with them.</td>
<td>After a failed normal promotion or out-of-sync promotion. Failure can occur by pulling the SP to which the promotion was directed.</td>
<td>Force the removal of the members with no secondary image, and then add secondaries to those mirrors. Add them back into the consistency group as necessary.</td>
<td>The consistency group is not promotable from the old primary until you remove the consistency group members that lack a secondary image. However, you can issue a local promotion on the old primary in this case.</td>
</tr>
<tr>
<td>If any type of promotion fails in mid-operation, some members may consist of only primary images. The remaining members are not successfully promoted.</td>
<td>After a promotion fails on the local SP before you attempt a remote promotion.</td>
<td>Force the removal of the members with no secondary image, add secondaries to those mirrors, and add them back into the consistency group as necessary.</td>
<td>Not until you remove the consistency group members that lack a secondary image. You can issue a <strong>Force Promote</strong> again in order to promote the mirrors that were not promoted.</td>
</tr>
</tbody>
</table>
**Failure of the secondary consistency group**

When a primary cannot communicate with a secondary consistency group, the group’s condition changes to system fractured. When a consistency group is system fractured, no writes are propagated to the secondary consistency group. The primary storage system attempts to minimize the amount of work required to synchronize the secondary after it recovers. It keeps track of the write requests to the consistency group, so that only modified areas will be copied to the secondary during recovery. Also, consider the case where the consistency group has some members whose primary image LUNs reside on SP A and some on SP B. If the MirrorView/A connection is broken between SP Bs of the primary and the secondary storage system, the consistency group is system fractured to maintain the consistent state of the consistency group.
This chapter explains the **navicli** command. These commands let you use MirrorView/A software to create a byte-for-byte copy of one or more local LUNs connected to a distant storage-system server.

---

**Note:** The **navicli** commands do not require Navisphere Manager to be installed on the storage system.

Topics include:

- navicli.......................................................... B-2
- Using navicli with MirrorView/A commands ....................... B-5
Navicli command

**Navicli**

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

**Note:** Before you can use Navisphere MirrorView/A 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.

**Description**
The CLI navicli command sends storage system management and configuration requests to a Navisphere Agent running on a storage system, either directly or (on older systems) via an Agent running on a local or remote server. Note that this server-based CLI differs from the web-based CLI used for SnapClone management.

**User access**
Anyone that can log in to the server running the Navisphere CLI.

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

```
navicli -help
```

or

```
```

where the navicli switches are:

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

- **-f filename**
  
  Specifies to store data in a file.

- **-h hostname**
  
  Specifies the hostname of the storage-system SP or server. For a CX3-series and CX-series storage system, each SP is a host, therefore the hostname is the IP address or network name of the destination SP.

- **-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.

-q

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

-t timeout

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

-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.
**Navicli command**

**Spaces in arguments**

Normally, each argument to a CLI command is composed 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 timeout values respectively if you omit the `-h`, `-d`, and `-t` switches from the `navicli` command. If you omit both `RaidAgentDevice` and the `-d` switch, the CLI will show an error on any command that needs device information. If you do not set `RaidAgentTimeout` is and omit the `-t` switch, the default timeout value is 25 seconds.

**Examples**

The following examples show sample environment variable settings:

```
set RaidAgentDevice  \.A-9162244 (Windows example)
set RaidAgentHost  server1 (or for CX/FC4700, ss1_spa )
set RaidAgentTimeout  45
```

**Error codes**

The navicli command might return any of a numeric sequence of errors explained in Appendix A of the CLI Reference manual.
Using navicli with MirrorView/A commands

You use the navicli command with the mirror command followed by a MirrorView/A command. For information about the MirrorView/A CLI functions, see Chapter 2.

An example of using the navicli command with a mirror -async command is the -create function. The navicli mirror -async command with -create creates a new mirror using an existing LUN.

The format for this command is:

```
mirror -async -create -name name -lun lun
[description description] [-syncdelay delay]
[-requiredimages num_of_images] [-o]
```

An example of this command follows:

```
navicli -h ss1_spa mirror -async -create -name db_mirror -lun 112
```

The mirror -async -create command creates the remote mirror named db_mirror to mirror LUN 112. The mirror automatically activates.
This appendix lists the MirrorView/A error codes returned by the CLI. The numbers proceed sequentially.

Toics include:

- MirrorView/A command error codes .................................................. C-1
## MirrorView/A command error codes

<table>
<thead>
<tr>
<th>Error value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>125</td>
<td>Invalid SP name.</td>
</tr>
<tr>
<td>127</td>
<td>Package number invalid.</td>
</tr>
<tr>
<td>128</td>
<td>This version of FLARE software does not support nondisruptive software installation (NDU).</td>
</tr>
<tr>
<td>129</td>
<td>Cannot open this file.</td>
</tr>
<tr>
<td>130</td>
<td>File does not exist.</td>
</tr>
<tr>
<td>131</td>
<td>RAID type must be r1 (RAID1), r3 (RAID 3), r5 (RAID 5), or r1_0 (RAID1/0).</td>
</tr>
<tr>
<td>132</td>
<td>Multiple subcommands specified. Check syntax.</td>
</tr>
<tr>
<td>133</td>
<td>Disk for PSM must be on DPE bus 0.</td>
</tr>
<tr>
<td>134</td>
<td>Configuration does not exist.</td>
</tr>
<tr>
<td>135</td>
<td>Configuration already exists.</td>
</tr>
<tr>
<td>136</td>
<td>Size specified is too small.</td>
</tr>
<tr>
<td>137</td>
<td>Configuration does not exist. Run the navicli <code>initializearray</code> command to configure the system.</td>
</tr>
<tr>
<td>138</td>
<td>First option must be a subcommand.</td>
</tr>
<tr>
<td>139</td>
<td>Cannot create RAID group for PSM (Persistent Storage Manager).</td>
</tr>
<tr>
<td>140</td>
<td>Name or UID (unique ID) is required.</td>
</tr>
<tr>
<td>141</td>
<td>Invalid name specified.</td>
</tr>
<tr>
<td>142</td>
<td>Image UID is required.</td>
</tr>
<tr>
<td>143</td>
<td>Name and LUN are required.</td>
</tr>
<tr>
<td>144</td>
<td>Storage system UID (unique ID) and LUN UID are required.</td>
</tr>
<tr>
<td>Error value</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>145</td>
<td>Mirror not found.</td>
</tr>
<tr>
<td>146</td>
<td>Image not found.</td>
</tr>
<tr>
<td>147</td>
<td>Synchronized rate cannot be changed since input image is primary.</td>
</tr>
<tr>
<td>148</td>
<td>Cannot specify both <code>-uid</code> and <code>-filename</code>.</td>
</tr>
<tr>
<td>149</td>
<td>Invalid cache.</td>
</tr>
<tr>
<td>150</td>
<td>Invalid session.</td>
</tr>
<tr>
<td>151</td>
<td>Session does not exist.</td>
</tr>
<tr>
<td>152</td>
<td>Session is stopping.</td>
</tr>
<tr>
<td>155</td>
<td>The <code>-o</code> option requires <code>--all</code> or <code>--filename</code>.</td>
</tr>
<tr>
<td>156</td>
<td>Path to store files is required.</td>
</tr>
<tr>
<td>157</td>
<td>Cannot specify both <code>--all</code> and <code>--filename</code>.</td>
</tr>
<tr>
<td>158</td>
<td>Enter file index or <code>quit</code>.</td>
</tr>
<tr>
<td>159</td>
<td>Invalid input.</td>
</tr>
<tr>
<td>160</td>
<td>Index is out of range.</td>
</tr>
<tr>
<td>161</td>
<td>File not found.</td>
</tr>
<tr>
<td>162</td>
<td>Space not available to retrieve file.</td>
</tr>
<tr>
<td>163</td>
<td>Specified feature not supported.</td>
</tr>
<tr>
<td>164</td>
<td>Feature must be specified.</td>
</tr>
<tr>
<td>165</td>
<td>Cannot specify both <code>-lun</code> and <code>-lunuid</code>.</td>
</tr>
<tr>
<td>166</td>
<td>Invalid Storage Processor name.</td>
</tr>
<tr>
<td>167</td>
<td>PSM (Persistent Storage Manager) is not broken.</td>
</tr>
<tr>
<td>168</td>
<td>PSM (Persistent Storage Manager) is broken. Cannot list or create PSM.</td>
</tr>
<tr>
<td>169</td>
<td>LUN cannot be unbound.</td>
</tr>
<tr>
<td>170</td>
<td>Operation not supported on this type of storage system.</td>
</tr>
<tr>
<td>Error value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>171</td>
<td>Incompatible arguments. Invalid storage system serial number.</td>
</tr>
<tr>
<td>172</td>
<td>Directory not specified.</td>
</tr>
<tr>
<td>173</td>
<td>Invalid number of blocks.</td>
</tr>
<tr>
<td>174</td>
<td>Number of blocks not specified.</td>
</tr>
<tr>
<td>175</td>
<td>Reading of data not supported on this storage system.</td>
</tr>
<tr>
<td>176</td>
<td>Invalid snapshot World Wide Name (WWN).</td>
</tr>
<tr>
<td>177</td>
<td>Invalid storage system serial number.</td>
</tr>
<tr>
<td>178</td>
<td>Navicli <code>-f</code> option required to store data in file.</td>
</tr>
<tr>
<td>179</td>
<td>Invalid IP Address format.</td>
</tr>
<tr>
<td>180</td>
<td>Storage group is not shareable.</td>
</tr>
<tr>
<td>181</td>
<td>Invalid HLU number.</td>
</tr>
<tr>
<td>182</td>
<td>Invalid ALU number.</td>
</tr>
<tr>
<td>183</td>
<td>Invalid port ID.</td>
</tr>
<tr>
<td>184</td>
<td>Remote server can not be managed.</td>
</tr>
<tr>
<td>185</td>
<td>Email response test failed.</td>
</tr>
<tr>
<td>186</td>
<td>Emailpage response test failed.</td>
</tr>
<tr>
<td>187</td>
<td>Modempage response test failed.</td>
</tr>
<tr>
<td>188</td>
<td>Snmp response test failed.</td>
</tr>
<tr>
<td>189</td>
<td>Phone home response test failed.</td>
</tr>
<tr>
<td>190</td>
<td>Mandatory switch for email/emailpage.</td>
</tr>
<tr>
<td>191</td>
<td>Mandatory switch for modempage.</td>
</tr>
<tr>
<td>192</td>
<td>Mandatory switch for snmp.</td>
</tr>
<tr>
<td>193</td>
<td>Only one message or file can be specified.</td>
</tr>
<tr>
<td>194</td>
<td>Valid dial string contains only digits, parentheses, hyphen.</td>
</tr>
<tr>
<td>195</td>
<td>File does not exist or cannot open.</td>
</tr>
<tr>
<td>196</td>
<td>Specified user already exists.</td>
</tr>
<tr>
<td>Error value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>197</td>
<td>The offset switch is not supported for this storage system.</td>
</tr>
<tr>
<td>198</td>
<td>Valid COM Port number is 1,2,3, or 4.</td>
</tr>
<tr>
<td>199</td>
<td>Valid Dial Command is atd, atDp or atD.</td>
</tr>
<tr>
<td>200</td>
<td>Valid Message delay contains only &quot;,,&quot; (one or more commas).</td>
</tr>
<tr>
<td>202</td>
<td>Target LUN number is missing.</td>
</tr>
<tr>
<td>203</td>
<td>Session name is missing.</td>
</tr>
<tr>
<td>206</td>
<td>Cannot specify both -mode and -simulation.</td>
</tr>
<tr>
<td>207</td>
<td>This command is not supported on remote server.</td>
</tr>
<tr>
<td>208</td>
<td>Switch -pathname must be specified.</td>
</tr>
<tr>
<td>209</td>
<td>Get local server attributes failed.</td>
</tr>
<tr>
<td>210</td>
<td>This version of FLARE software does not support Hi5 RAID type.</td>
</tr>
<tr>
<td>215</td>
<td>The HLU (host LUN) number you specified cannot be found.</td>
</tr>
<tr>
<td>216</td>
<td>This command must be issued from the SP to which the LUN will trespass.</td>
</tr>
<tr>
<td>217</td>
<td>Invalid bus or enclosure number.</td>
</tr>
<tr>
<td>218</td>
<td>Invalid WWN Seed.</td>
</tr>
<tr>
<td>219</td>
<td>Invalid EMC Part Number.</td>
</tr>
<tr>
<td>220</td>
<td>This RAID group has Maximum number of LUNs already.</td>
</tr>
<tr>
<td>221</td>
<td>Not enough cache LUNs in the specified SP's cache.</td>
</tr>
<tr>
<td>222</td>
<td>This LUN cannot be added to the storage group since it is participating in a remote mirror.</td>
</tr>
<tr>
<td>223</td>
<td>Allocate log need specify LUN with valid owner.</td>
</tr>
<tr>
<td>224</td>
<td>This request has been issued through the SP that is not the current owner of the targeted LUN.</td>
</tr>
<tr>
<td>234</td>
<td>The LUN is in a storage group.</td>
</tr>
<tr>
<td>276</td>
<td>Multiple mirrors with the same name have been found. Please use UID for a specific mirror.</td>
</tr>
<tr>
<td>337</td>
<td>Minimum LUN size for MirrorView Write Intent Log is 128 MB.</td>
</tr>
</tbody>
</table>
A
account, user 2-8, 2-15
application integration 1-16
arguments, spaces in 2-14, 2-19, B-4

B
bidirectional mirroring 1-15

C
case (of characters in CLI commands) 2-8, 2-15
CLI
  environment variables B-4
    RaidAgentDevice B-4
    RaidAgentHost B-4
    RaidAgentTimeout B-4
Java
  about 2-15
  basic commands 2-10, 2-17
  requirements 2-8, 2-15
  spaces in arguments 2-14, 2-19
  using security 2-8, 2-15
MirrorView/A subcommands, Chapter 2
command line format, spaces in 2-14, 2-19, B-4
consistency group
  adding a mirror to 3-7
  changing properties 3-11
  creating 3-2
  destroying 3-5
  fracturing images in 3-16
  listing information about 3-20
  promoting images in 3-18
  removing mirror from 3-9
  updating 3-14

D
diagnostics, navcli command B-4

E
environment variables, CLI B-4
error
codes for MirrorView/A subcommands C-2

F
failures, how MirrorView handles A-2
file, security 2-8, 2-15

I
integration with EMC SAN Copy software 1-15
integration with EMC SnapView Snapshot software 1-15

M
mirror addimage subcommand 2-22, 3-7
mirror addtogroup subcommand 3-7
mirror change subcommand 2-26
mirror changegroup subcommand 3-11
mirror changimage subcommand 2-28
mirror create subcommand 2-31
mirror creategroup subcommand 3-2
mirror destroy subcommand 2-33
mirror destroygroup subcommand 3-5
mirror disablepath subcommand 2-35
mirror enablepath subcommand 2-37
mirror fracturegroup subcommand 3-16
mirror fractureimage subcommand 2-39
mirror info subcommand  2-41
mirror list subcommand  2-43
mirror listgroups subcommand  3-20
mirror promotegroup subcommand  3-18
mirror promoteimage subcommand  2-46, 3-2, 3-18
mirror removefromgroup subcommand  3-9
mirror removeimage subcommand  2-49
mirror syncgroup subcommand  3-14
mirror syncimage subcommand  2-51, 2-53
MirrorView
  handling failures A-2
MirrorView/A
  about Chapter 2
  configuration guidelines 1-4
  connection requirements 1-7
  operations overview 2-3
  overview 1-2
  prerequisites 1-3
  terminology 1-11
MirrorView/A error codes C-2
MirrorView/A subcommands, Chapter 2

R
RaidAgent environment variables B-4
recovery
  by promoting a secondary image A-6
remote mirror
  adding image 2-22, 3-7
  changing 2-26
  changing image 2-28
  creating 2-31
  destroying 2-33
  fracturing image 2-39
  info command 2-41
  listing mirror information 2-43
  path, disabling 2-35
  path, enabling 2-37
  promoting image 2-46, 3-2, 3-18
  removing image 2-49
  synchronizing image 2-51, 2-53
remote mirroring, promotion 1-6
replication over long distances 1-16
requirements, Java CLI 2-8, 2-15

S
secondary image
  failure A-8
  promotion 1-6, A-10
security, Navisphere 2-8, 2-15
session, MirrorView/A, see MirrorView/A
set command B-4
spaces in arguments B-4
subcommands
  mirror addimage 2-22, 3-7
  mirror addtogroup 3-7
  mirror change 2-26
  mirror changegroup 3-11
  mirror changeimage 2-28
  mirror create 2-31
  mirror creategroup 3-2
  mirror destroy 2-33
  mirror destroygroup 3-5
  mirror disablepath 2-35
  mirror enablepath 2-37
  mirror fracturegroup 3-16
  mirror fractureimage 2-39
  mirror info 2-41
  mirror list 2-43

orphan image 2-33

password 2-8, 2-15
path
  remote mirror, disabling 2-35
  remote mirror, enabling 2-37
primary image 1-2
  failure A-2
privileged user B-2
promoting
  a secondary image to a primary A-6
  promotion of secondary image 1-6, A-10

Navisphere security 2-8, 2-15
<table>
<thead>
<tr>
<th>Command</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>mirror listgroups</td>
<td>3-20</td>
</tr>
<tr>
<td>mirror promotegroup</td>
<td>3-18</td>
</tr>
<tr>
<td>mirror promoteimage</td>
<td>2-46, 3-2, 3-18</td>
</tr>
<tr>
<td>mirror removefromgroup</td>
<td>3-9</td>
</tr>
<tr>
<td>mirror removeimage</td>
<td>2-49</td>
</tr>
<tr>
<td>mirror syncgroup</td>
<td>3-14</td>
</tr>
<tr>
<td>mirror syncimage</td>
<td>2-51, 2-53</td>
</tr>
</tbody>
</table>

**U**

- user, security: 2-8, 2-15
- username: 2-8, 2-15