EMC® VNX® Series

Release 8.1

Using VNX® SnapSure™

P/N 300-015-121 Rev 01

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

Published August 2013

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

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

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

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

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

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

Corporate Headquarters: Hopkinton, MA 01748-9103

Contents

Contents

Preface

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

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

Special notice conventions

EMC uses the following conventions for special notices:

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

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

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

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

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

Where to get help

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

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

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

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

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

Your comments

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

Please send your opinion of this document to:

techpubcomments@EMC.com

1 Introduction

EMC SnapSure is an EMC VNX software feature that enables you to create and manage checkpoints, which are point-in-time, logical images of a Production File System (PFS). Chapter 2 provides more details.

This document is part of the VNX documentation set and is intended for use by system administrators responsible for maintaining business continuance.

Topics included are:

System requirements

Table 1 on page 10 describes the EMC® VNX® software, hardware, network, system resources, and system environment.

Table 1. System requirements

Software	EMC VNX version 8.1 or later.
Hardware	Refer to the EMC E-Lab™ Interoperability Navigator on EMC Online Support for specific hardware requirements.
Network	No specific network requirements.
System resources	Sufficient Data Mover blockmap memory, system space, and disk storage must be available to support EMC SnapSure™ operations. Scope resource requirements for SavVol on page 21 provides more details.
System environment	While using SnapSure, ensure that: ◆ All parts of a PFS, including all parts of its checkpoint such as the SavVol, are in the same VNX cabinet. ◆ All volumes, including the PFS and the SavVol, are accessible to the same Data Mover. ◆ The PFS is mounted on a single Data Mover to enable SnapSure to access data and create checkpoints. ◆ All checkpoints are mounted to conserve the SavVol space usage and enable checkpoint application access to the data. ◆ The checkpoints are mounted on the same Data Mover as the PFS.

Restrictions and limitations

The restrictions while using SnapSure on EMC VNX are:

- A checkpoint is not intended to be a mirror, disaster recovery, or high-availability tool. It is partially derived from realtime PFS data. A checkpoint might become inaccessible or unreadable if the associated PFS is inaccessible. Only a PFS and its checkpoints saved to a tape or an alternate storage location can be used for disaster recovery.
- For each Data Mover, the total size of all file systems, the size of all SavVols used by SnapSure, and the size of all SavVols used by EMC VNX Replicator must be less than the total supported capacity of the Data Mover. VNX Release Notes, available on EMC Online Support, include a list of Data Mover capacities.

- All parts of a PFS, including all parts of its associated checkpoint, such as the SavVol, must reside on the same storage array.
- Writeable checkpoints do not support quota operations.

Cautions

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

- Do not perform data migration tasks by using VNX File System Migration while SnapSure is active. Migration activity produces significant changes in the data environment. Complete the migration tasks before using SnapSure to avoid consuming SnapSure resources for capturing information unnecessary to checkpoint.
- Do not create or refresh a checkpoint during a Unicode conversion process. This can cause the create or refresh operation to fail. If the operation fails, delete the checkpoint associated with the command failure. Retry creating or refreshing the checkpoint after the conversion process is completed.
- Do not perform a full restore of a PFS from a checkpoint until sufficient system resources are available for SnapSure to complete the operation. Without resources for the restore operation, data in all PFS checkpoints might be lost.
- Be cautious when restoring a file system from one of its checkpoints. If you create a checkpoint of any file system and write a file (for example, file1) into the file system, and later restore the file system by using the checkpoint, then the file (file1) does not remain. This is because the file did not exist when the checkpoint was created. VNX for File provides a warning when a restore is attempted.
- Be cautious with applications that write large amounts of data to writeable checkpoints. Writing large amounts of data causes the SavVol to extend. After it is extended, SavVol cannot be shrunk without loss of data.
- Be cautious with applications that write large amounts of data to writeable checkpoints to prevent automatic inactivation of read-only checkpoints.
- Creating new file systems and extending or mounting file systems might fail if a SnapSure checkpoint upgrade is in progress.
- Baseline and writeable checkpoints cannot be refreshed. When creating a schedule, ensure that the baseline and writeable checkpoints are not included. Otherwise, the schedule fails.

User interface choices

VNX for File offers flexibility in managing networked storage that is based on your support environment and interface preferences. This document describes how to configure and manage SnapSure by using the command line interface (CLI). You can also perform SnapSure tasks, including checkpoint scheduling, by using the EMC Unisphere® software.

EMC Unisphere online help provides more information about user interface choices.

Related information

These VNX documents provide information beyond the scope of this document:

- *Configuring Events and Notifications on VNX for File*
- *Configuring NDMP Backups on VNX*
- *Configuring NDMP Backups to Disk on VNX*
- *VNX Glossary*
- *EMC VNX Command Line Interface Reference for File*
- VNX for File man pages
- *Celerra Network Server Error Messages Guide*
- *Parameters Guide for VNX*
- *VNX 1.0 Release Notes*
- *VNX System Operations*
- *Security Configuration Guide for File*
- *Problem Resolution Roadmap for VNX*
- *Using VNX FileMover*
- *Using VNX Replicator*
- *Using SRDF/S with VNX for Disaster Recovery*

EMC VNX documentation on EMC Online Support

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

VNX wizards

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

2 Concepts

SnapSure enables you to create point-in-time logical images of a PFS.

SnapSure uses a "copy on first modify" principle. A PFS consists of blocks. When a block within the PFS is modified, a copy containing the block's original contents is saved to a separate volume called the SavVol. Subsequent changes made to the same block in the PFS are not copied into the SavVol. The original blocks from the PFS in the SavVol and the unchanged PFS blocks remaining in the PFS are read by SnapSure according to a bitmap and blockmap data-tracking structure. These blocks combine to provide a complete point-in-time image called a checkpoint. Topics included are:

Checkpoint types

A checkpoint reflects the state of a PFS at the point in time the checkpoint is created. SnapSure supports two types of checkpoints:

- Read-only checkpoint — Read-only file system created from a PFS.
- Writeable checkpoint — Read/write file system created from a read-only checkpoint.

SnapSure can maintain a maximum of 96 read-only checkpoints and 16 writeable checkpoints per PFS while allowing PFS applications continued access to realtime data.

Note: Each writeable checkpoint is associated with a read-only checkpoint, referred to as the baseline checkpoint. Each baseline checkpoint can have only one associated writeable checkpoint.

Read-only and writeable checkpoints can serve as a direct data source for applications that require point-in-time data. Such applications include simulation testing, data warehouse population, and automated backup engines performing backup to tape or disk. You can also use a read-only or a writeable checkpoint to restore a PFS or part of a file system, such as a file or directory, to the state in which it existed when the checkpoint was created. Writeable checkpoints can also serve as a source for applications that require incremental, temporary changes. For example, a writeable checkpoint of a database hosted on a VNX for File can be subjected to data processing and warehousing without committing the changes to the production database. Another example is that of a software patch applied on a database. An administrator first applies the patch on the database's writeable checkpoint and tests it before applying the patch on the production database.

Checkpoint operations

SnapSure supports the following checkpoint operations:

- Create — Creates a read-only checkpoint from a PFS, or a writeable checkpoint from a baseline read-only checkpoint.

  Note: You can create 96 read-only checkpoints on a PFS, and one writeable checkpoint per read-only checkpoint. A maximum of 16 writeable checkpoints can exist concurrently on a PFS.

- Refresh — Recycles the SavVol space by deleting the data in the checkpoint of a PFS and creating a new checkpoint by using the same checkpoint filename, file system identification number (ID), and mount state. Baseline and writeable checkpoints cannot be refreshed.
- Restore — Restores a PFS to a point in time by using a read-only or writeable checkpoint.
- Mount — Mounts a read-only checkpoint as a read-only file system, or a writeable checkpoint as a read-only or read/write file system. Both read-only and writeable checkpoints are mounted automatically upon creation.

Note: The number of writeable checkpoints per PFS does not count towards the total number of read-only checkpoints per PFS. However, mounted writeable checkpoint file systems count towards the 2048 maximum number of file systems per Data Mover and the 4096 maximum number of file systems per VNX cabinet. Also, writeable checkpoints might increase the size of the SavVol as changed blocks are written to it directly.

- Delete — Deletes a read-only or writeable checkpoint. A read-only checkpoint that is a baseline checkpoint cannot be deleted unless the associated writeable checkpoint is deleted.
- Export — Makes a read-only or writeable checkpoint available to network file system (NFS) users on a network.
- Share — Makes a read-only or writeable checkpoint available to Common Internet File System (CIFS) users on a network.
- Schedule — Automates read-only checkpoint creation and refresh operations. Writeable checkpoints cannot be scheduled.

Checkpoint views

SnapSure provides two views of a PFS during a checkpoint window. The views depend on the file system applications access:

- PFS view — PFS applications view realtime PFS data.
- Point-in-time view — Checkpoint applications view point-in-time PFS data.

PFS applications read and write on PFS data blocks. Checkpoint applications read a point-in-time state of the original data blocks. Figure 1 on page 15 shows the two views during a window. In this model, PFS and checkpoint data are kept separate.

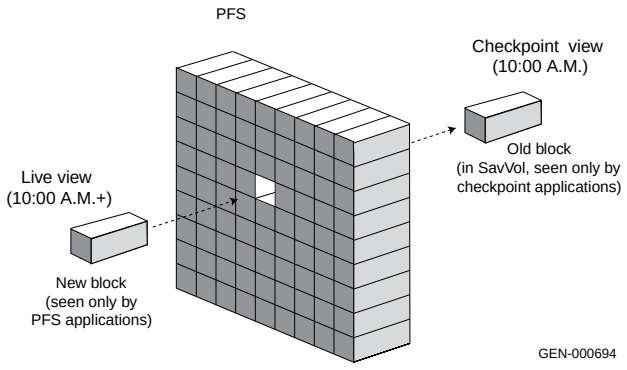


Figure 1. Two views of a PFS during a checkpoint window

Copy PFS data blocks into SavVol

The first time a change instruction from a PFS application is detected, SnapSure copies the original PFS block into the SavVol (unless the block is unused by a checkpoint), and then allows the instruction to execute on the PFS block. Figure 2 on page 16 shows how PFS data blocks are copied into SavVol.

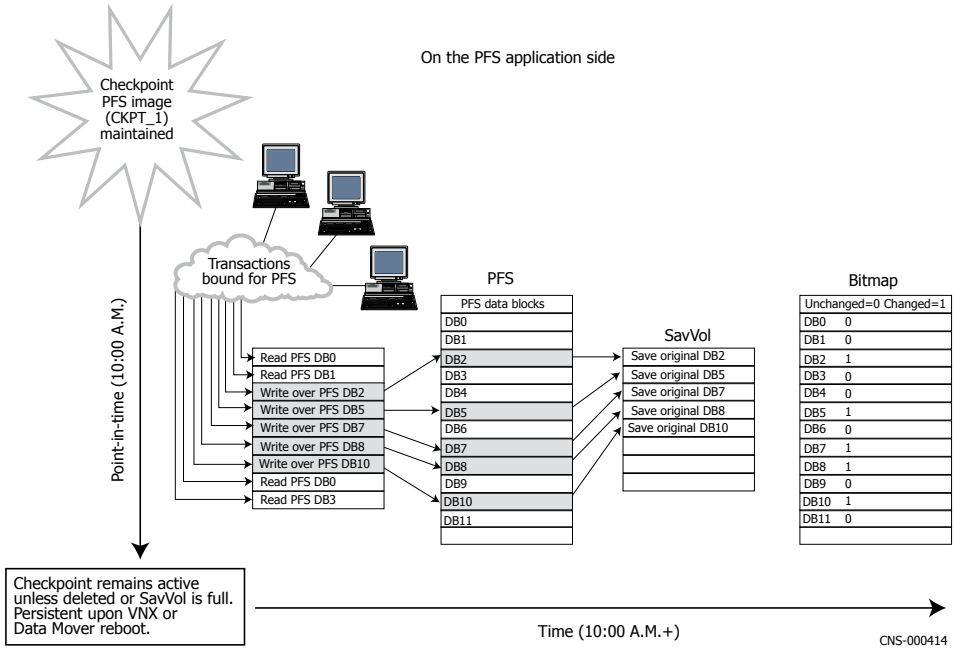


Figure 2. Using the SavVol to save PFS blocks before the blocks change

In this model, the original PFS data blocks are copied into the SavVol before being overwritten by the initial write requests from the PFS application. The system then performs bitmap tracking to show the modified data blocks. Note that only the latest checkpoint has a bitmap, but all checkpoints, including the newest, have their own blockmap.

Provide the checkpoint view for the client

When NFS or CIFS clients access the latest checkpoint, SnapSure queries the checkpoint bitmap for the existence of the needed point-in-time block. If the bitmap argument is ON, the data is read from the SavVol by using the checkpoint blockmap to determine the location (SavVol block number) to read. If the bitmap argument is OFF, the data is read from the PFS.

When NFS or CIFS clients access a checkpoint that is not the latest, SnapSure directly queries the checkpoint's blockmap for the SavVol block number to read. If the block number is in

the blockmap, the data is read from the SavVol space for the checkpoint. If the block number is not in the blockmap, SnapSure queries the next newer checkpoint to find the block number. This method is repeated until the PFS is reached.

Bitmap and blockmaps play similar roles. A bitmap is used for the latest checkpoint because processing data with it is faster than with the blockmap, and the latest checkpoint is typically accessed most frequently.

shows how data is read with a bitmap and blockmap. If the bitmap indicates that the PFS data block has changed since the creation of the latest checkpoint, as shown with a one (1) in this example, the system consults the checkpoint's blockmap to determine the SavVol block that contains the original PFS block. The read is then directed to that SavVol block.

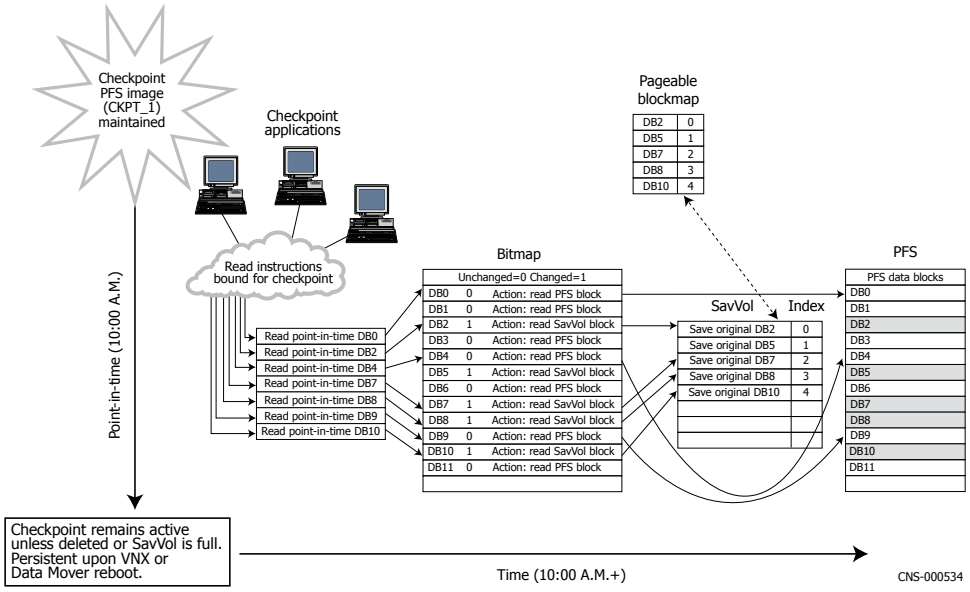


Figure 3. Constructing the checkpoint image by using bitmap and blockmap

If the PFS data block has not changed since the creation of the checkpoint, as indicated with a zero (0) in the bitmap, the application reads the data block directly from the PFS. The bitmap is present only for the latest checkpoint. In its absence, the blockmap for the checkpoint is consulted immediately to determine if a mapping is present.

Consider a PFS with blocks a, b, c, d, e, f, g, and h. When the checkpoint is created, the blocks have values a0, b0, c0, d0, e0, f0, g0, and h0. Thereafter, PFS applications modify blocks a and b, writing the values a1 and b1 into them. At this point, the following contents are in the volumes.

PFS	a1, b1, c0, d0, e0, f0, g0, h0
SavVol	a0, b0
Checkpoint view of PFS	a0, b0, c0, d0, e0, f0, g0, h0

As shown in this example, when the checkpoint is read, you get the complete point-in-time picture of the PFS.

SnapSure adds efficiency to this method by tracking blocks used by the PFS when a checkpoint is created. It does not copy the original PFS data when changes are made to blocks not used by the PFS (blocks that were free) at the time of checkpoint creation.

For example, consider a PFS with blocks a, b, c, d, e, f, g, and h, but among these only a, b, c, and e are used by the PFS at the time of checkpoint creation. Thereafter, if block d, which was free at the time of checkpoint creation, is modified by the PFS application, its original data is not copied into the SavVol. However, if block c is modified, its original data is copied into the SavVol.

Checkpoints are automatically mounted on creation, and should remain so, to ensure SnapSure efficiency.

Writeable checkpoint view

A writeable checkpoint is a read/write file system created from a read-only checkpoint called a baseline checkpoint. The read/write file system provides the writeable checkpoint view that supports incremental changes on the baseline checkpoint. When NFS or CIFS clients write to a writeable checkpoint, SnapSure saves the incremental changes in the SavVol. When reading a writeable checkpoint, SnapSure queries the checkpoint's blockmap and the corresponding blocks in the SavVol that saved the incremental changes to provide a complete view of the PFS. Disk space for writeable checkpoints on page 23 provides more details on writeable checkpoints.

Manage multiple checkpoints

You can create a maximum of 96 read-only and 16 writeable checkpoints on the same PFS, each representing the state of the PFS at a given point in time, as shown in Figure 4 on page 18.

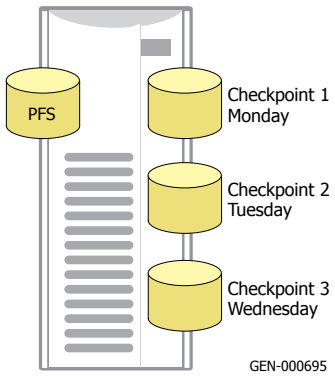


Figure 4. SnapSure supports multiple checkpoints of the same PFS

Multiple checkpoints share the same SavVol, but are logically separate point-in-time file systems. When multiple checkpoints are active, their blockmaps are linked in chronological order. For any checkpoint, blockmap entries needed by the system, but not resident in main memory, are paged in from the SavVol. The entries stay in the main memory until system memory consumption requires them to be purged. If a PFS data block remains unchanged since the checkpoints were created, the system directs the read request to the PFS data block where the original point-in-time data resides.

Planning considerations

With SnapSure, you can perform administrative tasks, such as creating, listing, refreshing, or deleting checkpoints, and employing these checkpoints for restoring a PFS or copying to a backup device. Clients can use checkpoint data to independently restore files and directories or for business modeling.

Consider these planning and management considerations when:

- Upgrade SnapSure from previous versions on page 19
- Perform checkpoint operations on page 19
- Work with PFS on page 20
- Scope resource requirements for SavVol on page 21
- Use SnapSure options for conserving space on page 26
- Restore from a checkpoint on page 27
- Access checkpoints online on page 28
- Use automated checkpoint scheduling on page 29
- Work with NDMP backup-created checkpoints on page 31
- Work with other VNX features on page 32

Upgrade SnapSure from previous versions

You can upgrade VNX to version 7.0 from version 5.4 or later. To upgrade from a version earlier than 5.4, you must first upgrade to version 5.4 or 5.5, then to version 6.0, and then upgrade to version 7.0. *VNX 1.0 Release Notes* on the EMC Online Support website provide more details.

Perform checkpoint operations

SnapSure checkpoint operations affect the system performance as follows:

- Creating a checkpoint requires the PFS to be paused. Therefore, PFS write activity suspends, but read activity continues while the system creates the checkpoint. The pause

time depends on the amount of data in the cache, but it is typically one second or less. EMC recommends a 15-minute minimum interval between the creation or refresh of checkpoints of the same PFS.

- Restoring a PFS from a checkpoint requires the PFS to be frozen. Therefore, all PFS activities are suspended while the system restores the PFS from the selected checkpoint. The freeze time depends on the amount of data in the cache, but it is typically one second or less.

  Note: When read activity is suspended during a freeze, connections to CIFS users are broken. However, this is not the case when write activity is suspended.

- Refreshing a checkpoint requires the checkpoint to be frozen. Therefore, checkpoint read activity suspends while the system refreshes the checkpoint. Clients attempting to access the checkpoint during a refresh process experience the following:

  - NFS — The system continuously tries to connect indefinitely. When the system thaws, the file system automatically remounts.
  - CIFS — Depending on the application running on Windows, or if the system freezes for more than 45 seconds, the Windows application might drop the link. The share might need to be remounted and remapped.

- Deleting a checkpoint requires the PFS to be paused. Therefore, PFS write activity suspends momentarily, but read activity continues while the system deletes the checkpoint.

If a checkpoint becomes inactive for any reason, read/write activity on the PFS continues uninterrupted.

Work with PFS

When working with PFS, consider:

- Checkpoint operations and PFS state on page 20
- Checkpoint persistence on page 21

Checkpoint operations and PFS state

When checkpoint operations, such as create, refresh, restore, mount, recover, or delete, are evoked on a PFS, SnapSure queries the underlying PFS to verify its status. If a corruption is detected in the PFS, checkpoint operations are affected in the following ways:

- Checkpoint create and refresh operations are rejected. A LOG_ERROR message “PFS on volume <‘volume ID’> is marked as corrupted, operation rejected." is displayed.
- Checkpoint restore operation proceeds. However, a LOG_INFO message “PFS on volume <‘volume ID’> is marked corrupted when ckpt restore is attempted." is displayed.

- Checkpoint mount operation is rejected. A LOG_ERROR message “PFS <‘file system ID’> is marked as corrupted, ckpt mount is rejected." is displayed.

  Note: You can force mount a checkpoint by setting the SVFS.forceCkptMount parameter to 1. Setting SVFS.forceCkptMount to 1 allows the checkpoint mount operation to proceed, but with the LOG_NOTICE message: “PFS <‘file system ID’> is marked as corrupted when ckpt mount is forced to proceed."

- Checkpoint recover and delete operations proceed uninterrupted as these operations are independent of the PFS state.

Checkpoint persistence

Checkpoints persist in the following manner:

- If the Data Mover on which a PFS is mounted is rebooted or fails over to another Data Mover, the associated checkpoints remain active. The checkpoints are recovered before the PFS is available to allow any cumulative PFS modifications to be captured by the checkpoints.

  Note: When a checkpoint accumulates a large amount of data due to a heavy volume of PFS modifications, for example, the checkpoint approaches 500 GB, it should be refreshed regularly. When large checkpoints are up-to-date, system performance is optimized during a reboot or failover to another Data Mover.

- The PFS can be temporarily unmounted if its checkpoints are mounted, but it cannot be permanently unmounted until all checkpoints are unmounted.
- When a PFS with a checkpoint attached to it is unmounted and then mounted on another Data Mover, the checkpoints are recovered, but must be remounted.
- In versions 5.2 and later, all checkpoints, by default, are mounted automatically upon creation. However, checkpoints created in other versions of SnapSure are not mounted automatically, and are also not mounted automatically upon upgrade to version 5.2 or later. Mount these checkpoints on the same Data Movers as the PFS after upgrade to ensure SnapSure efficiency.

Scope resource requirements for SavVol

Important resource requirement considerations include:

- Disk space for creating and extending SavVols on page 22
- SavVol checkpoint storage allocation on page 23
- SavVol size customization on page 23
- Disk space for writeable checkpoints on page 23
- System space for disks that can be used for SavVols on page 24

- System space and checkpoint inactivation on page 24

Disk space for creating and extending SavVols

SnapSure requires a SavVol, or checkpoint-specific volume to hold checkpoint data. The SavVol is created when the first checkpoint is created on a PFS. For subsequent checkpoints on the same PFS, SnapSure uses the same SavVol, but logically separates the point-in-time data by using unique checkpoint names. Note that a SavVol is not an EMC Symmetrix® business continuance volume (BCV) as required by the EMC TimeFinder®/FS feature.

When you create the first checkpoint, you can opt to let SnapSure automatically create and manage a SavVol by using its default settings, or you can create and manage the SavVol activity manually by using the SnapSure volume and high water mark (HWM) options.

You can create the first checkpoint by using the SnapSure default values. For example, if you use the CLI fs_ckpt <fsname> -Create command and specify no volume_name, size= or %full= option, SnapSure automatically:

- Creates a SavVol by using the following criteria:
  - If PFS ≥ 20 GB, then SavVol = 20 GB.
  - If (PFS < 20 GB) and (PFS > 64 MB), then SavVol = PFS size.
  - If PFS ≤ 64 MB, then SavVol = 64 MB, which is the minimum SavVol size.
- Monitors the SavVol, and if a checkpoint reaches a HWM of 90 percent (default), extends it to keep the checkpoint active or readable. The SavVol extends by 20 GB each time until the SavVol usage is below the HWM. If the SavVol was created manually (not using AVM), then it extends by 10 percent of the PFS size each time. The maximum SavVol size also affects the SavVol extension size. If the maximum SavVol size is set by the user, and extending another 20 GB would make the SavVol exceed the maximum size, then the extension size is adjusted to the maximum size minus the current size.

Note: By default, EMC Automatic Volume Management (AVM) algorithms determine the selection of disks used for a SavVol. AVM tries to match the storage pool for the SavVol with that of the PFS whenever possible. However, if you do not want to use the AVM selection method, SnapSure provides the option of specifying any available pool for the SavVol when you create the first PFS checkpoint. The SavVol can be autoextended as long as space is available on the same disk type. For example, if the SavVol is built on an VNX for Block standard (CLSTD) disk, then autoextend can occur only if space is available on another CLSTD disk. The autoextend feature cannot use a VNX for Block ATA (CLATA) disk for the autoextend. Use the nas_disk -list command to display the disk types.

You can also specify a limit to which SavVol can automatically extend by using the fs_ckpt -modify command. An event message is sent to the system log each time an

automatic SavVol extension succeeds or fails. Set the SavVol extension limit on page 43 provides details on setting SavVol extension limit.

SavVol checkpoint storage allocation

When creating a checkpoint, SnapSure pre-allocates SavVol space in chunks. This space is used for both management purposes and data storage. SnapSure allocates either 8 MB or 128 MB chunks depending on the size of the primary file system:

- If PFS > 1 GB, then SavVol space allocated = 128 MB.
- If PFS < 1 GB, then SavVol space allocated = 8 MB.

If the checkpoint is a writable checkpoint, an additional chunk is reserved during checkpoint creation. After the chunk size has been determined, it cannot be changed during the lifetime of the SavVol. After a new checkpoint is created, one chunk of storage space is allocated from the SavVol. As the checkpoint requires more space, additional chunks will be allocated on demand.

SavVol size customization

A large and busy PFS requires a bigger SavVol than a smaller, less active PFS. You can customize the SavVol size proportionally to the PFS size and the activity level the checkpoints need to capture. In general, start by specifying a SavVol size 10 percent of the PFS and, as required, extend or shrink the SavVol size.

You can extend the SavVol size by using the nas_fs -xtend command. Extend the SavVol size on page 44 provides details on extending SavVol size.

You can shrink the SavVol size as follows:

1. Back up the existing checkpoint data to a tape or disk.
2. Delete all checkpoints. SnapSure automatically deletes associated SavVol.
3. Create a new checkpoint of the PFS, specifying a custom-size SavVol.

Note: If you use only the size option, for example, size = 100 MB, when creating the first checkpoint of a PFS, VNX for File first selects the same pool and storage system as the PFS when creating the SavVol. If the PFS spans multiple storage systems, the system picks the same set of storage systems to find space for the SavVol after it ensures that there is at least the specified 100 MB amount of space in the pool on the storage system. If not, the system extends the storage pool by adding unused disk volumes, and then creates the SavVol. If the pool cannot be extended, the QoS method of disk selection is used.

Disk space for writeable checkpoints

Writeable checkpoints share the same SavVol space as read-only checkpoints. When NFS or CIFS clients write to a writeable checkpoint, SnapSure saves the changes to SavVol. If SavVol becomes full, space is made available through automatic or manual SavVol extension. System space and checkpoint inactivation on page 24 provides more details on SavVol space utilization.

NOTICE Handle applications that write large amounts of data to a writeable checkpoint with caution. After it is extended, SavVol cannot be shrunk without loss of data.

System space for disks that can be used for SavVols

The system allows SavVols to be created and extended until the sum of the space consumed by all SavVols on the system exceeds the default 20 percent of the total space available on the system. This calculation includes SavVols used by SnapSure and other applications. Before using SnapSure, ensure that the system has enough space for the SavVols for each PFS of which you plan to create checkpoints.

If the 20 percent value is reached, the following message appears in the command output, and also in the /nas/log/cmd_log.err and /nas/log/nas_log.al.err files:

```
Error 2238: Disk space: quota exceeded
Feb 1 15:20:07 2008 NASDB:6:101 CKPT volume allocation quota exceeded
```

To gauge system space manually, use the nas_fs -size command to calculate the SavVol size for each PFS with a checkpoint as shown in this example:

```
$ nas_fs -size msf001_ckpt1
volume: total = 200 avail = 119 used = 81 ( 40% ) (sizes in MB)
ckptfs: total = 196 avail = 196 used = 1 ( 0% ) (sizes in MB) (
blockcount = 409600 ) ckpt_usage_on_savevol: 8MB ( 4% )
```

Note: The ckpt_usage_on_savevol value is rounded up to the nearest integer.

Additional checkpoints on a PFS need not be calculated because checkpoints share the same SavVol per PFS. Then compare the sum of all SavVols to the entire volume size by totaling the size of all system disks. To find the system disk size, use the nas_disk -list command as shown in this example:

```
$ nas_disk -list
id   inuse  sizeMB    storageID-devID   type  name          servers
1      y      11263  APM00043200225-0000 CLSTD root_disk     1,2
2      y      11263  APM00043200225-0001 CLSTD root_ldisk    1,2
3      y       2047  APM00043200225-0002 CLSTD d3            1,2
```

System space and checkpoint inactivation

If the system space quota for SavVols exceeds and autoextension fails, SnapSure attempts to keep checkpoints active by using other methods. First, SnapSure uses the remaining SavVol space for the checkpoint. For example, SnapSure uses the remaining 10 percent of the space when an HWM of 90 percent is reached. Next, SnapSure starts inactivating older checkpoints. Inactivating older checkpoints involve unmounting the checkpoint file systems and marking them as deleted. Checkpoint inactivation can be triggered either by a client write operation on the PFS or a writeable checkpoint, or by a checkpoint restore operation.

If the inactivation is triggered by a write operation, SnapSure inactivates any existing writeable checkpoint of the oldest read-only checkpoint. If there is no writeable checkpoint, SnapSure inactivates the oldest read-only checkpoint.

If the inactivation is triggered by a restore operation, checkpoint inactivation depends on whether the checkpoint is older or newer than the restore source checkpoint. For checkpoints that are older than the restore source checkpoint, SnapSure inactivates any existing writeable checkpoint of the oldest read-only checkpoint followed by the oldest read-only checkpoint. If further space is required, SnapSure inactivates checkpoints newer than the restore source checkpoint. For newer checkpoints, SnapSure inactivates writeable checkpoints, one at a time, starting with the oldest. Then, SnapSure inactivates the restore backup checkpoint, followed by the read-only checkpoints. Note that the restore operation completes successfully even when the restore source checkpoint is inactivated.

NOTICE Handle applications that write large amounts of data to a writeable checkpoint with caution to prevent automatic inactivation of read-only checkpoints.

If SnapSure uses up all possible space it can find to keep a checkpoint active, all PFS checkpoints, including the newest checkpoint and the checkpoint from which the PFS is being restored, if applicable, become inactive and the data is lost. To make space available, delete unwanted checkpoints or change the system space parameter. Change the percentage of system space allotted to SavVol on page 45 provides details on changing the percentage of system space allotted to SavVols.

Use SnapSure options for conserving space

To conserve checkpoint space:

- Specify an HWM of zero percent on page 26
- Use the SnapSure refresh feature on page 26
- Delete unwanted checkpoints on page 27

Specify an HWM of zero percent

When you create a checkpoint, specify an HWM of zero percent by using the %full option. If you set the HWM to zero percent, SnapSure does not extend the SavVol when a checkpoint reaches full capacity. Instead, it deletes the data in the oldest checkpoint and recycles the space to keep the most recent checkpoint active. SnapSure repeats this behavior each time a checkpoint needs space. If you use this setting and if a specific checkpoint is important, periodically verify the used SavVol space by using the fs_ckpt <fsname> -list command, and before the SavVol becomes full, perform one of the following:

- Copy the checkpoint to tape.
- Read the checkpoint with checkpoint applications.
- Extend the SavVol to keep the checkpoint active.

Note: The maximum user-specifiable HWM value is 99. Using an HWM of 99 percent does not prevent SavVol autoextension. Only a zero percent HWM prevents the SavVol from autoextending. Using 99 percent allows the checkpoint to use practically all of its SavVol space before autoextending and does not provide notification to allow you to ensure resource availability, if necessary.

In summary, if you plan to use the zero percent HWM option at creation time, auditing and extending the SavVol are important checkpoint management tasks to consider.

Use the SnapSure refresh feature

Use the SnapSure refresh feature anytime after creating the checkpoint. This conserves SavVol space by recycling used space.

You can recycle SavVol space by using the SnapSure refresh feature. Rather than use new SavVol space when creating a new checkpoint of the PFS, use the refresh feature anytime after you create one or more checkpoints. SnapSure deletes the data in the checkpoint that you specify and creates a new one. You can refresh any active checkpoint of a PFS, and in any order. If the checkpoint contains important data, back it up or use it before you refresh it. The refresh operation is irreversible. When you refresh a checkpoint, SnapSure maintains the file system name, ID, and mount state of the checkpoint for the new one. The PFS must remain mounted during a refresh. With Unisphere, you can schedule checkpoint refreshes to occur regularly.

Note: When a checkpoint accumulates a large amount of data due to a heavy volume of PFS modifications, for example, the checkpoint approaches 500 GB, it should be refreshed regularly. When large checkpoints are up-to-date, system performance is optimized during a reboot or failover to another Data Mover.

Delete unwanted checkpoints

Delete unwanted checkpoints to conserve SavVol space. Deleting the oldest checkpoint typically frees more blockmap space than does deleting the newer checkpoints. Deleting and refreshing checkpoints frees up the same amount of SavVol space.

Checkpoints must be unmounted to be deleted. Unisphere unmounts the checkpoints as part of the deletion process. You cannot delete a checkpoint that is in use, or a pending scheduled checkpoint that is not yet created.

Restore from a checkpoint

From an active checkpoint, you can restore an online PFS back to the point in time in which it existed when the checkpoint was created.

Before restoring, SnapSure creates a new checkpoint of the PFS that enables you to undo the restore, if required. SnapSure uses a default 75 percent HWM for the SavVol when creating the new checkpoint. Before you attempt a restore, ensure that SnapSure has enough SavVol space available to create the new checkpoint and to keep the checkpoint you are restoring active during the operation. Otherwise, all data in the PFS checkpoints might be lost, including the new checkpoint.

SnapSure rejects any HWM less than 10 percent or greater than 75 percent by automatically overriding it with 75 percent for the SavVol when restoring. The PFS must remain mounted during a restore process. However, during a restore operation, if the HWM reaches 75 percent, VNX for File remounts the PFS as read-only until the extension is done, at which point the PFS is remounted as read/write.

Use caution when restoring a file system from one of its checkpoints. If you create a checkpoint of any file system and write a file (for example, file1) into the file system, and later restore the file system by using the checkpoint, then the file (file1) does not remain. This is because the file did not exist when the checkpoint was created. VNX for File provides a warning when a restore is attempted.

The restore operation might fail the first time if SnapSure determines there is insufficient SavVol space and extends the SavVol. The next restore attempt succeeds because needed space is available.

Note: The restore operation for a corrupted PFS proceeds normally, but generates the following LOG_INFO message: “PFS on volume <‘volume ID’> is marked corrupted when ckpt restore is attempted."

If you create a checkpoint of a PFS, extend the PFS, and then try to restore the PFS by using its smaller checkpoint, this action does not shrink the PFS, nor does it free the volume space used for the extension. Rather, the system reserves the space used for the extension and allows you to reclaim it the next time you extend the PFS. You only need to extend the PFS by a small amount, for example 2 MB, to reclaim the original extension space for the new extension request. The 2 MB is added to the original extension size to create the new extension.

Access checkpoints online

SnapSure supports online access to checkpoints. Online access eliminates system administrator involvement when a client needs to list, view, or copy a point-in-time file or directory in a read-only checkpoint, or use it to restore information back to a point in time.

To access online, mounted, read-only checkpoints in the PFS namespace, clients can use:

- EMC CVFS on page 28
- Microsoft SCSF on page 28

EMC CVFS

CVFS version 2, available in systems using versions 4.2 and later, supports both CIFS and NFS clients. It simulates virtual directory links to previous versions of file system objects on the PFS. CVFS smoothes navigation through these virtual links as if the virtual directories are read-only directories on the PFS. The virtual checkpoints require no storage.

CVFS also enables clients to change the name of the virtual checkpoint directory from its default name, .ckpt, and to change the name of the checkpoints listed in the directory, as appropriate. The CVFS functionality is enabled on VNX for File by default. Access a checkpoint by using CVFS on page 65 provides details of the procedure for using CVFS.

Note: Writeable checkpoints cannot be accessed through CVFS.

Microsoft SCSF

SnapSure supports Microsoft Windows SCSF, a feature enabled by Windows Server 2003. SCSF provides Windows Server 2003 and Windows XP clients direct access to point-in-time versions of their files in checkpoints created with SnapSure by providing a Previous Versions tab listing all folders available in the checkpoint shadow-copy directory.

Note: The SCSF software is preloaded on Windows Server 2003 clients. Windows XP clients must install the SCSF software, which you can download from the Microsoft website. To download the Shadow Copy Client, visit the Microsoft website, and download the ShadowCopyClient.msi file.

In systems using version 5.6, SnapSure does not support the SCSF feature for Windows 2000 clients. Contact your EMC Customer Support Representative for the latest SnapSure support information.

Access a checkpoint by using SCSF on page 72 provides details of the procedure for using SCSF.

Use automated checkpoint scheduling

With SnapSure, you can automate the creation and refresh of read-only checkpoints.

Automated checkpoint refresh can be configured with the CLI nas_ckpt_schedule command, Unisphere, or a Linux cron job script. The scheduling created with Unisphere is not visible from and cannot be modified by using the UNIX crontab command. All checkpoint schedules created with Unisphere can be managed by using the CLI, and all checkpoint schedules created with the nas_ckpt_schedule command can be managed by using Unisphere.

Note: Writeable checkpoints cannot be scheduled.

You must have appropriate VNX for File administrative privileges to use the various checkpoint scheduling and management options. Administrative roles that have read-only privileges can only list and view checkpoint schedules. Roles with modify privileges can list, view, change, pause, and resume checkpoint schedules. Roles with full-control privileges can create and delete checkpoint schedules in addition to all other options. *Security Configuration Guide for File* provides more details on administrative roles and privileges.

Using Unisphere to automate checkpoints

You can easily create and manage automated checkpoint schedules with Unisphere. You can schedule checkpoint creation and refreshes on arbitrary, multiple hours of a day and days of a week or month. You can also specify multiple hours of a day on multiple days of a week, more than one schedule per PFS, name scheduled checkpoints, name and describe each schedule, and query the schedule associated with a checkpoint. You can also create a schedule of a PFS with a checkpoint created on it, and modify existing schedules.

A checkpoint virtual file system (CVFS) is a link to the checkpoint mount point and is used only in the hidden directory /PFS/.ckpt. This .ckpt directory is hidden by design and cannot be made visible. By default, a CVFS name is generated based on its creation time, which is maintained by the Control Station. For example, a checkpoint created on May 12, 2010 at 12:27:29 GMT is named /.ckpt/2010_05_12_12.27.29_GMT. This method of naming a CVFS is called time-based CVFS naming. However, you can specify a CVFS name by using the relative prefix option while creating the checkpoint schedule. To use this option, you must specify the relative CVFS name prefix, the delimiter, and the starting index. The delimiter and starting index values are recorded in the nas_param database and must be manually edited. The complete CVFS name while using the relative prefix option is "<prefix>+<delimiter>+<starting_index>". An example is "nightly.001" where "nightly" is the prefix, "." is the delimiter, and "001" is the starting index.

The prefix must be a PFS-wide unique string and can contain up to 20 ASCII characters. The prefix must not include intervening spaces, colons (:), or slashes (/). The delimiter must be either a period (.) or an underscore (_). The starting index must be either 0 or 1.

Note: You must have the appropriate VNX for File administrative privileges to schedule checkpoint operations by using Unisphere.

Unisphere online help provides more details.

Creating multiple schedules

SnapSure supports up to 96 checkpoints, scheduled or otherwise, per PFS, as system resources permit. To avoid system resource conflicts, schedules should not have competing refresh times. For example, if you plan to create three schedules, all the three should not each have a refresh time occurring at 2:10 P.M. on Monday. Instead, the times on the schedules should be staggered, similar to:

- Schedule on ufs1: Monday 2:10 P.M.
- Schedule on ufs2: Monday 2:20 P.M.
- Schedule on ufs3: Monday 2:30 P.M.

Note: Do not schedule a checkpoint-refresh from 1 to 5 minutes past the hour because this conflicts with an internal VNX for File database backup process. Allow at least 15 minutes between creation or refresh of SnapSure checkpoints of the same PFS. This includes checkpoints in the same schedule or between schedules that run on the same PFS.

Minimizing scheduled checkpoint creation or refresh failures

SnapSure checkpoint creation and refresh failures can occur if the schedule conflicts with other background processes, such as the internal VNX for File database backup process that occurs from 1 to 5 minutes past the hour. If a refresh failure occurs due to a schedule or resource conflict, you can manually refresh the affected checkpoint, or let it automatically refresh in the next schedule cycle.

When scheduled tasks are missed because resources are temporarily unavailable, the schedules are automatically retried for a maximum of 15 times, each time sleeping for 15 seconds before retrying. Retries do not occur on such conditions as network outages or insufficient disk space.

Note: Scheduled refresh operations on checkpoints that are baseline checkpoints fail until the associated writeable checkpoints are deleted.

Note: Writeable checkpoints cannot be refreshed.

Refresh-failure events are sent to the /nas/log/sys_log file. You can also configure the VNX for File system to provide e-mail or SNMP trap notification of refresh failures. Facility and event ID numbers for SNMP and e-mail event notification on page 82 provides the facility and event numbers to use. *Configuring Events and Notifications on VNX for File* provides details of the procedures.

Note: Allow at least 15 minutes between creation or refresh of SnapSure checkpoints of the same PFS. This includes checkpoints in the same schedule, or between schedules that run on the same PFS. Unisphere helps to verify and prevent the 15-minute checkpoint operations overlap during schedule creation and modification. This restriction does not apply when using the CLI.

Work with NDMP backup-created checkpoints

When listing checkpoints created on a PFS, a temporary checkpoint created by the NDMP backup facility appears in the list. This type of checkpoint, named automaticTempNDMPCkpt<id>-<srcFsid>-<timestamp>, is created by the system on the request of an NDMP client software, such as EMC NetWorker®, and is meant for backups only.

To use a checkpoint for backup automatically, you must set the NDMP environment variable SNAPSURE=y on the NDMP client software. Then a checkpoint is created automatically and employed when the NDMP backup is initiated. After the backup is complete, the checkpoint is automatically deleted. If you manually delete the temporary NDMP checkpoint, no warning message appears and the NDMP backup fails.

Note: Writeable checkpoints cannot be created from NDMP temporary checkpoints.

If the NDMP variable SNAPSURE is not configured or supported by the DMA software, you can set the NDMP.snapsure parameter to use SnapSure. The NDMP SNAPSURE variable always overrides the NDMP.snapsure parameter. Table 2 on page 31 shows how the NDMP SNAPSURE variable and NDMP.snapsure parameter work together. *Configuring NDMP Backups on VNX* provides more details.

Table 2. Creating a checkpoint with NDMP SnapSure variable and parameter

NDMP variable SNAPSURE (set in DMA)	NDMP.snapsure parameter (set in VNX for File)	Checkpoint created
Y	1	Yes
Y	0	Yes
N	1	No
N	0	No
Not set	1	Yes
	0	No

NOTICE Do not unmount, refresh, or delete the checkpoint in use by NDMP backup, otherwise the backup fails.

Work with other VNX features

Consider these limitations when working with:

VNX file-level retention

You can create read-only checkpoints of a VNX file-level retention (FLR) file system. However, as with any file system, use caution when restoring an FLR file system from one of its checkpoints. Suppose you create a checkpoint of any file system and write a file (for example, file1) into the file system, and later restore the file system by using the checkpoint, then the file (file1) does not remain. This is because the file did not exist when the checkpoint was created. VNX provides a warning when the restore of an FLR file system is attempted. If the warning appears, perform one of the following:

- At the CLI — Use the fs_ckpt -Restore -Force command.
- On Unisphere — Click OK at the confirmation pop-up window.

Using VNX File-Level Retention provides more details.

Note: Writeable checkpoints are not supported with file-level retention.

SRDF

If the VNX environment is configured with EMC SRDF® protection and you plan to create read-only checkpoints of a PFS residing on an RDF volume, ensure that the entire SnapSure volume, including the SavVol, which stores the checkpoints, resides in the same pool of SRDF volumes used to create the PFS. Otherwise, if any part of the SavVol is stored on a local standard (STD) volume rather than on an RDF volume, the checkpoints are not failed over and thus are not recoverable in the SRDF-failback process.

Note: Writeable checkpoints are not supported with SRDF.

Using SRDF/S with VNX for Disaster Recovery provides more details.

VNX FileMover

The VNX FileMover feature supports SnapSure checkpoints. However, with writeable checkpoints, the support is limited to:

- Creation of stub files
- Recall of files for write operations

Note: The VNX fs_dhsm command is not supported on writeable checkpoints.

Using VNX FileMover provides more details.

Data migration

Do not perform data migration tasks by using VNX File System Migration version 2.0 for NFS and CIFS while SnapSure is active. Migration activity produces significant changes in the data environment. Complete the migration tasks before using SnapSure to avoid consuming SnapSure resources for capturing information unnecessary to checkpoint.

TimeFinder/FS

You can use the EMC TimeFinder/FS feature to create and mirror on a snapshot of a PFS while using SnapSure to create a checkpoint of the PFS. However, the information in the snapshot might differ from that in the checkpoint. This is because, as the snapped TimeFinder/FS copy completes, changes can occur on the PFS that is captured only by the checkpoint.

Note: Writeable checkpoints are not supported with TimeFinder.

Replication

Writeable checkpoints cannot be created from VNX Replicator internal checkpoints. These checkpoints are identified by the prefix "root_rep_ckpt". For example, root_rep_ckpt_1372_ v40.

VDMs

If a PFS is mounted on a Virtual Data Mover (VDM), the checkpoint must be mounted on the same VDM.

If a checkpoint is mounted on a VDM and if the VDM is moved to a different Data Mover, active checkpoints persist. Checkpoint schedules resume at the next scheduled runtime after the move occurs. The CVFS name parameter is set to that of the new Data Mover

where the VDM is loaded. The CVFS timestamp is adjusted to the new Data Mover time and time zone. The SCSF timestamp also gets adjusted to the new Data Mover's time, but not the new time zone.

ATA drives

SnapSure allows the PFS and the SavVol to be on an Advanced Technology-Attached (ATA) drive. It also allows the PFS to be on a non-ATA drive and the SavVol to be on an ATA drive. You can specify an ATA drive for the SavVol by using both the pool and the storage option when creating the first PFS checkpoint. EMC recommends that the PFS and the SavVol be on the same type of storage system.

VNX 1.0 Release Notes provide the most recent technical information on using SnapSure with other VNX features.

Guidelines

When using SnapSure in your business environment, configure the system to provide e-mail or SNMP trap notification for the following events:

- A SavVol or PFS becomes full or reaches its percent-full threshold, also known as HWM.
- A scheduled checkpoint refresh attempt fails or the schedule fails to run.

Facility and event ID numbers for SNMP and e-mail event notification on page 82 provides the facility and event numbers to use. *Configuring Events and Notifications on VNX for File* provides more details about configuring e-mail and SNMP trap notifications.

3 Configuring

The tasks to configure SnapSure are:

- Create a read-only checkpoint on page 36
- Create a writeable checkpoint on page 38

Create a read-only checkpoint

Before you begin

- When creating checkpoints, do not exceed the system limit. VNX for File permits 96 read-only and 16 writeable checkpoints per PFS. This is regardless of whether the PFS is replicated or not, for all systems except the Model 510 Data Mover, which permits 32 checkpoints with PFS replication and 64 without. This limit counts existing checkpoints or those already created in a schedule. The limit may also count two internal checkpoints created by certain replication operations on either the PFS or SFS. If you are at the limit, delete existing checkpoints to create space for newer ones or decide not to create new checkpoints at all if existing ones are more important.
- Allow at least 15 minutes between the creation or refresh of SnapSure checkpoints of the same PFS. This includes checkpoints created by using the CLI, and those created or refreshed in an automated schedule, or between schedules that run on the same PFS.
- Be aware when you start to replicate a file system because the facility must be able to create two checkpoints. Otherwise, replication fails to start. For example, if you have 95 checkpoints and want to start a replication, the 96th checkpoint gets created, but replication fails when the system tries to create the 97th checkpoint because the limit is breached.

Procedure

With SnapSure, you can create a read-only checkpoint of a PFS.

Action
To create a read-only checkpoint with default storage and checkpoint name assignments, use this command syntax: `$ fs_ckpt <fs_name> -Create` where: `<fs_name>` = name of the PFS Example: To create a checkpoint of PFS ufs1, type: `$ fs_ckpt ufs1 -Create`

Output

```
operation in progress (not interruptible)...id = 61
name = ufs1
acl = 0
in_use = True
type = uxfs
worm = off
volume = v181
pool = clar_r5_economy
member_of = root_avm_fs_group_4
rw_servers= server_2
ro_servers=
rw_vdms =
ro_vdms =
auto_ext = no,virtual_provision=no
ckpts = ufs1_ckpt1
stor_devs = APM00062400708-0013
disks = d31
 disk=d31 stor_dev=APM00062400708-0013 addr=c16t1l3
server=server_2
 disk=d31 stor_dev=APM00062400708-0013 addr=c0t1l3
server=server_2
id = 63
name = ufs1_ckpt1
acl = 0
in_use = True
type = ckpt
worm = off
volume = vp184
pool = clar_r5_economy
member_of =
rw_servers=
ro_servers= server_2
rw_vdms =
ro_vdms =
checkpt_of= ufs1 Wed Oct 17 17:45:02 EDT 2007
used = 1%
full(mark)= 90%
stor_devs = APM00062400708-0012
disks = d25
 disk=d25 stor_dev=APM00062400708-0012 addr=c0t1l2
server=server_2
 disk=d25 stor_dev=APM00062400708-0012 addr=c16t1l2
server=server_2
```

Note
◆ In the example, a checkpoint name is not specified in the command line. Therefore, SnapSure assigned a default name to it, ufs1_ckpt1, based on the PFS name. ◆ A metavolume or volume pool for the SavVol is not specified in the command line. Therefore, SnapSure assigned a volume from the volume pool by using the same pool where the PFS is built, named vp<n>, where n = volume ID by using AVM. ◆ The HWM for the SavVol used in the creation of the checkpoint, 90% by default, overrides any previously set HWM. The default HWM for subsequent checkpoint creation and refresh is the current HWM value. In other words, to change the HWM for the SavVol, create a new checkpoint and specify the HWM. ◆ The in_use=True field shows that the checkpoint is in mounted state.

Create a writeable checkpoint

Before you begin

- When creating checkpoints, do not to exceed the system limit. VNX for File permits 96 read-only and 16 writeable checkpoints per PFS. This is regardless of whether the PFS is replicated or not, for all systems except the Model 510 Data Mover, which permits 32 checkpoints with PFS replication and 64 without. This limit counts existing checkpoints or those already created in a schedule. The limit may also count two internal checkpoints created by certain replication operations on either the PFS or SFS. If you are at the limit, delete existing checkpoints to create space for newer ones or decide not to create new checkpoints at all if existing ones are more important.
- Allow at least 15 minutes between the creation or refresh of SnapSure checkpoints of the same PFS. This includes checkpoints created by using the CLI, and those created or refreshed in an automated schedule, or between schedules that run on the same PFS.
- Be aware when you start to replicate a file system because the facility must be able to create two checkpoints. Otherwise, replication fails to start. For example, if you have 95 checkpoints and want to start a replication, the 96th checkpoint gets created, but replication fails when the system tries to create the 97th checkpoint because the limit is breached.

Procedure

With SnapSure, you can create a writeable checkpoint from a read-only (baseline) checkpoint.

Action
To create a writeable checkpoint with default storage and name assignments, use this command syntax: `$ fs_ckpt <fs_name> -Create -readonly n` where: `<fs_name>` = name of the baseline checkpoint

Action
Example: To create a writeable checkpoint from baseline checkpoint ufs1_ckpt1, type: **$ fs_ckpt ufs1_ckpt1 -Create -readonly n**

Output

```
operation in progress (not interruptible)...id        = 73
name      = ufs1_ckpt1
acl       = 0
in_use    = True
type      = ckpt
worm      = off
volume    = vp196
pool      = clar_r5_economy
member_of =
rw_servers=
ro_servers= server_2
rw_vdms   =
ro_vdms   =
checkpt_of= ufs1 Wed Oct 17 17:45:02 EDT 2007
ckpts     = ufs1_ckpt1_writeable1
used      = 9%
full(mark)= 90%
stor_devs = APM00062400708-0012
disks     = d25
 disk=d25   stor_dev=APM00062400708-0012 addr=c0t1l2
server=server_2
 disk=d25   stor_dev=APM00062400708-0012 addr=c16t1l2
server=server_2
id        = 90
name      = ufs1_ckpt1_writeable1
acl       = 0
in_use    = True
type      = wckpt
worm      = off
volume    = vp196
pool      = clar_r5_economy
member_of =
rw_servers= server_2
ro_servers=
rw_vdms   =
ro_vdms   =
checkpt_of= ufs1
baseline_ckpt = ufs1_ckpt1 Wed Oct 17 17:45:02 EDT 2007
used      = 9%
full(mark)= 90%
stor_devs = APM00062400708-0012
disks     = d25
 disk=d25   stor_dev=APM00062400708-0012 addr=c0t1l2
server=server_2
 disk=d25   stor_dev=APM00062400708-0012 addr=c16t1l2
server=server_2
```

Note
In the example, a writeable checkpoint name is not specified in the command line. Therefore, SnapSure assigned a default name to it, ufs1_ckpt1_writeable1, based on the baseline checkpoint name.

4 Managing

The tasks to manage SnapSure are:

Modify the SavVol

The tasks to modify the SavVol are:

Modify the SavVol HWM

Action
To modify the SavVol HWM, use this command syntax: `$ fs_ckpt <fs_name> -modify %full=<value>` where: `<fs_name>` = name of the PFS `<value>` = percentage threshold permitted for the SavVol Example: To set the SavVol HWM to 95% of file system ufs1, type: `$ fs_ckpt ufs1 -modify %full=95`

Output

```
operation in progress (not interruptible)...id        = 61
name      = ufs1
acl       = 0
in_use    = True
type      = uxfs
worm      = off
volume    = v181
pool      = clar_r5_economy
member_of = root_avm_fs_group_4
rw_servers= server_2
ro_servers=
rw_vdms   =
ro_vdms   =
auto_ext  = no,virtual_provision=no
ckpts     = ufs1_ckpt2,ufs1_ckpt2_writeable1,ufs1_ckpt6,
ufs1_ckpt7,ufs1_ckpt1,ufs1_ckpt3,ufs1_ckpt4,ufs1_ckpt5
stor_devs = APM00062400708-0013,APM00062400708-0012
disks     = d31,d25
 disk=d31   stor_dev=APM00062400708-0013 addr=c16t1l3
server=server_2
 disk=d31   stor_dev=APM00062400708-0013 addr=c0t1l3
server=server_2
 disk=d25   stor_dev=APM00062400708-0012 addr=c0t1l2
server=server_2
 disk=d25   stor_dev=APM00062400708-0012 addr=c16t1l2
server=server_2
```

Set the SavVol extension limit

Action

To set the SavVol extension limit, use this command syntax:

$ **fs_ckpt** *<fs_name>* **-modify maxsavsize=***<integer>*

where:

<fs_name> = name of the PFS

<integer> = maximum value for the SavVol size in MB

Example:

To set the SavVol extension limit to 1024000 MB of file system ufs1, type:

```
$ fs_ckpt ufs1 -modify maxsavsize=1024000
```

Output

```
operation in progress (not interruptible)...id        = 61
name      = ufs1
acl       = 0
in_use    = True
type      = uxfs
worm      = off
volume    = v181
pool      = clar_r5_economy
member_of = root_avm_fs_group_4
rw_servers= server_2
ro_servers=
rw_vdms   =
ro_vdms   =
auto_ext  = no,virtual_provision=no
ckpts     = ufs1_ckpt2,ufs1_ckpt2_writeable1,ufs1_ckpt6,
ufs1_ckpt1,ufs1_ckpt3,ufs1_ckpt4,ufs1_ckpt5,ufs1_ckpt7
stor_devs = APM00062400708-0013
disks     = d31
 disk=d31   stor_dev=APM00062400708-0013 addr=c16t1l3
server=server_2
 disk=d31   stor_dev=APM00062400708-0013 addr=c0t1l3
server=server_2
```

Extend the SavVol size

Action

To extend the SavVol size, use this command syntax:

$ **nas_fs -xtend** *<fs_name>* **size=***<integer>*

where:

<fs_name> = name of the checkpoint file system

<integer> = maximum value for the SavVol size in GB

Example:

To extend the SavVol size by 10 GB for the file system ufs1, select a checkpoint for the file system (ufs1_ckpt1) and type:

$ nas_fs -xtend ufs1_ckpt1 size=10

Output

```
id         = 61
name       = ufs1_ckpt1
acl        = 0
in_use     = True
type       = uxfs
worm       = off
volume     = v181
pool       = clar_r5_economy
member_of  = root_avm_fs_group_4
rw_servers= server_2
ro_servers=
rw_vdms    =
ro_vdms    =
auto_ext   = no,virtual_provision=no
ckpts      = ufs1_ckpt2, ufs1_ckpt1,ufs1_ckpt2_writeable1,
ufs1_ckpt6,ufs1_ckpt1,ufs1_ckpt3,uf
s1_ckpt4,ufs1_ckpt5,ufs1_ckpt7
stor_devs = APM00062400708-0013,APM00062400708-0012
disks      = d31,d25
 disk=d31   stor_dev=APM00062400708-0013 addr=c16t1l3
server=server_2
 disk=d31   stor_dev=APM00062400708-0013 addr=c0t1l3
server=server_2
 disk=d25   stor_dev=APM00062400708-0012 addr=c0t1l2
server=server_2
 disk=d25   stor_dev=APM00062400708-0012 addr=c16t1l2
server=server_2
```

Change the percentage of system space allotted to SavVol

1. Log in to the Control Station.
2. Open /nas/site/nas_param with a text editor.

   A short list of configuration lines appears.
3. Locate this SnapSure configuration line in the file:

   ckpt:10:200:20:

   where:

   10 = Control Station polling interval rate in seconds

   200 = maximum rate at which a file system is written in MB/s

   20 = percentage of the entire system volume allotted to the creation and extension of all the SavVols used by VNX for File software features

   Note: If the line does not exist, the SavVol space allotment parameter is currently set to its default value of 20. To change this setting, you must first add the line: ckpt:10:200:20:

4. Change the third parameter, which is the percentage of entire system volume allotted to SavVols. The values are 1–100 and the default is 20.

   To ensure proper SnapSure functionality, do not use a value below 20 percent for the third value.

   Do not change the Control Station event polling interval rate, default =10, or the maximum rate to which a file system is written, default =200. Doing so impacts system performance negatively. Do not change any other lines in this file without a thorough knowledge of the potential effects on the system. Contact your EMC Customer Support Representative for more information.

5. Save and close the file.

Note: Changing this value does not require a Data Mover or Control Station reboot.

List checkpoints

The tasks to list checkpoints are:

- List PFS checkpoints with creation dates and SavVol details on page 46
- List PFS checkpoints with PFS details on page 48
- List individual PFS checkpoint details on page 49
- List contents of a checkpoint directory on page 50

List PFS checkpoints with creation dates and SavVol details

Action
To list PFS checkpoints in the order of creation with checkpoint creation dates and SavVol details, use this command syntax: `$ fs_ckpt <fs_name> -list` where: `<fs_name>` = name of the PFS Example: To list checkpoints of PFS ufs1, type: `$ fs_ckpt ufs1 -list`

Output

```
id  ckpt_name  creation_time          inuse fullmark total   ckpt
                                                   _savvol  _usage_on
                                                   _used    _savvol
73  ufs1_ckpt1 10/17/2007-17:45:02-EDT  y    90%       12%     2%
74  ufs1_ckpt2 10/17/2007-18:45:02-EDT  y    90%       12%     2%
75  ufs1_ckpt3 10/17/2007-19:45:03-EDT  y    90%       12%     2%
76  ufs1_ckpt4 10/17/2007-20:45:02-EDT  y    90%       12%     2%
77  ufs1_ckpt5 10/17/2007-21:45:02-EDT  y    90%       12%     2%

id  wckpt_name           inuse fullmark total     base   ckpt
                                       _savvol           _usage_on
                                       _used             _savvol
90  ufs1_ckpt1_writeable1  y    90%       12%      73      2%
91  ufs1_ckpt2_writeable1  y    90%       12%      74      2%
```

Note

- In the example, ufs1_ckpt5 is the newest checkpoint of the PFS and ufs1_ckpt1 is the oldest. The order of checkpoint creation is the order in which they appear. When SnapSure creates checkpoints by using default names (that is, the PFS name followed by ckpt and a numeric suffix), it uses the first available suffix. Thus, when the oldest checkpoints become inactive, and a new one is created, its numeric suffix might not indicate the order of its creation. Also, when you refresh any checkpoint, SnapSure automatically changes its order on the list to reflect it is the most recent point-in-time file system.
- The y in the inuse field shows that the checkpoints are mounted.
- The value in the fullmark field is the current SavVol HWM.
- The value in the total_savvol_used field is the cumulative total of the SavVol used by all PFS checkpoints and not each individual checkpoint in the SavVol. Deleting the latest checkpoint does not free SavVol space immediately. Therefore, the value in the total_savvol_used field might not be accurate. If NOT ACCESSIBLE appears in the used field, the Data Mover is inaccessible.
- The value in the ckpt_usage_on_savvol field is the SavVol space used by a specific checkpoint.
- The values displayed in the total_savvol_used and ckpt_usage_on_savvol fields are rounded up to the nearest integer. Therefore, the displayed sum of all ckpt_usage_on_savvol values might not equal the total_savvol_used value.
- The checkpoint creation_time timestamp is the time and time zone of the local Control Station. The value in the base field is the baseline checkpoint ID from which the writeable checkpoint is created.
- The file system name might be truncated if it is more than 19 characters. To display the full file system name, list it by using the file system ID by replacing <fs_name> with id=<fs_ID> in the preceding command syntax.
- The list option displays NDMP backup-created checkpoints but not VNX Replicator internal checkpoints. To view all checkpoints, including VNX Replicator checkpoints, use the fs_ckpt <fs_name> -list -all option.

List PFS checkpoints with PFS details

Action
To list PFS checkpoints in the order of creation with the mount state, storage pool, and volume used for the PFS, and the Data Mover on which the checkpoints are mounted, use this command syntax: `$ nas_fs -info <fs_name>` where: *<fs_name>* = name of the PFS Example: To list checkpoints of PFS ufs1, type: `$ nas_fs -info ufs1`
Output

```
id         = 61
name       = ufs1
acl        = 0
in_use     = True
type       = uxfs
worm       = off
volume     = v181
pool       = clar_r5_economy
member_of  = root_avm_fs_group_4
rw_servers= server_2
ro_servers=
rw_vdms    =
ro_vdms    =
auto_ext   = no,virtual_provision=no
ckpts      = ufs1_ckpt1,ufs1_ckpt1_writeable1,ufs1_ckpt2,
ufs1_ckpt2_writeable1,ufs1_ckpt3,ufs1_ckpt4,ufs1_ckpt5
stor_devs = APM00062400708-0013
disks      = d31
 disk=d31   stor_dev=APM00062400708-0013 addr=c16t1l3
server=server_2
 disk=d31   stor_dev=APM00062400708-0013 addr=c0t1l3
server=server_2
```

Note
◆ In the example, ufs1_ckpt5 is the newest PFS checkpoint and ufs1_ckpt1 is the oldest. The order of checkpoint creation is the order in which they appear. When SnapSure creates checkpoints by using default names (that is, the PFS name followed by ckpt and a numeric suffix), it uses the first available suffix it finds. Thus, when the oldest checkpoints become inactive, and a new one is created, its numeric suffix might not indicate the creation order. Writeable checkpoints are listed next to their baseline checkpoints. Also, when you refresh any checkpoint, SnapSure automatically changes its order on the list to reflect that it is the most recent point-in-time file system. ◆ The file system name might be truncated if it is more than 19 characters. To display the full file system name, list it by using the file system ID by replacing <fs_name> with id=<fs_ID> in the preceding command syntax. ◆ You can list the checkpoints of a PFS in the order of creation and by common checkpoint-name component, such as ckpt, but without the creation date, by using the grep ckpt option with the nas_fs -list command.

List individual PFS checkpoint details

Action
To list the details of an individual PFS checkpoint including size and pool type, use this command syntax: `$ nas_fs -info -size <fs_name>` where: *<fs_name>* = name of the checkpoint Example: To list the details of checkpoint ufs1_ckpt1, type: `$ nas_fs -info -size ufs1_ckpt1`

Output

```
id          = 73
name        = ufs1_ckpt1
acl         = 0
in_use      = True
type        = ckpt
worm        = off
volume      = vp196
pool        = clar_r5_economy
member_of   =
rw_servers=
ro_servers= server_2
rw_vdms     =
ro_vdms     =
checkpt_of= ufs1 Wed Oct 17 17:45:02 EDT 2007
ckpts       = ufs1_ckpt1_writeable1
size        = volume: total = 10000 avail = 8915
used = 1085 ( 12% ) (sizes in MB)
ckptfs: total = 1008375 avail = 1008371 used = 4 ( 0% )
(sizes in MB) ( blockcount = 20480000 ) ckpt_usage_on_savevol: 8MB ( 2% )
used        = 12%
full(mark)= 90%
stor_devs = APM00062400708-0012
disks       = d25
 disk=d25   stor_dev=APM00062400708-0012 addr=c0t1l2
server=server_2
 disk=d25   stor_dev=APM00062400708-0012 addr=c16t1l2
server=server_2
```

Note

- In the example, ufs1_ckpt1 is a baseline checkpoint that has a writeable checkpoint, ufs1_ckpt1, associated with it.
- Checkpoint ufs1_ckpt1 uses volume pool 196, represented by vp196, for its SavVol. Additional checkpoints created of ufs1 share this SavVol.
- The SavVol is at 12% capacity. It autoextends by 20 GB when it reaches 90% capacity. The HWM for the SavVol used in the creation of the latest checkpoint (90% by default) overrides any previously set HWM. The default HWM for subsequent checkpoint creation and refresh is the current HWM value. In other words, you can change the HWM for the SavVol when you create a new checkpoint.
- The file system name might be truncated if it is more than 19 characters. To display the full file system name, list it by using the file system ID by replacing <fs_name> with id=<fs_ID> in the preceding command syntax.

List contents of a checkpoint directory

In Microsoft Windows Vista, Microsoft Windows 2008, and Windows 2007, which use SMB2, you cannot list the contents of a .ckpt directory just by typing cd .ckpt in the MS-DOS console.

While performing common operations, the SMB2 client uses its directory-caching feature to significantly reduce the number of network round trips to the server. The client sends a query to the server to list all the contents of the current directory. It populates its directory cache with these contents so that subsequent listings can be performed directly from the

cache. However, in response to this query, the server does not include the .ckpt directory as part of the contents of its parent directory.

To access checkpoint files, perform the following:

1. Right-click the user's directory (for example, `C:\Documents and Settings`) from Windows Explorer.
2. Select **Properties**.
3. Click the **Previous Versions** tab.

   The **Previous Versions** dialog box appears and displays all available checkpoints, which is the same as the content of the .ckpt directory. This is the recommended method for accessing checkpoint files.

   You can type dir .ckpt in the MS-DOS console to list the contents of a checkpoint directory. The dir.ckpt command displays the dates of the different checkpoints. To display checkpoints for a particular date, time, and time zone from this list, you can type cd .ckpt\2008_10_08_11.27.02_CEST. Alternatively, you can also use Windows Explorer and define the absolute path in the address bar. For example, T:\.ckpt\2008_10_08_11.27.02_CEST.

Note: You can use cd .ckpt with SMB2 by adding an entry to the registry. However, this may affect performance. You can disable the directory caching feature of the SMB2 client by adding the following registry entry:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters
DirectoryCacheLifetime=0 (REG_DWORD)

Refresh a checkpoint

Before you begin

- If a checkpoint contains important data, back it up or use it before you refresh it.
- Allow at least 15 minutes between the creation or refresh of SnapSure checkpoints of the same PFS. This includes checkpoints created by using the CLI, and those created or refreshed in an automated schedule, or between schedules that run on the same PFS.
- Baseline and writeable checkpoints cannot be refreshed.

Procedure

With SnapSure, you can refresh a checkpoint to recycle the SavVol space. Checkpoint refresh deletes the data in the checkpoint of a PFS and creates a new checkpoint by using the same checkpoint filename, file system ID, and mount state. The CVFS name of the refreshed checkpoint will remain unchanged.

Action
To refresh a checkpoint, use this command syntax:

Action
$ **fs_ckpt** *<fs_name>* **-refresh** where: *<fs_name>* = name of the checkpoint Example: To refresh checkpoint ufs1_ckpt1, type: **$ fs_ckpt ufs1_ckpt1 -refresh**

Output

```
operation in progress (not interruptible)...id        = 73
name      = ufs1_ckpt1
acl       = 0
in_use    = True
type      = ckpt
worm      = off
volume    = vp196
pool      = clar_r5_economy
member_of =
rw_servers=
ro_servers= server_2
rw_vdms   =
ro_vdms   =
checkpt_of= ufs1 Thu Oct 18 12:56:29 EDT 2007
used      = 9%
full(mark)= 75%
stor_devs = APM00062400708-0012
disks     = d25
 disk=d25   stor_dev=APM00062400708-0012 addr=c0t1l2
server=server_2
 disk=d25   stor_dev=APM00062400708-0012 addr=c16t1l2
server=server_2
```

Note
In the example, the HWM value for the SavVol is 90%. If you do not specify an HWM for the SavVol in the -refresh command line, SnapSure, by default, uses the current HWM value of the SavVol.

Delete a checkpoint

Before you begin

- If a checkpoint contains important data, back it up or use it before you delete it.
- Deleting the latest checkpoint does not free the SavVol space immediately.
- There is no fs_ckpt command for deleting checkpoints, instead use the nas_fs command.
- Deleting a checkpoint does not affect the point-in-time view of the other checkpoints. List checkpoints on page 46 provides details on viewing all checkpoints.

- SnapSure does not automatically delete inactive checkpoints.
- The SavVol space from deleted checkpoints is recycled for new checkpoints. When you delete all PFS checkpoints, SnapSure automatically deletes the corresponding SavVol.
- To delete a checkpoint, you must first unmount the checkpoint.
- You cannot delete a checkpoint if:
  - It is mounted, in use, part of a group, or if it is scheduled, but not yet created.
  - It is a baseline checkpoint. If you attempt to delete a baseline checkpoint, SnapSure displays an error message: “Error 10293: ufs1_ckpt1 has a writeable checkpoint associated with it."
- Deleting a checkpoint also deletes the corresponding CVFS name from the .ckpt directory.

Procedure

Delete unnecessary checkpoints to reuse SavVol space.

Action
To unmount and delete a checkpoint, use this command syntax: `$ nas_fs -delete <fs_name> -option umount=yes` where: *<fs_name>* = name of the checkpoint Example: To unmount and delete checkpoint usf1_ckpt2, type: `$ nas_fs -delete ufs1_ckpt2 -option umount=yes`
Output
id = 73 name = ufs1_ckpt1 acl = 0 in_use = False type = ckpt worm = off volume = rw_servers= ro_servers= rw_vdms = ro_vdms =

Restore PFS from a checkpoint

Before you begin

- Do not create checkpoints under nested mount points. If you do so by using Volume Shadow Copy Service (VSS), you cannot restore the PFS from a checkpoint of the PFS without first permanently unmounting the checkpoints and the PFS from the nested mount file system (NMFS). Then you must mount the PFS and its checkpoint to another mount point, perform a restore with VSS, then reverse the process to mount the PFS and its checkpoints on the NMFS again.
- Before restoring a PFS, ensure that enough space exists on the system to support the operation. If SnapSure needs to extend the SavVol during the restore process, the restore might fail upon first attempt. Retry the restore after SnapSure extends the SavVol. Scope resource requirements for SavVol on page 21 provides more details.
- If you create a checkpoint of any file system and write a file (for example, file1) into the file system, and later restore the file system by using the checkpoint, then the file (file1) does not remain. This is because the file did not exist when the checkpoint was created.

Procedure

With SnapSure, you can restore a PFS to a point in time from a read-only or writeable checkpoint.

Action
To restore a PFS from a checkpoint, use this command syntax: `$ /nas/sbin/rootfs_ckpt <fs_name> -Restore` where: `<fs_name>` = name of the checkpoint Example: To restore the PFS from checkpoint ufs1_ckpt1, type: `$ /nas/sbin/rootfs_ckpt ufs1_ckpt1 -Restore`

Output

```
operation in progress (not interruptible)...id        = 73
name      = ufs1_ckpt1
acl       = 0
in_use    = True
type      = ckpt
worm      = off
volume    = vp196
pool      = clar_r5_economy
member_of =
rw_servers=
ro_servers= server_2
rw_vdms   =
ro_vdms   =
checkpt_of= ufs1 Wed Oct 17 17:45:02 EDT 2007
ckpts     = ufs1_ckpt1_writeable1
used      = 13%
full(mark)= 75%
stor_devs = APM00062400708-0012
disks     = d25
 disk=d25   stor_dev=APM00062400708-0012 addr=c0t1l2
server=server_2
 disk=d25   stor_dev=APM00062400708-0012 addr=c16t1l2
server=server_2
```

Note

- In the example, ufs1_ckpt1 is the name of the checkpoint from which to restore the PFS. The PFS name is unnecessary because SnapSure inherently associates a checkpoint with its PFS. SnapSure prohibits restoring a PFS from a checkpoint of another PFS.
- An HWM of 75% is set for the SavVol when the new checkpoint is created in the restore process. This value can be changed by specifying the %full option with an integer value between 10 and 75.
- The checkpoint has used 13% of the SavVol space. When it reaches 75%, the SavVol automatically extends by 10 GB, as space permits, to keep the checkpoints active.
- If you are restoring a PFS extended since the time the checkpoint which you are restoring from was created, read Restore from a checkpoint on page 27.
- When the restore operation completes, the message "Restore completed successfully" appears in the /nas/log/sys_log file.

Schedule checkpoint operations

With SnapSure, you can automate the creation and refresh of read-only checkpoints. Checkpoint schedules and scheduled checkpoint operations continue on the failover Control Station in the event of a primary Control Station outage or failure. You can also manage checkpoint schedules on the failover Control Station.

Before you begin

- Ensure that you have appropriate administrative privileges. Administrative roles that have read-only privileges can only list and view checkpoint schedules. Roles with modify privileges can list, view, change, pause, and resume checkpoint schedules. Roles with full-control privileges can create and delete checkpoint schedules in addition to all other options. *Security Configuration Guide for File* provides more details on administrative roles and privileges.
- Allow at least 15 minutes between the creation or refresh of SnapSure checkpoints of the same PFS. This includes checkpoints created by using the CLI and those created or refreshed in an automated schedule, or between schedules that run on the same PFS.
- SnapSure allows the same checkpoint name to be used when creating different checkpoint schedules of the same PFS, and of different PFSs. No alert or error message appears. Avoid the use of the same checkpoint names when creating multiple schedules. Use SnapSure default checkpoint names to avoid this problem.
- All checkpoint schedules created with the Unisphere software can be managed by using the CLI, and all checkpoint schedules created with the nas_ckpt_schedule command can be managed by using the Unisphere software.
- Ensure that baseline checkpoints are not scheduled for refresh operations. Scheduled refresh operations on checkpoints that are baseline checkpoints fail until the associated writeable checkpoints are deleted.
- Do not keep any checkpoints that would surpass the limit. Otherwise, you cannot start a replication.
- Writeable checkpoints cannot be scheduled.
- To customize the delimiter and starting index for CVFS names, you must manually modify the nas_param database. The modified parameters will be applied only to newly created schedules or to schedules for which the cvfsname_prefix was modified after the delimiter and starting index parameters were modified.
- To keep the delimiter and starting index parameter configurations the same across system upgrades, ensure that you copy the ckpt_schedule entry from the /nas/sys/nas_param database to the /nas/site/nas_param database and modify it. This is because the /nas/sys/nas_param database is overwritten during the system upgrade.

Create a one-time checkpoint schedule

Action
To schedule a one-time checkpoint creation at a future date, use this command syntax: `$ nas_ckpt_schedule -create <name> -filesystem <name> -recurrence once -start_on <YYYY-MM-DD> -runtimes <HH:MM> {-cvfsname_prefix <prefix>| -time_based_cvfsname}` where: `<name>` = name of the checkpoint schedule; the name must be a system-wide unique name and can contain up to 128 ASCII characters, including a–z, A–Z, 0–9, and period (.), hyphen (-), or underscore (_). Names cannot start with a hyphen, include intervening space, or contain all numbers `<name>` = name of the file system of which to create the checkpoint; you can also use the file system ID; the syntax for the file system ID is filesystem id=<n> `<YYYY-MM-DD>` = date on which the checkpoint creation occurs `<HH:MM>` = time at which the checkpoint creation occurs; specify this value by using the 24-hour clock format; for example, 23:59 `<prefix>` = prefix for the customized relative CVFS name; the prefix must be a PFS-wide unique string and can contain up to 20 ASCII characters; prefixes must not include intervening spaces, colons (:), or slashes (/) Example: To create an automated checkpoint schedule to occur on October 5, 2008 at 10:22, type: `$ nas_ckpt_schedule -create ufs01_ckpt_sch -filesystem ufs01 -recurrence once -start_on 2008-10-05 -runtimes 10:22 -cvfsname_prefix morning`
Output
None

Create a daily checkpoint schedule

Action
To schedule the creation of daily checkpoints, use this command syntax: `$ nas_ckpt_schedule -create <name> -filesystem <name> -recurrence daily -every <number_of_days> -start_on <YYYY-MM-DD> -end_on <YYYY-MM-DD> -runtimes <HH:MM> -keep <number_of_ckpts> {-cvfsname_prefix <prefix>| -time_based_cvfsname}` where:

Action
<name> = name of the checkpoint schedule; the name must be a system-wide unique name and can contain up to 128 ASCII characters, including a–z, A–Z, 0–9, and period (.), hyphen (-), or underscore (_); names cannot start with a hyphen, include intervening space, or contain all numbers *<name>* = name of the file system of which to create the checkpoint; you can also use the file system ID; the syntax for the file system ID is filesystem id=<n> *<number_of_days>* = every number of days to create the checkpoints; for example, 2 generates checkpoints every 2 days; the default is 1 *<YYYY-MM-DD>* = dates on which to start and end the checkpoint schedule; the schedule takes effect immediately unless start_on is specified and runs indefinitely unless end_on is specified; the start date cannot be in the past and an end date cannot be before the start date *<HH:MM>* = time or times at which the checkpoint creation occurs on the specified days; specify this time by using the 24-hour clock format; for example, 23:59; use a comma to separate each runtime *<number_of_ckpts>* = number of checkpoints to keep or create before the checkpoints are refreshed *<prefix>* = prefix for the customized relative CVFS name; the prefix must be a PFS-wide unique string and can contain up to 20 ASCII characters; prefixes cannot include intervening spaces, colons (:), or slashes (/) Example: To create an automated daily checkpoint schedule occurring every day at 10:22, starting on October 5, 2008 and ending on October 11, 2008, and keeping the latest three checkpoints before refreshing them, type: **$ nas_ckpt_schedule -create ufs01_ckpt_daily -filesystem ufs01-recurrence daily -every 1 -start_on 2008-10-05 -end_on 2008-10-11 -runtimes 10:22 -keep 3 -cvfsname_prefix morning**
Output
None

Create a weekly checkpoint schedule

Action
To schedule the creation of weekly checkpoints, use this command syntax: $ **nas_ckpt_schedule -create** *<name>* **-filesystem** *<name>* **-recurrence weekly -every** *<number_of_weeks>* **-days_of_week** *<days>* **-start_on** *<YYYY-MM-DD>* **-end_on** *<YYYY-MM-DD>* **-runtimes** *<HH:MM>* **-keep** *<number_of_ckpts>* {**-cvfsname_prefix** *<prefix>*| **-time_based_cvfsname**} where: *<name>* = name of the checkpoint schedule; the name must be a system-wide unique name and can contain up to 128 ASCII characters, including a–z, A–Z, 0–9, and period (.), hyphen (-), or underscore (_); names cannot start with a hyphen, include intervening space, or contain all numbers

Action
<name> = name of the file system of which to create the checkpoint; you can also use the file system ID; the syntax for the file system ID is filesystem id=<n> *<number_of_weeks>* = every number of weeks to create the checkpoints; for example, 2 runs the checkpoint schedule bimonthly; the default is 1 *<days>* = days of the week (Mon, Tue, Wed, Thu, Fri, Sat, Sun) to run the checkpoint schedule; use a comma to separate each day *<YYYY-MM-DD>* = dates on which to start and end the checkpoint schedule; the schedule takes effect immediately unless start_on is specified and runs indefinitely unless end_on is specified; the start date cannot be in the past and an end date cannot be before the start date *<HH:MM>* = time or times at which the checkpoint creation occurs on the specified days; specify this time by using the 24-hour clock format; for example, 23:59; use a comma to separate each runtime *<number_of_ckpts>* = number of checkpoints to keep or create before the checkpoints are refreshed *<prefix>* = prefix for the customized relative CVFS name; the prefix must be a PFS-wide unique string and can contain up to 20 ASCII characters; prefixes cannot include intervening spaces, colons (:), or slashes (/) Example: To create an automated checkpoint schedule occurring every week on Wednesday at 10:22, starting on October 5, 2008 and ending on October 11, 2009, and keeping the latest four checkpoints before refreshing them, type: **$ nas_ckpt_schedule -create ufs01_ckpt_weekly -filesystem ufs01 -recurrence weekly -every 1 -days_of_week Wed -start_on 2008-10-05 -end_on 2009-10-11 -runtimes 10:22 -keep 4 -cvfsname_prefix morning**
Output
None

Create a monthly checkpoint schedule

Action
To schedule the creation of monthly checkpoints, use this command syntax: *$* **nas_ckpt_schedule -create** *<name>* **-filesystem** *<name>* **-recurrence monthly -every** *<number_of_months>* **-days_of_month** *<days>* **-start_on** *<YYYY-MM-DD>* **-end_on** *<YYYY-MM-DD>* **-runtimes** *<HH:MM>* **-keep** *<number_of_ckpts>* {**-cvfsname_prefix** *<prefix>*| **-time_based_cvfsname**} where: *<name>* = name of the checkpoint schedule; the name must be a system-wide unique name and can contain up to 128 ASCII characters, including a–z, A–Z, 0–9, and period (.), hyphen (-), or underscore (_); names cannot start with a hyphen, include intervening space, or contain all numbers *<name>* = name of the file system of which to create the checkpoint; you can also use the file system ID; the syntax for the file system ID is filesystem id=<n>

Action
<number_of_months> = every number of months to create the checkpoints; for example, 3 runs the checkpoint schedule every 3 months; the default is 1 *<days>* = one or more days of the months to run the automated checkpoint schedule; specify an integer between 1 and 31; use a comma to separate the days *<YYYY-MM-DD>* = date or dates on which to start and end the checkpoint schedule; the default is the current date; the schedule takes effect immediately unless start_on is specified and runs indefinitely unless end_on is specified *<HH:MM>* = time or times at which the checkpoint creation occurs on the specified days; the default is the current time; specify this time by using the 24-hour clock format; for example, 23:59; use a comma to separate each runtime *<number_of_ckpts>* = number of checkpoints to keep or create before the checkpoints are refreshed *<prefix>* = prefix for the customized relative CVFS name; the prefix must be a PFS-wide unique string and can contain up to 20 ASCII characters; prefixes cannot include intervening spaces, colons (:), or slashes (/) Example: To create an automated checkpoint schedule occurring every month on the 30th, except February, at 10:22, starting on October 5, 2008 and ending on October 11, 2009, and keeping the latest four checkpoints before refreshing them, type: **$ nas_ckpt_schedule -create ufs01_ckpt_monthly -filesystem ufs01 -recurrence monthly -every 1 -days_of_month 30 -start_on 2008-10-05 -end_on 2009-10-11 -runtimes 10:22 -keep 4 -cvfsname_prefix morning**
Output
None

List all checkpoint schedules

Action
To list all automated checkpoint schedules, type: $ **nas_ckpt_schedule -list**

Output

```
Id           = 4
Name         = ufs01_ckpt_daily
Description =
CVFS name prefix = morning
Next Run     = Sun Oct 05 10:22:00 EDT 2008

Id           = 6
Name         = ufs01_ckpt_monthly
Description =
CVFS name prefix = morning
Next Run     = Thu Oct 30 10:22:00 EST 2008

Id           = 3
Name         = ufs01_ckpt_sch
Description =
CVFS name prefix = morning
Next Run     = Sun Oct 05 10:22:00 EDT 2008

Id           = 5
Name         = ufs01_ckpt_weekly
Description =
CVFS name prefix = morning
Next Run     = Wed Oct 08 10:22:00 EDT 2008
```

Display checkpoint schedule information

Action

To display detailed information of a checkpoint schedule, use this command syntax:

$ **nas_ckpt_schedule -info** *<name>*

where:

<name> = name of the checkpoint schedule

Example:

To display detailed information of checkpoint schedule ufs01_ckpt_weekly, type:

$ nas_ckpt_schedule -info ufs01_ckpt_weekly

Output

```
Id                      = 5
Name                    = ufs01_ckpt_weekly
Description             =
CVFS name prefix        = morning
Tasks                   = Checkpoint ckpt_ufs01_ckpt_weekly_001
on file system id=281,  Checkpoint ckpt_ufs01_ckpt_weekly_002
on file system id=281,  Checkpoint ckpt_ufs01_ckpt_weekly_003
on file system id=281,  Checkpoint ckpt_ufs01_ckpt_weekly_004
on file system id=281
Next Run                = Wed Oct 08 10:22:00 EDT 2008
State                   = Pending
Recurrence              = every 1 weeks
Start On                = Sun Oct 05 00:00:00 EDT 2008
End On                  = Wed Oct 11 23:59:59 EST 2009
At Which Times          = 10:22
On Which Days of Week   = Wed
On Which Days of Month  =
```

Modify a checkpoint schedule

Action

To change the properties of an automated checkpoint schedule, use this command syntax:

$ **nas_ckpt_schedule -modify** *<name>* **-name** *<new_name>* {**-cvfsname_prefix** *<prefix>*| **-time_based_cvfsname**} **-recurrence** *{daily | weekly | monthly}* **-every** {*<number_of_days>* | *<number_of_weeks>* | *<number_of_months>*} **-start_on** *<YYYY-MM-DD>* **-end_on** *<YYYY-MM-DD>* **-runtimes** *<HH:MM>*

where:

<name> = name of the checkpoint schedule or checkpoint schedule ID

<new_name> = new name of the checkpoint schedule

<prefix> = prefix for the customized relative CVFS name; the prefix must be a PFS-wide unique string and can contain up to 20 ASCII characters; prefixes cannot include intervening spaces, colons (:), or slashes (/)

{daily|weekly|monthly} = interval of the checkpoint schedule; if you change the recurrence of a checkpoint schedule, you must also change the associated *<number_of_days>*, *<number_of_weeks>*, or *<number_of_months>*; for example, *<days_of_week>* is rejected if monthly is set for the checkpoint schedule interval

<number_of_days> = every number of days to create the checkpoints; for example, 2 generates checkpoints every 2 days

<number_of_weeks> = every number of weeks to create the checkpoints; for example, 2 runs the checkpoint schedule biweekly

<number_of_months> = every number of months to create the checkpoints; for example, 3 runs the checkpoint schedule every 3 months

<YYYY-MM-DD> = date or dates on which to start and end the checkpoint schedule; you can change the start date only if the checkpoint schedule is pending

Action
<HH:MM> = time or times at which the checkpoint creation occurs on the specified days; specify this time by using the 24-hour clock format; for example, 23:59; use a comma to separate each runtime Example: To change the name of the automated checkpoint schedule from ufs01_ckpt_sch to run_daily_checkpoint and to reschedule it to run every day at 10:22, starting on October 5, 2008 and ending on October 11, 2008, type: **$ nas_ckpt_schedule -modify ufs01_ckpt_sch -name run_daily_checkpoint -cvfsname_prefix morning -recurrence daily -every 1 -start_on 2008-10-05 -end_on 2008-10-11 -runtimes 10:22**
Output
None

Pause a checkpoint schedule

Action
To pause an active checkpoint schedule, stopping all associated tasks including checkpoint creations, use this command syntax: $ **nas_ckpt_schedule -pause** *<name>* where: *<name>* = name of the checkpoint schedule Example: To pause checkpoint schedule ufs01_ckpt_sch, type: **$ nas_ckpt_schedule -pause ufs01_ckpt_sch**
Output
None

Resume a paused checkpoint schedule

Action
To resume a paused automated checkpoint schedule, which restarts all associated tasks, use this command syntax: `$ nas_ckpt_schedule -resume <name>` where: `<name>` = name of the checkpoint schedule Example: To resume checkpoint schedule ufs01_ckpt_sch, type: `$ nas_ckpt_schedule -resume ufs01_ckpt_sch`
Output
None

Delete a checkpoint schedule

Deleting an automated checkpoint schedule does not delete the checkpoints in the schedule, but moves them out from the automatic-refresh cycle. You can refresh or delete these checkpoints manually.

Action
To delete an automated checkpoint schedule, use this command syntax: `$ nas_ckpt_schedule -delete <name>` where: `<name>` = name of the checkpoint schedule Example: To delete checkpoint schedule ufs01_ckpt_sch, type: `$ nas_ckpt_schedule -delete ufs01_ckpt_sch`
Output
None

Access a checkpoint online

Point-in-time copies of client files or directories in read-only checkpoints can be accessed online.

You can:

- Access a checkpoint by using CVFS on page 65
- Access a checkpoint by using SCSF on page 72

Note: Writeable checkpoints cannot be accessed by using CVFS. Writeable checkpoints do not appear in the virtual checkpoint directory (.ckpt) that lists the checkpoint directories.

Access a checkpoint by using CVFS

You can access SnapSure read-only checkpoints by using CVFS. CVFS version 2 supports NFS and CIFS clients, provides access to checkpoint directories from any directory or subdirectory of the PFS, and hides the checkpoint directory name when the PFS directories are listed by using ls -a for UNIX/NFS clients, or dir for Windows CIFS clients.

Hiding the checkpoint directory from the list provides a measure of access control by requiring clients to know the directory name to access the checkpoint items. Access checkpoints online on page 28 provides more details on CVFS and other checkpoint file access features.

The tasks to access checkpoints by using CVFS version 2 are:

- Rename the virtual checkpoint directory on page 65
- Access checkpoints from an NFS client by using CVFS version 2 on page 66
- Access checkpoints from a CIFS client by using CVFS version 2 on page 68
- Disable checkpoint access with CVFS version 2 on page 70
- Rename a checkpoint with CVFS version 2 on page 70

Rename the virtual checkpoint directory

1. Log in to the Control Station.
2. Rename the virtual checkpoint directory by using this command syntax:

   ```
   $ server_param <movername> -facility cvfs -modify virtualDirName -value
   <new_value>
   ```

   where:

   <movername> = name of the Data Mover

<new_value> = name for the checkpoint directory

Note: Do not specify names that contain .ckpt for files, directories, and links because .ckpt is reserved as a virtual directory entry for NFS or CVFS client access to online checkpoints in the PFS namespace.

Note: The directory name can contain up to 64 alphanumeric characters, including dashes and underscores.

---

3. Reboot the Data Mover by using this command syntax:

   $ **server_cpu** *<movername>* **-reboot -monitor now**

   where:

   <movername> = name of the Data Mover controlled by the slot_<x>/param file; for example, slot_2/param affects server_2

Access checkpoints from an NFS client by using CVFS version 2

1. List a client directory in the PFS to view the existing files and directories by using this command syntax:

   $ **ls -l** *<mount point>/<client_directory>*

   where:

   <mount point> = name of the mount point on which the PFS is mounted

   <client_directory> = name of the client directory in the PFS

   Example:

   To list the files and directories contained in client directory mpl in the PFS mounted on mount point /EMC, type:

   $ ls -l /EMC/mpl

   Output:

```
drwxr-xr-x   2 32771    32772   80 Nov 21  8:05  2003 dir1
drwxr-xr-x   2 root     other   80 Nov 14 10:25  2003 resources
-rw-r--r--   1 32768    32772  292 Nov 19 11:15  2003 A1.dat
-rw-r--r--   1 32768    32772  292 Nov 19 11:30  2003 A2.dat
```

2. Access a virtual checkpoint directory entry named .ckpt (the default name) from the client directory on the PFS. The checkpoints displayed are relative to the directory in which the .ckpt entry resides. Only checkpoints that are mounted and read-only are displayed.

   Example:

   To list the virtual .ckpt entries for client directory mpl on the PFS mounted on /EMC, type:

   $ ls -la /EMC/mpl/.ckpt

Output:

```
drwxr-xr-x 5 root root 1024 Nov 19 08:02 2003_11_19_16.15.43_GMT
drwxr-xr-x 6 root root 1024 Nov 19 11:36 2003_11_19_16.39.39_GMT
drwxr-xr-x 7 root root 1024 Nov 19 11:42 2003_11_20_12.27.29_GMT
```

Note: In the example, CVFS found three virtual checkpoint entries. These entries were created on a Data Mover whose clock is set to Greenwich Mean Time (GMT) by default.

In systems with versions 5.3 and later, use the server_date command to use the local time zone of the Data Mover when constructing the entry name. This is the name that appears in the virtual checkpoint directory whenever a checkpoint is created or refreshed. For example, EST is used for Eastern Standard Time if the Data Mover time zone is so configured by using the server_date command. The VNX for File man pages and *VNX System Operations* provide more details of the server_date command.

If a checkpoint in this list is created before server_date is used to set the time zone, unmount and remount the checkpoint to view the latest timestamp.

If you change the default .ckpt directory name by using the parameter shown in Rename the virtual checkpoint directory on page 65, replace .ckpt with .<name> in steps 2 and 3 of this procedure.

Note: If you have a linked file system and you display checkpoints, two results are possible: (1) if the linked file system has checkpoints, only the checkpoints of the linked file system appear, not the checkpoints of the source file system. You can list the checkpoints of the source file system from any location, except under the linked file system. (2) If the linked file system has no checkpoints, the following error message appears: "Cannot find file or item <'pathname'>."

3. Access a virtual checkpoint from the list of entries to view the point-in-time information available for recovery for that directory.

   Example:

   To view the contents of checkpoint entry 2003_11_19_16.39.39_GMT found on client directory mpl on the PFS mounted on mount point EMC, type:

   ```
   $ ls -l /EMC/mpl/.ckpt/2003_11_19_16.39.39_GMT
   ```

   Output:

   ```
   -rw-r--r--   1 32768    32772    292 Nov 19 11:15 A1.dat
   -rw-r--r--   1 32768    32772    292 Nov 19 11:30 A2.dat
   -rw-r--r--   1 32768    32772    292 Nov 19 11:45 A3.dat
   drwxr-xr-x   2 root     other     80 Nov 14 10:25 resources
   ```

   Note: These read-only items represent a specific point-in-time view of the mpl directory. When these checkpoints were created, the directory dir1 did not exist on the mpl directory, which is why it does not appear in this list, but does when you list the directory's current contents. Conversely, A3.dat appears in this list, but does not appear when you list the directory's current contents. The A3.dat was deleted since the checkpoint was created. Use these point-in-time items to restore those accidentally deleted, or access the items to read point-in-time data.

Access checkpoints from a CIFS client by using CVFS version 2

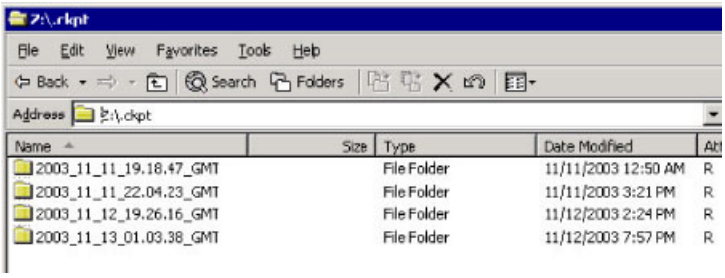
1. In the address field of an Explorer window, type the name of a client directory on the PFS to view the existing files and directories.

   Example:

   ```
   z:\
   ```

2. After the directory name, type .ckpt or the name given to the virtual checkpoint directory to access the virtual directory entry from the client directory on the PFS. The checkpoints displayed are relative to the directory in which the .ckpt entry resides. Only mounted, read-only checkpoints appear.

   Example:



Note: In the example, CVFS found four virtual checkpoint entries. These entries were created on a Data Mover with its clock set to Greenwich Mean Time (GMT) by default.

In systems with versions 5.3 and later, you can use the server_date command to use the local time zone of the Data Mover when constructing the entry name appearing in the virtual checkpoint directory whenever the checkpoint is created or refreshed. For example, EST is used for Eastern Standard Time if the Data Mover time zone is so configured by using the server_date command. The VNX for File man pages and *VNX System Operations* provide more details of the server_date command.

If a checkpoint in this list is created before server_date is used to set the time zone, unmount and remount the checkpoint to view the latest timestamp.

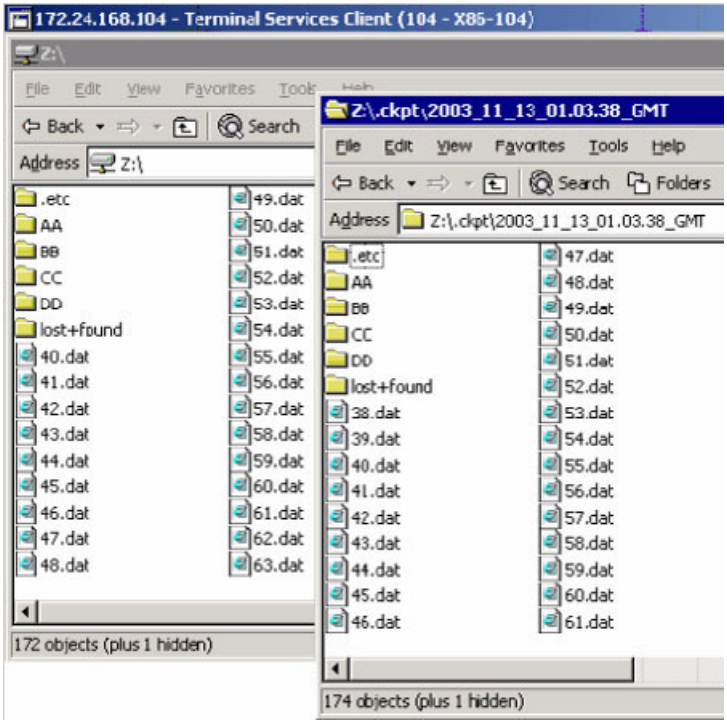
The checkpoint timestamp might indicate a different checkpoint creation or refresh time depending on the interface used to view the checkpoint. The section on differing checkpoint timestamps in Known problems and limitations on page 76 provides more details.

If you have changed the default .ckpt directory name by using the parameter shown in Access checkpoints from an NFS client by using CVFS version 2 on page 66, replace .ckpt with .<name> in steps 2 and 4 of this procedure.

Note: If you have a linked file system and you display checkpoints, two results are possible: (1) Only the checkpoints of the linked file system appear, not the checkpoints of the source file system. You can list the checkpoints of the source file system from any location, except under the linked file system. (2) If the linked file system lacks checkpoints, the following error message appears: “Cannot find file or item <pathname>."

3. Access a virtual checkpoint from the list of entries to view the point-in-time information available for recovery of that directory.

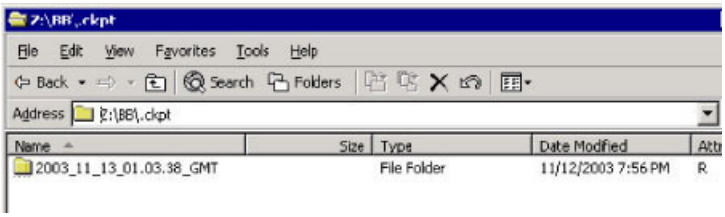
   Example:



Note: In the example, the checkpoint directory contents represent a specific point-in-time view of directory Z:\. Clients can use these point-in-time items to restore missing files accidentally deleted. For example, copy files 38.dat and 39.dat to the top level of directory Z.

4. Clients can also search for virtual checkpoint directories from the subdirectory level as this feature is not restricted to root directories. To do so, type \.ckpt after the subdirectory

name in the Explorer address field to display the virtual checkpoint directories it contains, as shown in the following example.



Note: In the example, only one virtual checkpoint directory is displayed because the other data was nonexistent when this checkpoint was created.

Disable checkpoint access with CVFS version 2

1. Log in to the Control Station.
2. Disable checkpoint access by using this command syntax:

   $ **server_param** *<movername>* **-facility cfs -modify showHiddenCkpt -value 0**

   where:

   <movername> = name of the Data Mover

   Note: NFS and CIFS client access to CVFS version 2 checkpoint directories is enabled by default. Disabling CVFS version 2 also disables SCSF.

3. Reboot the Data Mover by using this command syntax:

   $ **server_cpu** *<movername>* **-reboot -monitor now**

   where:

   <movername> = name of the Data Mover controlled by the slot_*<x>*/param file; for example, slot_2/param affects server_2

Rename a checkpoint with CVFS version 2

1. Log in to the Control Station.
2. List the checkpoints on the server by using this command syntax:

   $ **server_mount** *<movername>*

   where:

   <movername> = name of the Data Mover

   Example:

   To list checkpoints on Data Mover server_2, type:

```
$ server_mount server_2
```

A list of checkpoints similar to the following appears:

```
ufs1_ckpt3 on /ufs1_ckpt3 ckpt,perm,ro
ufs1_ckpt2 on /ufs1_ckpt2 ckpt,perm,ro
ufs1_ckpt1 on /ufs1_ckpt1 ckpt,perm,ro
```

3. Unmount the checkpoint to rename by using this command syntax:

   $ **server_umount** *<movername>* /*<fs_name>*

   where:

   <movername> = name of the Data Mover

   <fs_name> = name of the checkpoint

   Example:

   To unmount checkpoint ufs1_ckpt1 on Data Mover server_2, type:

   ```
   $ server_umount server_2 /ufs1_ckpt1
   ```

   To determine a checkpoint creation date to identify which one to rename, type:

   ```
   nas_fs -info or nas_fs -list
   ```

4. Rename the checkpoint when you mount it by using this command syntax:

   $ **server_mount** *<movername>* **-option cvfsname**=*<newname>* *<fs_name>* *<mount_point>*

   where:

   <movername> = name of the Data Mover on which to mount the checkpoint

   <newname> = new name visible to the NFS or CIFS client

   <fs_name> = current checkpoint name displayed by server_mount

   <mount_point> = mount point of the checkpoint

Note: The custom checkpoint name can contain up to 128 characters and must be a system-wide unique name. Names can include a–z, A–Z, 0–9, and period (.), hyphen (-), or underscore (_). Names cannot start with a period or hyphen.

Example:

To change the name of checkpoint ufs1_ckpt1 to Monday while mounting the checkpoint on Data Mover server_2 at mount point ufs1, type:

```
$ server_mount server_2 -option cvfsname=Monday ufs1_ckpt1 /ufs1
```

Output:

```
ufs1_ckpt3 on /ufs1_ckpt3 ckpt,perm,ro
ufs1_ckpt2 on /ufs1_ckpt2 ckpt,perm,ro
ufs1_ckpt1 on /ufs1_ckpt1 ckpt,perm,ro,cvfsname=Monday
```

When NFS and CIFS clients list the checkpoints by using the ls -l .ckpt command from the virtual checkpoint directory, ufs1_ckpt1 appears as Monday:

```
drwxr-xr-x 6 root root 1024 Nov 19 11:36 2003_11_19_16.39.39_GMT
drwxr-xr-x 7 root root 1024 Nov 19 11:42 2003_11_20_12.27.29_GMT
drwxr-xr-x 5 root root 1024 Nov 19 08:02 Monday
```

Note: Because of filename restrictions associated with DOS version 6.0 and earlier, clients of these DOS-based applications continue to see checkpoint names in the standard DOS filename convention: ddmmyyfile.IDx. For example, 01050400.011. This filename is not changeable.

Access a checkpoint by using SCSF

SCSF provides Windows Server 2003 and Windows XP clients direct access to point-in-time versions of files in checkpoints created with SnapSure.

The tasks to access a checkpoint by using SCSF are:

- (Optional) Enable SCSF on page 72
- Access a checkpoint from the Previous Versions tab on page 73

(Optional) Enable SCSF

1. To enable (or disable) CIFS client access to checkpoint directories through SCSF, use this command syntax:

   $ **server_param** *<movername>* **-facility cifs -modify allowSnapSureVss -value** *<new_value>*

   where:

   <movername> = name of the Data Mover

   <new_value> = value you want to set for the specified parameter; 0 to disable and 1 to enable

   Note: The SCSF feature is enabled by default on the VNX for File system. Disabling SCSF does not disable access to the .ckpt directory.

   Example:

   To disable CIFS client access to checkpoint directories through SCSF, type:

   $ server_param server_2 -facility cifs -modify allowSnapSureVss -value 0

   Output:

   ```
   server_2 : done
   ```

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

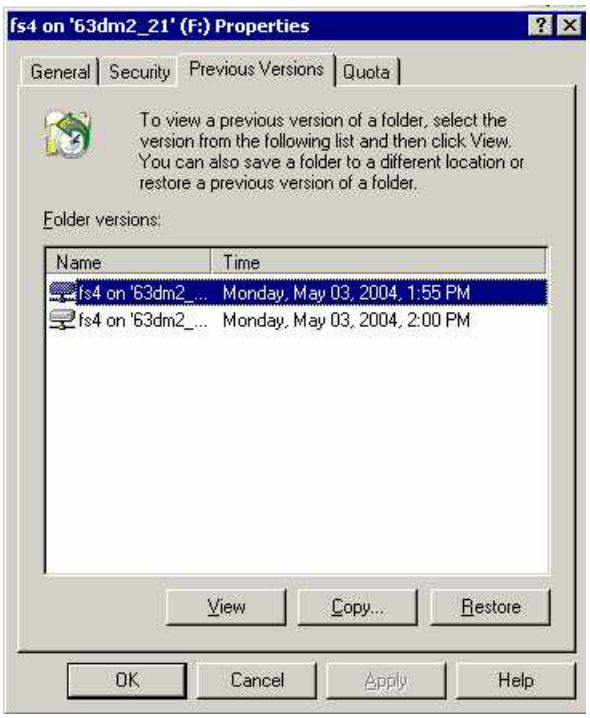
   $ **server_cpu** *<movername>* **-reboot -monitor now**

   where:

<movername> = name of the Data Mover

Access a checkpoint from the Previous Versions tab

1. Right-click a PFS in the directory and select **Properties** from the menu that appears. A new window with a **Previous Versions** tab enabled appears similar to the following figure.



Note: At least one checkpoint for the file or directory must exist for the Previous Versions tab to be visible. If the file is unchanged since the checkpoint was created or refreshed, it does not appear on the Previous Versions tab. Each checkpoint is timestamped with the date it was created or last refreshed.

2. From the list of folders, select a version and click **View**, **Copy**, or **Restore**, as applicable:

   View — Displays the file or directory, but does not modify it.

   Copy — Saves a copy of the file or folder to a different location. A dialog box appears in which you can specify a new name and location for the copy.

   Restore — Uses the selected file or directory to replace the corresponding PFS file or directory. A dialog box appears to confirm or cancel the operation.

   Microsoft online help for Shadow Copy offers detailed usage information.

Note: Two internal checkpoints used for replication sessions will also appear on this screen in a Microsoft Windows machine. This is to be expected.

5 Troubleshooting

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

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

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

Topics included in this chapter are:

EMC E-Lab Interoperability Navigator

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

Return codes for fs_ckpt

This section describes the possible fs_ckpt return codes that can be helpful when checking errors in scripts using the fs_ckpt command. Table 3 on page 76 lists the possible return codes.

Table 3. Return codes and their description

Return code	Description
0	Command completed successfully
1	Usage error
2	Invalid object error
3	Unable to acquire lock
4	Permission error
5	Communication error
6	Transaction error
7	Dart error
8	Backend error

Known problems and limitations

Table 4 on page 77 describes known problems that might occur when using SnapSure and presents workarounds. To diagnose problems, check the VNX system log (/nas/log/sys_log), which records SnapSure event information.

Table 4. Known problems and workarounds

Known problem	Workaround
Checkpoint creation fails	◆ The PFS is not mounted. Ensure that the PFS is mounted. If it is mounted, ensure that it is mounted only on a single Data Mover. If mounted on a VDM, ensure that both the PFS and its checkpoints are mounted on the VDM. ◆ The PFS is corrupted. SnapSure automatically rejects checkpoint creation of a corrupted PFS. ◆ The PFS is being restored and SnapSure cannot access it to create a checkpoint. Retry checkpoint creation when the PFS restoration is complete. ◆ SnapSure checkpoint creation was attempted during the Unicode conversion process, producing the message: "Error 5005: Input/output error." The checkpoint creation might actually succeed, but the checkpoint is not automatically mounted. You must manually mount any checkpoint created during Unicode conversion after the conversion completes and stops. ◆ There is insufficient system or disk space available to create or extend the SavVol. Planning considerations on page 19 provides more details. ◆ A maximum of 96 PFS checkpoints already exists. Delete unwanted checkpoints and retry.
Checkpoint refresh fails	◆ If insufficient space exists in the SavVol to complete the restore process upon first attempt, the restore might fail as SnapSure extends the SavVol. Retry the restore. Note: Use the nas_fs -info command to list a PFS checkpoint to view SavVol usage and the current HWM. ◆ A maximum of 96 PFS checkpoints already exists. The system cannot create the checkpoint required for the restore process. Delete any unwanted checkpoint and retry the restore. ◆ There is insufficient system or memory space available to create or extend the SavVol. Planning considerations on page 19 provides more details.

Table 4. Known problems and workarounds *(continued)*

Known problem	Workaround
Checkpoint restore fails	◆ If insufficient space exists in the SavVol to complete the restore process upon first attempt, the restore might fail as SnapSure extends the SavVol. Retry the restore. Note: Use the nas_fs -info command to list a PFS checkpoint to view SavVol usage and the current HWM. ◆ A maximum of 96 PFS checkpoints already exists. The system cannot create the checkpoint required for the restore process. Delete any unwanted checkpoint and retry the restore. ◆ There is insufficient system or memory space available to create or extend the SavVol. Planning considerations on page 19 provides more details.
Deleting a checkpoint fails	You cannot delete a mounted checkpoint that is being used, such as one being restored from, or a new checkpoint created in the restore process, or is part of a group. Ensure that these conditions do not exist, then retry the delete. You cannot also delete baseline checkpoints.
Mounting a PFS fails	Checkpoint mount operation for a corrupted PFS fails with the LOG_ERROR message: "PFS <'file system ID'> is marked as corrupted, ckpt mount is rejected. You can force mount a checkpoint by setting the SVFS.forceCkptMount parameter to 1."
Unmounting a PFS fails	After a checkpoint is mounted, the PFS can be temporarily unmounted. However, the PFS cannot be permanently unmounted until the checkpoint is unmounted.
Checkpoint access by using CVFS fails	◆ The checkpoint is not mounted and cannot be seen by CVFS. Mount the checkpoint and retry. Note: If the PFS is mounted read-only on multiple Data Movers, you cannot mount one of its checkpoints on the same multiple Data Movers. ◆ The checkpoint-access parameter is changed to disable checkpoint access through CVFS. If appropriate, reset the parameter to enable checkpoint access. Access a checkpoint online on page 65 provides details of the procedure for enabling and disabling CVFS checkpoint access.

Table 4. Known problems and workarounds *(continued)*

Known problem	Workaround
Differing checkpoint timestamps	A checkpoint's creation or refresh timestamp might differ based on the interfaces a checkpoint is viewed from as follows: ◆ CLI by using the fs_ckpt command — Timestamp uses the Control Station time and time zone. ◆ Unisphere — Timestamp uses the Control Station time and time zone. ◆ SCSF — Timestamp uses the Windows client's time zone. ◆ CVFS by using the .ckpt directory — Timestamp uses the Data Mover time and time zone.
Different states of a checkpoint: ◆ Active ◆ Inactive ◆ Restoring	◆ Active — Checkpoint is readable and is maintaining an accurate point-in-time image of the PFS. A checkpoint becomes active when created and remains active unless it is deleted or runs out of space to store point-in-time information. ◆ Inactive — Checkpoint is not readable or accurate due to insufficient resources needed to store point-in-time information. Scope resource requirements for SavVol on page 21 provides more details. ◆ Restoring — Checkpoint is used to restore a PFS to a point in time. Note: You can use the nas_fs -info -all command to verify the current state of a checkpoint. For example, if used = INACTIVE appears in this command output, the checkpoint is inactive.
Different states of a checkpoint schedule: ◆ Active ◆ Complete ◆ Paused ◆ Pending	◆ Active — Schedule started to run and its automated checkpoint creation or refresh cycle is in progress. ◆ Complete — Schedule reached its specified end-on date. It does not run again unless the end-on date is changed. ◆ Paused — Schedule is manually paused. It does not run again until it is manually resumed, at which point, its state becomes Active. ◆ Pending — Schedule is created, but its start-on date is not reached. When the start date is reached, the schedule becomes Active. Note: Schedule states can be checked by using only the Unisphere Replicas ➤ Checkpoints ➤ Schedules tab. You must have appropriate VNX for File administrative privileges to use various checkpoint scheduling options.

Error messages

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

To view message details, use any of these methods:

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

EMC Training and Professional Services

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

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

Appendix A

Facility and Event ID Numbers

The appendix lists the SnapSure facility and event ID numbers. The topic included is:

- Facility and event ID numbers for SNMP and e-mail event notification on page 82

Facility and event ID numbers for SNMP and e-mail event notification

Table 5 on page 82 lists the facility numbers and event IDs to configure the VNX to send SNMP and event notification for SnapSure events. *Configuring Events and Notifications on VNX for File* provides details about the procedure.

Table 5. SnapSure facility and event ID numbers

Facility number	Event ID
70	1 = HWM of SavVol reached 2 = Checkpoint inactive 3 = Source file system restore done 4 = Restore is in process 5 = Checkpoint conversion is paused due to a full SavVol
91	1 = Scheduled SnapSure checkpoint creation failed 2 = Scheduled SnapSure checkpoint refresh failed
137	10 = Checkpoint autoextension failed

Glossary

B

bitmap
Software register that tracks if a Production File System (PFS) block has changed since the latest checkpoint was created.

See also *Production File System*.

blockmap
Data structure that maps the older PFS volume block saved to the SavVol block. Each checkpoint has a blockmap and is pageable to the SavVol from the Data Mover memory.

See also *Production File System* and *PFS*.

blockmap index file
On-disk representation of a checkpoint's blockmap stored with each checkpoint on the SavVol.

C

checkpoint
Point-in-time, logical image of a PFS. A checkpoint is a file system and is also referred to as a checkpoint file system or an EMC SnapSure™ file system.

See also *Production File System.*

checkpoint application
Business application that reads a checkpoint for point-in-time information about a PFS. Checkpoint applications require point-in-time data, but not realtime data.

See also *Production File System*.

checkpoint audit
Process of monitoring the SavVol to determine how much free space remains for capturing checkpoint data. By default, SnapSure automatically audits the SavVol and writes a message to the system log when the SavVol's high water mark (HWM) is reached.

checkpoint refresh
Process of recycling the SavVol space by deleting a PFS checkpoint data and creating a new checkpoint by using the same checkpoint filename, file system identification number, and mount state.

checkpoint restore
Process of restoring a PFS to a point in time by using a checkpoint. As a precaution, SnapSure automatically creates a new checkpoint of the PFS before it performs a restore operation.

See also *Production File System*.

Checkpoint Virtual File System (CVFS)
File system that allows an NFS or a CIFS client read-only access to online snapshots of a file system directory by accessing a virtual .ckpt directory entry. CVFS and SVFS are synonymous terms.

checkpoint window
Length of time a checkpoint is active, beginning when the checkpoint is created and ending when the checkpoint's SavVol is full or the checkpoint is deleted.

P

PFS application
Business application that accesses a PFS and performs transactions, such as read, write, or delete on the PFS blocks.

See also *Production File System*.

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

R

read-only checkpoint
Read-only, point-in-time, logical image of a PFS.

See also *Production File System*.

S

SavVol extend
Process of enlarging the SavVol space to ensure that it does not become full and inactivate one or more checkpoints. By default, and as system resources permit, VNX SnapSure automatically extends the SavVol each time the HWM is reached.

See also *high water mark*.

Shadow Copy for Shared Folders (SCSF)
Microsoft Windows feature that provides Windows XP and Windows Server 2003 clients direct access to VNX SnapSure point-in-time images of their files and folders for the purpose of recovering point-in-time data.

Snapshot Virtual File System (SVFS)

File system that allows an NFS or a CIFS client read-only access to online snapshots of a file system directory by accessing a virtual .ckpt directory entry. SVFS and CVFS are synonymous terms.

SnapSure SavVol

The volume to which SnapSure copies original point-in-time data blocks from the Production File System (PFS) before the blocks are altered by a transaction. SnapSure uses the contents of the SavVol and the unchanged PFS blocks to maintain a checkpoint of the PFS.

W

writeable snapshot

Read-write, point-in-time, logical image of a PFS created from a read-only (baseline) snapshot.

See also *Production File System*.

Index

T

U

V

W