Sun StorEdge™ Availability Suite 3.1 Point-in-Time Copy Software Administration and Operation Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.
650-960-1300

Part No. 816-4313-11
June 2003, Revision A

Submit comments about this document at: `http://www.sun.com/hwdocs/feedback`

Please
Recycle

Adobe PostScript

Contents

Preface

The *Sun StorEdge Availability Suite 3.1 Point-in-Time Copy Software Administration and Operation Guide* is intended for use by the administrator of the system on which the point-in-time copy software is installed, or by qualified Sun technical personnel.

Before You Read This Book

In order to fully use the information in this document, you must have thorough knowledge of the Solaris™ operating environment.

How This Book Is Organized

This book contains the following chapters:

Chapter 1 describes the Sun StorEdge Availability Suite 3.1 point-in-time copy software in functional detail.

Chapter 2 discusses various operational considerations for the Sun StorEdge Availability Suite 3.1 point-in-time copy software.

Chapter 3 contains examples showing how to configure, enable, and use the point-in-time copy software. The command-line interface (CLI) commands are covered in functional order.

Chapter 4 discusses configuration considerations when using Sun StorEdge Availability Suite 3.1 point-in-time copy software with other data services software.

Chapter 5 describes the point-in-time copy software's CLI and provides the syntax of the CLI commands.

Appendix A describes the usage for the `dsstat` command, which enables you to track the performance of the point-in-time copy software.

The Glossary defines terms used in this book.

Using UNIX Commands

This document might not contain information on basic UNIX® commands and procedures such as shutting down the system, booting the system, and configuring devices.

See one or more of the following for this information:

- *Solaris Handbook for Sun Peripherals*
- AnswerBook2™ online documentation for the Solaris operating environment
- Other software documentation that you received with your system

Typographic Conventions

Typeface	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`
`AaBbCc123`	What you type, when contrasted with on-screen computer output	`% su` `Password:`
AaBbCc123	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the *User's Guide*. These are called *class* options. You *must* be superuser to do this. To delete a file, type `rm` *filename*.

Shell Prompts

Shell	Prompt
C shell	*machine-name*%
C shell superuser	*machine-name*#
Bourne shell and Korn shell	$
Bourne shell and Korn shell superuser	#

Related Documentation

Application	Title	Part Number
Release	*Sun StorEdge Availability Suite 3.1 Point-in-Time Copy Software Release Notes*	816-4314
	Sun StorEdge Availability Suite 3.1 Remote Mirror Software Release Notes	816-4414
	Sun Cluster 3.0 and Sun StorEdge Software Release Notes Supplement	816-5128
Installation	*Sun StorEdge Availability Suite 3.1 Point-in-Time Copy Software Installation Guide*	816-4312
	Sun StorEdge Availability Suite 3.1 Remote Mirror Software Installation Guide	816-4413
System administration	*Sun StorEdge Availability Suite 3.1 Remote Mirror Software Administration and Operation Guide*	816-4415
Sun cluster usage	*Sun Cluster 3.0 and Sun StorEdge Software Integration Guide*	816-5127

Accessing Sun Documentation Online

A broad selection of Sun system documentation is located at:

`http://www.sun.com/products-n-solutions/hardware/docs`

A complete set of Solaris documentation and many other titles are located at:

`http://docs.sun.com`

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can email your comments to Sun at:

`docfeedback@sun.com`

Please include the part number (8xx-xxxx-xx) of your document in the subject line of your email.

CHAPTER 1

Introduction

This chapter describes the Sun StorEdge Availability Suite 3.1 point-in-time copy software in functional detail. First, the software's architecture is explained. Then the allowable volume set configurations are discussed in detail followed by a full explanation of how these volume set configurations are tracked and controlled using bitmap volumes. Finally, additional features of the point-in-time copy software are presented.

This chapter is divided into the following main topics:

Point-in-Time Software Architecture

Sun StorEdge Availability Suite 3.1 point-in-time copy software is a *point-in-time snapshot* facility that runs in a Solaris operating environment. A point-in-time snapshot, also called a *point-in-time copy*, is an instantly-available, time-fixed, replicated view of a momentarily-quiesced volume. After a point-in-time copy is taken, you have immediate read and write access to both the original and copy volumes.

Point-in-time copy *shadow volume sets* consist of a *master volume*, a *shadow volume*, a *bitmap volume*, and an optional *overflow volume*. A shadow volume set can be *enabled* in several configurations, which are discussed in this chapter.

The point-in-time software tracks the differences between the master and shadow volumes, caused by writes, from the point in time that the copy is established. This capability allows the data on either of the two volumes to move forward in time independently of the other, which means that applications can access both volumes and modify the data on them independently.

Because the software is tracking differences between the volumes, the volumes can be quickly updated after the first point-in-time copy. A resynchronization of this type can occur either from the shadow volume to the master volume or from the master volume to the shadow volume.

Instantly after the point-in-time copy is established, or re-established, on the shadow volume set, the applications using the shadow volume set can resume processing. The point-in-time copy is established, or re-established, either when the CLI prompt returns or the next shell script command is read.

Architecture

The Sun StorEdge data services are implemented as layered drivers in the Solaris kernel I/O stack. These drivers rely on the `nsctl` framework to support this layering, as well as for runtime control. Point-in-time software is implemented as an `nsctl` I/O filter module, which enables it to be integrated with other Sun StorEdge data services. The architecture of the point-in-time software in the kernel I/O stack is shown in FIGURE 1-1.

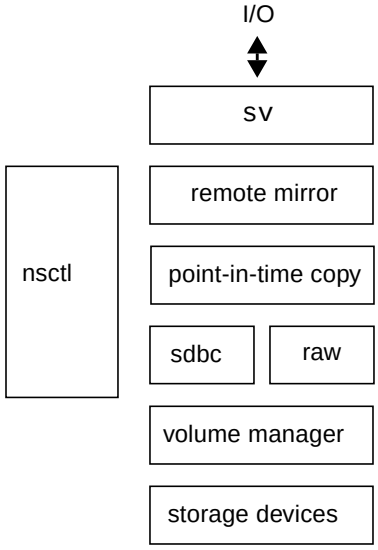

FIGURE 1-1 Point-in-Time Copy Software in the Sun StorEdge Services I/O stack.

The point-in-time copy software works by being in the data path. I/O commands and data enter and exit the point-in-time copy software through the Sun StorEdge storage volume (`sv`) software. Mediated by `nsctl`, the data optionally flows through the remote mirror software and the point-in-time software to the storage device block cache (`sdbc`) drivers and then to its destination either on the storage (for writes) or in application or kernel memory (for reads).

The point-in-time copy software is a Solaris kernel pseudo-device driver. It resides in the `nsctl` framework above the volume manager or the storage device driver, and below the file system. This architecture makes the point-in-time copy software independent of a volume manager or file system using the volume manager.

The point-in-time copy software enables flexibility in how you configure your volumes locally. The volumes can be protected by any RAID level desired. The protection level of the volumes in a shadow volume set does not have to match.

Point-in-Time Copy Software and the Data Services I/O Stack

Data flows to the point-in-time copy software driver from user layer applications accessing the shadow volume set via the SV layer. Sometimes user layer applications reside above the file system. Other times these applications run in *Data Base Management Systems* (DBMS), which can read and write directly to raw disk partitions or volumes created with the volume manager. In any case, I/O commands process the data to its destination on the storage.

The I/O commands targeted to shadow volume sets are intercepted by the SV driver and routed through the Sun StorEdge I/O stack prior to being passed on to either the storage device driver, or the volume manager. The SV layer is a very thin layer in the I/O stack, and operates by interposing onto the DDI entry points to the underlying device driver. I/O commands originating in user space are intercepted at the top of the Sun StorEdge service I/O stack. The SV layer routes them through the Sun StorEdge data services stack, and feeds them back to the storage device driver, or the volume manager, at the bottom of the stack. Data also flows in the opposite direction, from the storage back to user space.

Shadow Volume Sets

The master volume of a shadow volume set is that volume from which it is intended that a point-in-time copy be created. The master volume is the source of the data which will be copied when a shadow volume set is initially enabled. The shadow volume is the volume on which a point-in-time copy is created. At any given time, a master volume can have more than one shadow volume, but a shadow volume can have only one master.

The use of the terms master volume and shadow volume does not dictate the direction of a subsequent point-in-time copy or *update*, however, it serves as a reference point. Which volume is configured as the master volume and which volume is configured as the shadow volume is a choice that depends on how a point-in-time copy is being used.

Shadow volumes can be *independent*, *dependent*, or *compact dependent*. An independent shadow volume can be utilized separately from its corresponding master because a complete duplicate of the master volume is started on the independent shadow volume when the point-in-time copy is initiated.

Dependent and compact dependent shadow volumes cannot be utilized separately from their corresponding master volumes. Both types of dependent shadow volume access their master volume to return the contents of the volume in those areas which have not been written since the point-in-time copy was established.

When a shadow volume set is enabled with an independent shadow volume, it automatically starts to *synchronize* the master and shadow volumes in the shadow volume set. Simply put, the synchronization of an independent shadow volume with its master volume refers to the background process of copying all of the data on the master volume to the shadow volume. In a shadow volume set configured with an independent shadow volume, the shadow volume is treated as a dependent shadow volume until synchronization has completed.

Additional details of how independent and dependent shadow volume sets behave is detailed in the sections “Independent Copy Operation” on page 6 and “Dependent Copy Operation” on page 9. For details about compact dependent shadow volumes, see “Compact Dependent Shadow Volumes” on page 11.

Independent Copy Operation

Shadow volume sets can be configured with independent shadow volumes when *any* of the following apply:

- You want to access the point-in-time copy on the shadow volume without incurring the overhead of an access on the master

  In other words, access performance on either the master volume or the shadow volume is a priority. Independent shadow volume sets divide the accesses between the volumes, and accesses to the shadow incur no I/O on the master.
- The overhead in storage and I/O CPU bandwidth caused by creating a duplicate of the master volume on the shadow volume is not prohibitive;
- The point-in-time copy is intended to serve as a backup copy in case of failure on the master, or the shadow is intended for *export* to another host.

Creating an Independent Shadow Volume

When a shadow volume set is enabled with an independent shadow volume, a *full volume copy* (or simply, *full copy*) is started, and proceeds along two distinct routes:

- The master volume data is sequentially copied to the shadow volume in the background.

  If no writes are sent to the master volume during this synchronization, the process continues to completion as a simple copy.
- Existing master volume data blocks are written to the shadow volume as writes to the master are received.

  Writes to a block on the master volume trigger a write of the existing data in the block to the shadow volume. Then the new data is written to the master. This preserves the validity of the point-in-time copy on the shadow volume.

Upon completion of the full copy, the shadow volume is treated as an independent shadow volume.

At the beginning of the full copy, all the bits in the bitmap for the master volume are set. When a bit in the bitmap is set, indicating that the block has not been synchronized, the block is said to be *changed*. During synchronization, as data is moved from the master volume to the shadow volume, the bits in the bitmap corresponding to the blocks updated are cleared, and the blocks are said to be *unchanged*.

When a write destined for a master volume block which has not been copied to the shadow volume comes through the I/O stack, the block which is the target of the incoming write is processed along with the ongoing synchronization in the following fashion:

1. The data at the block which is the target of the write is copied to the shadow volume.
2. The block on the master volume is updated with the new data.
3. The corresponding bit in the bitmap is cleared.

Since the point-in-time copy software checks each bit to see if the block is changed prior to copying it, when it gets to this block it will skip over it. In this way an independent PIT copy is established on the shadow volume.

Once the background copy completes, the shadow volume is fully independent, and it is possible to perform update or *fast synchronization* point-in-time copies. An update point-in-time copy is created after a full copy has completed on a shadow volume set by copying only those blocks which have been modified since the full copy to the target of the update. Update copies are described in "Resynchronizing the Shadow and Master Volumes" on page 8.

Accessing the Independent Shadow Volume

Independent shadow volumes can be accessed in a variety of ways once established:

- The shadow can be exported to another host. See "Export, Import, and Join of Dual-Ported Shadow Volumes" on page 29.
- The shadow can be kept under point-in-time copy software control, and mounted and accessed independently of the master volume, which enables you to update a point-in-time copy subsequent to such access.
- The shadow volume set can be disabled, removing the shadow volume from point-in-time copy software control, and accessed independently of the master volume.

No matter which approach is taken, I/O on an independent shadow volume is performed directly on the shadow volume, unlike I/O on a dependent shadow volume.

If the shadow volume set is disabled, the master and the shadow volumes no longer bear any relationship to each other, and will diverge over time.

If either of the first two approaches are taken, bitmap management continues, which allows:

- In the case of an exported shadow volume, a later join of the exported shadow

- In the case where point-in-time copy software control is maintained, a later update, or fast, copy to be performed

Joins are explained in "Export, Import, and Join of Dual-Ported Shadow Volumes" on page 29. Update point-in-time copy is explained in "Resynchronizing the Shadow and Master Volumes" on page 8.

If an independent shadow volume is accessed by another host with the export and import commands, a bitmap volume is maintained on the accessing host to track which of the blocks in the shadow are modified by the host. Changes to the master volume are tracked in the bitmap of the originating host.

If an independent shadow volume is not disabled after a full synchronization and remains under point-in-time copy software control, changes to either the master or the shadow volume are tracked in the bitmap of the shadow volume set. Since a single bitmap is used to track which blocks differ between the two volumes, no distinction as to where the modification originated is available.

Resynchronizing the Shadow and Master Volumes

The term *resynchronization* is used to describe a synchronization which occurs between volumes in a shadow volume set which have been previously synchronized.

Synchronizations can be full synchronizations, or update synchronizations. Full synchronization of an independent shadow volume is described in "Creating an Independent Shadow Volume" on page 6.

An update synchronization is a synchronization which copies only those blocks marked changed in the bitmap to the target of the update, which can be the master volume or the shadow volume, depending upon the direction of the synchronization.

Dependent Copy Operation

Shadow volume sets can be configured with dependent shadow volumes when *any* of the following apply:

- The proportion of data changed between point-in-time copies is low.
- The shadow is either never accessed or the overhead of accessing the shadow volume through the master volume is not prohibitive in terms of the performance of accesses to the master or to the shadow.
- The overhead in I/O CPU and bandwidth use engendered in creating a complete duplicate of the master on the shadow is prohibitive;
- The point-in-time copy is not intended to serve as a backup copy in case of failure on the master.
- The shadow will not be exported to another host.

Creating a Dependent Shadow Volume

When a shadow volume set is enabled with a dependent shadow volume, tracking in the bitmap volume of changes made on the master volume is initiated. Enabling a shadow volume set with a dependent shadow volume does not initiate a background synchronization process because all data that has remained unmodified on the master volume since the point-in-time copy was created is accessed on the master volume.

Data is only written to the shadow volume when writes to the master volume commence, which is after the point-in-time copy has been established. When a write destined for the master volume is processed by the point-in-time copy software, the block on the master volume is first copied to the shadow volume. Then, the new block data is written to the master volume, and the associated bit in the bitmap volume is marked changed.

Dependent shadow volumes are available for access immediately because the synchronization process inherent in the creation of an independent shadow volume does not apply.

Note – A dependent shadow volume can not be accessed without the master volume being available.

Accessing the Dependent Shadow Volume

Access to dependent shadow volumes is limited. Dependent shadow volumes can be mounted and can be the target of I/O. The shadow volume set of the dependent shadow volume must remain under point-in-time copy software control and the master volume must be available. This is because dependent shadow volumes are virtual volumes, formed by the union of the unmodified data on the physical master volume and the modified data on the physical shadow volume.

When data is read from a dependent shadow volume, the point-in-time copy software checks in the bitmap to determine if the data has been modified. If it hasn't, data from the block which is the target of the read is read from the master volume and returned to the caller. If the data has been modified, data from the block which is the target of the read is read from the physical shadow volume and returned.

When data is written to a dependent shadow volume, the point-in-time copy software updates the corresponding bit in the bitmap to indicate that the target block is changed, and the data is written to the physical shadow volume. It is the responsibility of the accessing client that this is the intended effect, because the dependent shadow volume is now no longer an accurate reflection of the master volume at the time the point-in-time copy was established.

Resynchronizing the Master Volume to the Shadow Volume

Resynchronization of a dependent shadow volume with its master volume is immediate. It involves only the bitmap volume: all the bits in the bitmap volume are cleared, or marked as unchanged.

Resynchronizing the Shadow Volume to the Master Volume

Resynchronization of a master volume with its dependent shadow volume is termed an update synchronization. In an update synchronization, only the blocks marked changed with a bitmap value of "1" are copied to the target of the copy. In the case of a dependent shadow volume, this would be any blocks modified on either the master volume or the shadow volume since the last point-in-time copy was established.

Compact Dependent Shadow Volumes

The point-in-time copy software supports the creation of *compact dependent shadow volumes*, which are dependent shadow volumes that are smaller than their corresponding master volume. The use of the term compact is intended to convey that less storage is allocated, not that the data in the blocks is compacted or compressed in any way.

Compact volumes are useful when *all* the following statements are true:

- The proportion of blocks marked with a “1” (changed) in the bitmap volume never reaches 100%
- Storage conservation is a priority
- A dependent shadow volume is suitable

Often, applications in user space do not modify the contents of the entire master volume over the course of time between planned point-in-time copies. For many applications, entire areas of storage are modified rarely relative to their neighbors.

If application characteristics or point-in-time copy scheduling is such that you know that this is the case for a particular shadow volume set and that a dependent shadow is appropriate, a compact dependent shadow volume can be used. For example, if you know that at most 10% of the blocks on the master volume are changed between point-in-time copies, a compact dependent shadow volume can be allocated at 10% of the size of the master volume.

The point-in-time copy software keeps track of the updated data blocks using an index in the bitmap. Blocks written to the master are first copied to the next available block in the compact dependent shadow volume. An index is assigned in the bitmap corresponding to the block on the shadow that the data was written to.

As the master volume and the shadow volume diverge, the data on the compact volume grows, and indexes are progressively assigned. If the number of blocks that differ between the master volume and the virtual shadow volume exceeds the number of blocks allocated on the physical shadow, the system will fail. To protect against such a failure, *overflow volumes* can be designated for a compact dependent shadow volume set.

Note – If a compact dependent shadow volume set overflows due to being incorrectly sized or to experiencing an unexpectedly large volume of writes, the point-in-time copy software displays a message indicating that the shadow volume is out of space. The shadow volume is left enabled so that read operations can continue, which allows you to recover data. *However, any subsequent write operations will force the shadow volume offline.*

Overflow Volumes for Compact Dependent Shadow Volumes

An overflow volume can be designated for one or more compact dependent shadow volumes. If a compact dependent shadow volume exceeds its limits (i.e. the number of blocks that differ between the master and the shadow is greater than the number of blocks allocated for the shadow), an attached overflow volume prevents data loss. Overflow volumes can also be exceeded, but careful planning makes the use of compact dependent shadow volumes and overflow volumes attractive and relatively risk-free.

Shadow volume sets configured with both a compact dependent shadow volume and an overflow volume are managed identically to shadow volume sets with a compact dependent shadow volume, except in the case that the shadow exceeds its capacity. When the point-in-time copy software detects that the storage on the compact dependent shadow volume has been exhausted, it starts to write the data on the designated overflow volume. The index in the bitmap volume is augmented to reflect whether the data was written to a block on the shadow, or a block on the overflow volume.

Initializing an Overflow Volume

When a volume is initialized as an overflow volume, information is written to a header area on the volume which the point-in-time copy software uses in order to keep track of how the volume is being used. For example, an overflow volume keeps track of the number of dependent shadow volumes which utilize this volume for overflow data.

The information in this header area is updated when an overflow volume is attached or detached from its corresponding compact dependent shadow volume.

Bitmap Management

The point-in-time copy software uses a bitmap volume in order to create point-in-time copies. For every 32K block of a master volume that is part of a shadow volume set, a bit is maintained which indicates if the data at the block has changed with respect to its associated point-in-time copy. This technique is called *scoreboarding*, and the shadow volume set's bitmap volume is sometimes referred to as the *bitmap* or the *scoreboard* or the *scoreboard log*.

FIGURE 1-2 shows what the master, shadow, and bitmap volume of an independent shadow volume set might look like at some time after a point-in-time copy has been established. In the figure, each 32K block on the master and shadow volumes is represented by a cell. The contents of the cell (e.g. AAA) represents the data in the 32K blocks on the volume. For every block which differs from the master since the point-in-time copy was established, a bit in the bitmap volume is set to a "1". This indicates that the data on the storage has changed since the point-in-time copy.

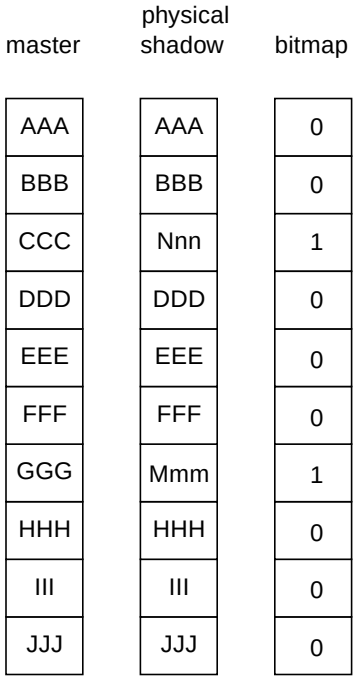

FIGURE 1-2 Independent Shadow Volume Set After a Point-in-Time Copy

FIGURE 1-3 shows what the master, physical shadow, *virtual shadow*, and bitmap volume of a dependent shadow volume set might look like at some time shortly after a point-in-time copy has been established. This figure shows both the *virtual shadow* and the *physical shadow* volumes. The virtual shadow is formed by the union of the master volume at all blocks marked as unchanged ("0") in the bitmap, and the physical shadow at all blocks marked changed ("1") in the bitmap.

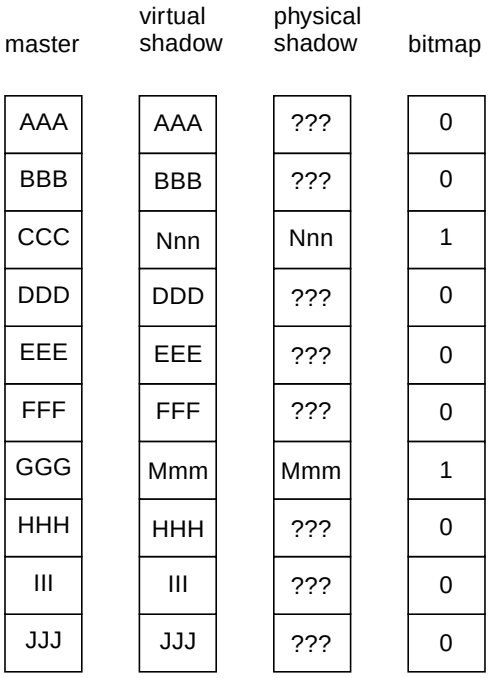

FIGURE 1-3 Dependent Shadow Volume Set After a Point-in-Time Copy

The point-in-time copy software allows the configuration of a compact dependent shadow volume. A compact shadow volume is one which occupies less physical space than the master volume of the shadow volume set. Compact dependent shadow volumes are useful in situations where:

- The proportion of blocks changed never reaches 100%
- Storage conservation is important
- A dependent shadow is suitable

With compact dependent shadow volumes, for every changed block being tracked in the bitmap volume, an index is maintained. It is an index to the block in the compact volume of the data as it existed at the time the point-in-time copy was created.

In this configuration, after a point-in-time copy is made, blocks written to the master are first copied to the compact dependent shadow volume, starting at the first changed block, and the index value is set. As the master volume and the shadow volume change, the data on the compact volume fills up, causing indexes to be progressively assigned. If the number of blocks that differ between the master volume and the virtual shadow volume exceeds the number of blocks allocated on the physical shadow, the following occurs:

- The I/O fails
- The target volume is off-lined
- The source volume becomes the only valid copy of the volume

To prevent this occurrence, overflow volumes can be designated for a compact dependent shadow volume set.

FIGURE 1-4 shows what the master volume, physical shadow volume, virtual shadow volume, and bitmap volume of a compact dependent shadow volume set might look like at some time after a point-in-time copy has been established.

FIGURE 1-4 Compact Dependent Shadow Volume Set After a Point-in-Time Copy

To prevent the problems described above that are associated with overrunning the physical bounds of a compact dependent shadow volume, associate the compact dependent shadow volume with a sharable overflow volume. If the number of blocks that differ between the master volume and the virtual shadow volume exceeds the number of blocks allocated on the compact dependent shadow volume, blocks are copied to the overflow volume. Bitmap management is done in the same way as with compact dependent shadow volumes. An additional index is maintained, to indicate whether an index entry is for the compact shadow volume or the overflow volume.

If the overflow volume itself becomes filled up, then the following occurs:

- The I/O fails
- The target volume is off-lined
- The source volume becomes the only valid copy of the volume

FIGURE 1-5 shows what the master volume, physical shadow volume, virtual shadow volume, and bitmap volume of a compact dependent shadow volume set with an associated overflow volume might look like at some time after a point-in-time copy has been established. In the index, bracketed cells in the example indicate an index to the overflow volume. Note that the first block of an overflow volume contains a header, and is not used for overflow data.

Note – Many compact dependent shadow volumes can be configured to a single overflow volume, but a single compact dependent shadow volume can be configured to only one overflow volume.

Note – Do not create bitmaps on cylinder 0 because the point-in-time copy software does a raw write and destroys the VTOC for that device.

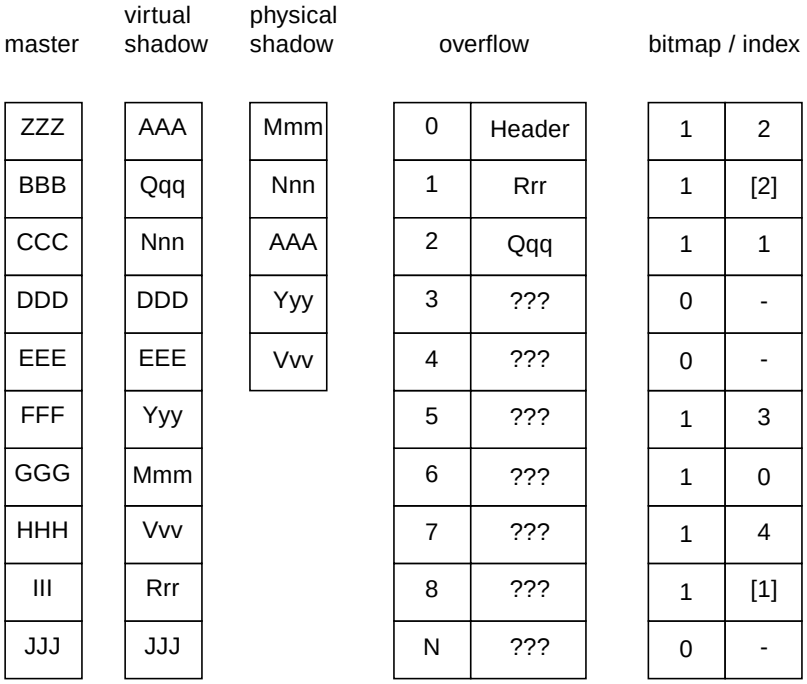

FIGURE 1-5 Compact Dependent Shadow Volume Set with Overflow After a Point-in-Time Copy

Additional Features

Multiple Shadows of a Single Master

Sun StorEdge Availability Suite 3.1 point-in-time copy software allows multiple point-in-time copies to be created from a single master volume. For each of the copies, a shadow volume set must be enabled. Each of the shadow volume sets is maintained according to its type: independent, dependent, compact dependent, or compact dependent with an overflow volume.

Multiple shadow volumes of the same master volume enable the user to perform multiple tasks on identical copies of one master volume. In other words, you can perform many separate analyses of the master data by creating multiple shadow volumes of that master volume.

Export Shadow

An independent shadow volume can be exported so that another host can import and use the shadow for any purpose. In order for the shadow to be exported, it must reside on a dual-ported device. The importing host is required to maintain a bitmap for tracking changes made to the shadow volume while it is imported. The shadow volume and its associated bitmap can be joined to its original master after the importing host has disabled the volume set that includes the shadow.

An exported shadow volume permits you to perform analysis of a point-in-time copy of your master data with no impact on operations involving the master volume. No matter how intensive the analysis is, it is being performed by a host that is separate from the master volume's host.

CHAPTER 2

Operation Considerations

This chapter consists of the following main topics:

Uses for the Point-in-Time Copy Software

The point-in-time copy software, running in the Solaris operating environment, provides applications with continuous access to data and enables nonintrusive access to a point-in-time copy of the same data by secondary applications. The point-in-time copy software supports both full copy and fast resynchronization to reestablish a new point-in-time shadow copy as needed. The volume's data can be resynchronized from either master to shadow or from shadow to master.

The point-in-time copy software supports both Sun StorEdge and all Sun-supported storage. It works independently of the underlying data reliability software (for example: RAID-1, RAID-5, or volume manager). Additionally, it can be an integral part of the data migration to and from differing storage types.

Typical uses include:

- Backup of 24 x 7 application data.
- Data warehouse loading (and fast resynchronization) at predefined intervals.
- Application development and testing on a point-in-time snapshot of live data.
- Data migration across different types of storage platforms and volumes.
- Hot backup of application data from frequent point-in-time snapshots.
- Application data replication to secondary site.

Caution – *Do not* install the Sun StorEdge Availability Suite 3.1 software on servers in a Sun Cluster 3.0 environment.

- The version 3.0/3.1 software is *not* coexistent with the Sun Cluster 3.0 environment
- The version 3.0/3.1 software is coexistent in the Sun Cluster 2.2 environment
- The Sun StorEdge Availability Suite 3.1 software is cluster aware in the Sun Cluster 3.0 Update 1 and Update 2 environments and provides high availability.

Task Summary: Getting Started

This table briefly lists the steps and tasks for getting started with the point-in-time copy software.

TABLE 2-1 Getting Started Task Summary

Step	See This Section (Command-Line Interface)
1. Choosing volumes to use.	"Sizing Requirements for Volume Sets" on page 38
2. Enabling shadow volume sets consisting of master volume, shadow volume, and bitmap volume.	"Enabling Volume Sets" on page 39
3. Initialize and attach overflow volumes to compact dependent shadow volume sets.	"Using Overflow Volumes" on page 41
4. Collect volume sets in I/O groups for ease of management.	"To Move Shadow Volume Sets into I/O Groups" on page 44
5. Perform copy, update, and other operations on the volume sets.	Most of Chapter 3, including: "Updating, Copying, and Waiting" on page 46 "To Perform Full Copy Operations" on page 49 "To Update Shadow Volume Sets" on page 47
6. Check volume set status.	"To Display Shadow Volume Set Status" on page 55

Reading the man pages

If you have not already set the `PATH` and `MANPATH` variables, follow the steps in "Adding the iiadm Command PATH and Man Page MANPATH to Your Shell Environment" on page 22. Doing this enables you to easily reach the man pages.

Adding the iiadm Command PATH and Man Page MANPATH to Your Shell Environment

This section describes how to add the point-in-time copy software command and man page paths to your environment.

▼ To Add the Paths to Your Bourne or Korn Shell

1. **Add** `/usr/opt/SUNWesm/sbin` **to your** `PATH` **statement in your** `.profile` **file.**

   This path enables you to access the point-in-time copy software commands like `iiadm`. For example, edit your `.profile` file in a text editor and add the command path:

   ```
   PATH=$PATH:/usr/opt/SUNWesm/sbin
   export PATH
   ```

   where `$PATH` indicates all other paths in your file.

2. **Add** `/usr/opt/SUNWesm/man` **to your** `MANPATH` **statement in your** `.profile` **file.**

   This path enables you to read the point-in-time copy software man pages.

   ```
   MANPATH=$MANPATH:/usr/opt/SUNWesm/man
   export MANPATH
   ```

   where `$MANPATH` indicates the default man page path of `/usr/share/man` and other man page locations you might have. See the `man`(1M) man page for more information about the `man` command.

3. **Save this file and exit.**

▼ To Add the Paths to Your C Shell

1. **Add** `/usr/opt/SUNWesm/sbin` **to your** `path` **statement in your** `.cshrc` **file.**

   This path enables you to access the point-in-time copy software commands like `iiadm`. For example, edit your `.cshrc` file in a text editor and add the command path:

   ```
   set path = ($path /usr/opt/SUNWesm/sbin)
   ```

   where `$path` indicates all other paths in your file.

2. **Save this file and exit.**

3. **Add** `/usr/opt/SUNWesm/man` **to your** `MANPATH` **statement in your** `.login` **file.**

   This path enables you to read the point-in-time copy software man pages. For example, edit your `.login` file in a text editor and add the command path:

   ```
   setenv MANPATH "$MANPATH:/usr/opt/SUNWesm/man"
   ```

   where `$MANPATH` indicates the default man page path of `/usr/share/man` and other man page locations you might have. See the `man`(1M) man page for more information about the `man` command.

4. **Save this file and exit.**

▼ To Use An Alternate Method to Read Man Pages

These procedures describe how to read man pages without having to add paths to your environment.

- **To read the point-in-time copy software man pages, type:**

```
# man -M /usr/opt/SUNWesm/SUNWii/man iiadm.1m
```

- **To read related manpages, type:**

```
# man -M /usr/opt/SUNWesm/SUNWscm/man/ manpage
```

where *manpage* is one of the following:

manpage	
	`ds.log.4`
	`dscfg.1m`
	`scmadm.1m`

What Happens During System Startup and Shutdown

During system startup and shutdown, the Solaris operating environment changes its mode of operation by transitioning a system's current run-level to the run-level requested (in this case the startup or shutdown run-levels). This transitioning is called run-control.

During startup run-control, the volumes of previously configured shadow volume sets will be resumed. During shutdown run-control, the volumes of previously configured shadow volume sets will be suspended.

Note – The commands for suspending and resuming shadow volume sets are not available to the user.

During these transitions, numerous other Solaris subsystems in the environment, including other Sun StorEdge Services, are also executing startup and shutdown scripts. The scripts used to resume and suspend point-in-time copy volume sets are numerically ordered relative to these other scripts. This ensures that the resume and suspend operations on volume sets are invoked at the appropriate time during startup and shutdown.

Quiescing a Master Volume Considerations

In Solaris operating environments, a disk block is the smallest atomic unit of I/O. Disk blocks are 512 bytes. An I/O operation is atomic if it is guaranteed either to fully complete (all of the data is confirmed to have been written) or to fail (none of the data is written because some part of the data was confirmed to have been unwritable).

Most file systems, databases, and applications create or update an item on a disk in an I/O operation involving more than a single disk block. For example, if you create a file, you need to populate the file, plus enter its existence in a directory. Or, if you create a record in a database, you need to write the record, plus update the index.

Because systems might experience hardware failure, software failure, and might crash or lose power, most file systems, volume managers and databases support a facility or mechanism to repair inconsistencies detected at initial access time. Deterministic repair of the data in these situations is possible, but it may cause partial data to be rolled-back or intentionally discarded.

If a point-in-time copy is established against a non-quiesced volume, it is possible that the point-in-time copy will span non-atomic I/O operations. If this occurs, at initial access time, deterministic repair may be invoked, yielding unpredictable results.

Therefore, it is important that a point-in-time copy or update not be performed against a source volume (master or shadow), without quiescing or stopping all application I/O and flushing any file system caches (`sync`(1M)) associated with the volume. This quiesced or stopped state is only required for the duration of the copy or update operation, which typically takes milliseconds or seconds to complete. As always, the destination volume (master or shadow) must be in the unmounted or unaccessed state.

A notable exception to this rule is encountered in systems, such as Oracle, which support *hot backup*. There is no reason to quiesce a volume set prior to a point-in-time copy if the database is in hot-backup mode. Refer to the specific application's documentation for detailed information, and see any applicable Sun documents available at: `http://docs.sun.com`

Sun StorEdge Availability Suite 3.1 point-in-time copy software provides an attractive complement to a hot-backup facility. Instead of the database being in hot backup mode for the minutes or hours needed to perform disk or tape replication, point-in-time copy software allows it to be in this mode for only milliseconds or seconds.

Point-in-Time Copy of Mounted Volumes Considerations

When the point-in-time copy software makes a copy or update, the source, which is usually the master volume, can be mounted and should be in the quiesced or stopped state. Instantly after the copy or update, the target, which is usually the unmounted shadow volume, contains on-disk metadata that states that the volume is currently mounted, *but it is not*.

When a point-in-time copy is established in this way and when the target volume is first mounted, the software detects that a currently dismounted volume has mounted metadata on it. It usually forces `fsck` to run under these conditions because the assumption is that the only time a volume contains mounted metadata, but is not currently mounted, is after a system crash. Point-in-time copy software breaks this assumption: `fsck` or the database recovery mechanism should return no errors, unless the master was not quiesced when the point-in-time copy was initiated (see "Quiescing a Master Volume Considerations" on page 26).

It is also important to note that the target of a point-in-time copy operation, which is usually the shadow volume, must not be mounted, because the application accessing the target volume will read inconsistent and changing data.

Note – A remote mirror volume set must be in logging mode for the point-in-time copy software to successfully perform an enable, copy, update, or reset operation on a remote mirror volume. If not, the point-in-time copy operation fails and the remote mirror software reports that the operation is denied.

How the Delay Units Function Affects Volume Copy Operations

During an enable, copy, or update, a background process is started to synchronize the contents of the master and the shadow volumes. Dependent shadow volume sets do not require this background process, see "Dependent Copy Operation" on page 9. This background processing is driven by the bitmap, and traverses from the start of the bitmap to the end, performing the I/O operations and bitmap processing to clear every set bit.

This background synchronization is done in a loop, and is mediated by two variables: the *units* and the *delay* variables. Copy units are expressed in kilobytes, delay units in milliseconds. The loop performs copy units worth of I/O, then sleeps for copy delay milliseconds, until it completes the synchronization.

By adjusting the copy units and copy delay values, a system administrator can tune the impact that background synchronization imposes on the system. The default setting for all II volume sets is set in the file `/usr/kernel/drv/ii.conf`. Once a shadow volume set has been enabled, the system administrator can tune individual or grouped shadow volume sets as desired.

See "To Set Copy Parameters" on page 97.

Export, Import, and Join of Dual-Ported Shadow Volumes

An independent shadow volume that resides on a dual-ported storage array that is attached to two hosts can be utilized by both hosts using the export, *import*, and *join* facilities.

The export, import, and join facilities allow point-in-time copy information to be retained across the entire process of moving the shadow volume from the original host to the partner host, and back again. The independent shadow volume can be exported from the original host, imported by a second host, and later rejoined to its original shadow volume set with no loss of continuity.

The export command removes an independent shadow volume from its shadow volume set, leaving the master volume and the bitmap volume in place to track changes to the master volume. Any attempt to process I/O to the shadow volume during this process will fail, since it is exported and is no longer an active member of its original shadow volume set.

The import command enables a new shadow volume set on the importing host which includes the exported shadow volume as its shadow volume, and a new bitmap volume on the partner host. While enabled, any writes from the partner host are tracked on the bitmap volume. After partner host processing is complete, the shadow volume set is disabled and the exported shadow volume with its new bitmap volume must be made available for the join command on the original host.

The join command reassociates the exported shadow volume with its original shadow volume set by OR-ing the contents of the bitmap volume from the partner host with the contents of the original bitmap. If no writes occurred to the shadow volume while it was on the secondary host, the bitmap will contain only “0”s, and this OR-ing leaves the bitmap on the original host unchanged. The bitmap volume from the partner host is no longer needed after the join command completes, and may be reused.

If a write did occur on the partner host, the OR-ing of a “1” in the partner bitmap with the value in the bitmap on the original host, sets the bitmap for the associated block to “1” (or changed). Using the OR-ed bitmap volume it is now possible to create a point-in-time copy using the update copy.

See “Exporting, Importing, and Joining Shadows” on page 62 for details.

Grouping of Volume Sets

The point-in-time copy software allows shadow volume sets to be placed into groups, which are called I/O groups.

Groups are helpful in administering many volume sets in much the same way that a script would be (i.e. to perform the same operations on many volume sets). With an I/O group, a single CLI command can be issued, and the result will be that the command is executed on every member of the group in turn.

I/O groups allow shadow volume sets to be controlled as a single unit for point-in-time copy or update operations. This facility is especially useful for making self consistent point-in-time copies among a group of shadow volume sets. Group point-in-time copy or update operations are atomic, which means that an operation performed on a group is guaranteed to occur on every volume set in the group or to fail on all the volume sets if it fails on a single volume set in the group.

I/O groups can be specified for the update, full volume copy, wait, list, display, abort, reset, disable, and export operations. I/O groups can be used to create self consistent point-in-time copies among a group of master volumes. This is often required by DBMS which span multiple volumes.

Data Services Logging File

The Sun StorEdge Availability Suite 3.1 point-in-time copy software, like all data services software, generates entries in the data services log file:

`/var/opt/SUNWesm/ds.log`.

This file serves as a running history of commands that have been executed, and includes any related warnings or error messages. This file is maintained by default.

You can rename the file if you want to keep dated versions of it, or you can delete the file if it becomes too large. In either case, a new logging file is automatically created by the software.

A sample section of a logging file containing point-in-time copy software messages is shown below.

```
Feb 06 16:09:49 scm: scmadm cache enable succeeded
Feb 06 16:09:50 ii: iiboot resume cluster tag <none>
Feb 06 16:15:16 sv: enabled /dev/vx/rdsk/rootdg/ii_10mb_0
Feb 06 16:15:16 ii: Enabled /dev/vx/rdsk/rootdg/ii_10mb_0
/dev/vx/rdsk/rootdg/ii_1mb_0 /dev/vx/rdsk/rootdg/ii_mb_0 (dependent)
Feb 06 16:15:17 sv: enabled /dev/vx/rdsk/rootdg/ii_1mb_0
Feb 07 08:14:43 ii: Disabled /dev/vx/rdsk/rootdg/ii_1mb_0
Feb 07 08:15:05 sv: enabled /dev/vx/rdsk/rootdg/ii_10mb_0
Feb 07 08:15:05 ii: Enabled /dev/vx/rdsk/rootdg/ii_10mb_0
/dev/vx/rdsk/rootdg/ii_1mb_0 /dev/vx/rdsk/rootdg/ii_mb_0 (dependent)
Feb 07 08:15:05 sv: enabled /dev/vx/rdsk/rootdg/ii_1mb_0
Feb 07 08:15:19 ii: Create overflow succeeded /dev/vx/rdsk/rootdg/ii_9mb_0
Feb 07 08:15:28 ii: Attach /dev/vx/rdsk/rootdg/ii_1mb_0
/dev/vx/rdsk/rootdg/ii_9mb_0
Feb 07 08:19:59 ii: Start update /dev/vx/rdsk/rootdg/ii_1mb_0 to shadow
Feb 07 08:20:02 ii: Finish update /dev/vx/rdsk/rootdg/ii_1mb_0 to shadow
Feb 07 08:21:21 ii: Disabled /dev/vx/rdsk/rootdg/ii_1mb_0
Feb 07 08:21:27 sv: enabled /dev/vx/rdsk/rootdg/ii_10mb_0
Feb 07 08:21:27 ii: Enabled /dev/vx/rdsk/rootdg/ii_10mb_0
/dev/vx/rdsk/rootdg/ii_1mb_0 /dev/vx/rdsk/rootdg/ii_mb_0 (dependent)
Feb 07 08:21:27 sv: enabled /dev/vx/rdsk/rootdg/ii_1mb_0
Feb 07 08:21:38 ii: Attach /dev/vx/rdsk/rootdg/ii_1mb_0
/dev/vx/rdsk/rootdg/ii_9mb_0
Feb 07 08:22:42 ii: Disabled /dev/vx/rdsk/rootdg/ii_1mb_0
```

Operational Notes

Cautions: Enable, Copy, and Update

- The point-in-time copy software does not support the use of files for bitmaps. You must use a volume for a bitmap.
- Do not perform enable, copy, or update operations on mounted shadow volumes. If mounted, you must unmount the shadow volume before invoking an enable, copy, or update operation to prevent a file system panic from occurring.

  Either unmount or quiesce all applications using the master volume for only the instant when the point-in-time snapshot is enabled.

  Immediately after the point-in-time is (re-)established, (either when the CLI prompt returns or the next shell script command is read), the master volume can be remounted or the applications using them can be resumed. Also the shadow volume can be mounted and immediately accessed.

Using the CLI for Copy and Update Operations

Always specify the shadow volume name of the volume group when using the copy or update commands.

Length of Volume Names

Master, shadow, and bitmap volume names (absolute path name) are currently limited to a maximum of 64 characters consisting of any legal characters that can be part of a filename.

Note – As of Sun StorEdge Availability Suite 3.1 point-in-time copy software, only path names that begin with `/dev/` are accepted.

Shadowing the Root File System

You cannot make a shadow volume copy of the root device `/` or `/usr`.

Shadowing Encapsulated Volumes

The point-in-time copy software does not support encapsulated volumes; you cannot create a shadow volume of an encapsulated volume.

Interaction With svadm

Using the command option `iiadm -e` to enable a volume set automatically adds the volumes to the `sv` layer and using the `iiadm -d` command option to disable a volume set automatically remove volumes from the `sv` layer.

There is no checking in the `sv` layer to prevent you from deleting volumes with `svadm` that are actively being used by the point-in-time copy software or other data services. If you remove volumes from the `sv` layer that are still in use by point-in-time copy software or remote mirror software, you will be able to continue operations on these volumes with no error messages, but the data in the volume set will become inconsistent.

reboot Command

Always use the `shutdown` command, not the `reboot` command.

Note – Do not use the `reboot` command. Always use the `/etc/shutdown` command. The `/etc/shutdown` command also ensures that any shutdown scripts in the `/etc/init.d` directory are executed.

Creating and Configuring Sun StorEdge Volume Sets

Caution – Only one system administrator or root user at a time should create and configure Sun StorEdge point-in-time copy shadow volume sets. This restriction helps avoid causing the Sun StorEdge configuration becoming inconsistent.

Two administrators should not be writing to the Sun StorEdge services configuration at the same time. The operations that access the configuration include, but are not limited to:

- Creating and deleting shadow volume sets
- Adding and removing shadow volume sets from I/O groups
- Assigning new bitmap volumes to a shadow volume set
- Updating the disk device group or resource name
- Any operation that changes the Sun StorEdge services and related shadow volume set configuration

CHAPTER 3

Using Point-in-Time Copy Software

This chapter contains examples showing how to configure, enable, and use the point-in-time copy software. The examples in this chapter show how to perform operations using the command-line interface (CLI). This chapter includes:

Point-in-Time Copy Operations

This section discusses the operation of the point-in-time copy software from a system administrator's viewpoint. Typical tasks are described in detail with examples. The example use these volume names:

Volume Name	Definition
`/dev/rdsk/c1t3d0s0`	The name of the master volume
`/dev/rdsk/c1t3d0s4`	The name of the shadow volume
`/dev/rdsk/c1t2d0s5`	The name of the bitmap volume
io-groupname	The name of the I/O group
`/dev/rdsk/c1t4d0s6`	The name of the overflow volume

All commands are accessed on the following path:

`/usr/opt/SUNWesm/sbin/iiadm`

Point-in-time copy software operations include but are not limited to:

1. Enabling shadow volume sets for selected volumes.
2. Using fast resynchronization, or update, commands to quickly update volumes when necessary.
3. Using the copy command to perform full volume copies.

   You typically perform a full volume copy only when you need two physically-separate volumes, or if a volume has become corrupted or has failed. Copying from master to shadow overwrites the shadow volume with the master contents; copying from shadow to master overwrites the master volume with the changed shadow contents.
4. Disabling shadow volume sets, if you choose to do so, but you can keep shadow volume sets enabled indefinitely.

Note – A remote mirror volume set must be in logging mode for the point-in-time copy software to successfully perform an enable, copy, update, or reset operation on a remote mirror volume. If not, the point-in-time copy operation fails and the remote mirror software reports that the operation is denied.

Increasing the Storage Volume Limit

The Sun StorEdge Availability Suite 3.1 software has a default limit of 1024 storage volumes for use with the software. For example, if you use point-in-time copy software only, you can have 341 volume sets, each consisting of master, shadow, and bitmap volumes. However, if you also use the remote mirror software along with point-in-time copy software, the volumes are divided between the two packages.

If you want to increase the storage volume limit, edit `/kernel/drv/nsctl.conf` and specifically the number associated with `nsc_max_devices`. Doing so will cause more memory to be consumed and could cause you to have to adjust the value in `nsc_global_pages` value in `mc_rms.conf`. Only an experienced system administrator should make these changes.

The following procedure describes how to increase this default limit.

▼ To Increase the Storage Volume Limit

Caution – Increasing this limit causes more memory to be consumed. You might have to adjust the `nsc_global_pages` value in the `/kernel/drv/mc_rms.conf` file. Only an experienced system administrator should make these changes.

---

1. **Log on as the root user.**
2. **Open the** `/kernel/drv/nsctl.conf` **file using a text editor such as** `vi`**(1) or** `ed`**(1).**
3. **Search for the** `nsc_max_devices` **field.**
4. **Edit the number in this field to increase your volume limit.**
5. **Save and exit the file.**
6. **Reboot your server as follows:**

```
# /etc/shutdown -y -g 0 -i 6
```

Sizing Requirements for Volume Sets

Before you can enable shadow volume sets for use by the point-in-time copy software, you must create the volumes with your volume manager. The point-in-time copy software supports VERITAS Volume Manager, Sun Solstice Disk Suite™, and raw volumes.

Shadow volume sets can be organized in I/O groups so that commands can be carried out against multiple shadow volume sets with one command; the number of I/O groups is limited only by the amount of storage you have physically available.

- A master volume can be any size.
- A shadow volume, independent or dependent, must be the same size (or greater) than the master volume size. However, to prevent reverse sync problems, the shadow should be no larger than the master.
- A compact dependent shadow volume can be any size, but should easily exceed the total number of expected writes to both the master and shadow volumes during the time the compact dependent shadow volume is expected to be in use. The point-in-time copy software supports a shared overflow volume, which can store excess changes, over the configured size of the compact dependent shadow volume. If the overflow volume is full, or not specified, then a change can't be recorded in the compact dependent shadow volume, forcing it to become inconsistent and subsequently taken off-line.
- A bitmap volume's size is based on the size of the master volume and the type of volume set being created (independent, dependent, or compact dependent).
  - For independent or dependent shadow volume sets:

    8 KB per 1 GB of master volume size (rounded-up to the nearest whole GB), plus an additional 24 KB for overhead.

    For example, to shadow a 3 GB master volume, the bitmap size must be (3 × 8 KB) + 24 KB, or 48 KB in size. A 50 GB master volume requires a 424 KB bitmap volume.
  - For compact dependent shadow volume sets:

    264 KB per 1 GB of master volume size (rounded up to the nearest whole GB), plus an additional 24 KB for overhead.

    For example, to shadow a 3 GB master volume, the bitmap size must be (3 × 264 KB + 24 KB, or 816 KB in size. A 50 GB master volume in a compact dependent shadow volume set requires a 13224 KB bitmap volume.

If you enable a shadow volume set with a bitmap that is too large, the shadow volume set is created even though space may be wasted. If you enable a shadow volume set with a bitmap that is too small, the enable command fails with an error message.

Enabling Volume Sets

Note – Use a volume for the bitmap. File-based bitmaps are not supported.

Note – Do not create bitmaps on cylinder 0 because the point-in-time copy software does a raw write and destroys the VTOC for that device.

Note – A remote mirror volume set must be in logging mode for the point-in-time copy software to successfully perform an enable, copy, update, or reset operation on a remote mirror volume. If not, the point-in-time copy operation fails and the remote mirror software reports that the operation is denied.

Note – The remote mirror and point-in-time copy software does not support metatrans devices (also known as trans metadevices) created by the Sun Solstice DiskSuite and Solaris Volume Manager software.

▼ To Enable an Independent Volume Set

Use the `-e ind` option to enable an independent shadow volume set. A full volume copy of the master volume to the shadow volume will be performed. Both master volume and shadow volume will be available at once, but the shadow volume will be treated as a dependent shadow until the full copy is complete. See also "To Enable Volume Sets" on page 86.

1. **Unmount the shadow volume** `/dev/dsk/c1t3d0s4`**.**
2. **Enable an independent shadow volume set by entering:**

```
# iiadm -e ind /dev/rdsk/c1t3d0s0 /dev/rdsk/c1t3d0s4 /dev/rdsk/c1t2d0s5
(to enable an independent shadow volume set)
# iiadm -g io-groupname -e ind /dev/rdsk/c1t3d0s0 /dev/rdsk/c1t3d0s4 \
/dev/rdsk/c1t2d0s5
(to enable an independent shadow volume set in an I/O group)
```

3. **Mount the shadow volume,** `/dev/dsk/c1t3d0s4`**.**

   You now have an independent shadow volume set. If the shadow volume set is enabled in a non-existent I/O group, the I/O group will be created.

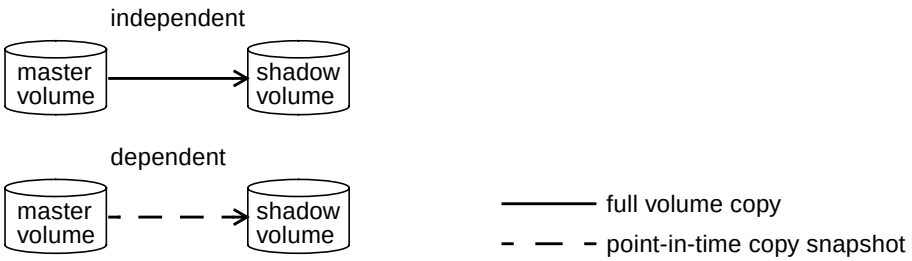

FIGURE 3-1 Enabling Shadow Volume Sets

▼ To Enable a Dependent Volume Set

Use the `-e dep` option to enable a dependent shadow volume set. No copy will take place and the volume set will be available at once. See also "To Enable Volume Sets" on page 86.

1. **Unmount the shadow volume** `/dev/dsk/c1t3d0s4`**.**
2. **Enable a dependent volume set by entering:**

```
# iiadm -e dep /dev/rdsk/c1t3d0s0 /dev/rdsk/c1t3d0s4 /dev/rdsk/c1t2d0s5
(to enable a dependent volume set)
# iiadm -g io-groupname -e dep /dev/rdsk/c1t3d0s0 /dev/rdsk/c1t3d0s4 \
/dev/rdsk/c1t2d0s5
(to enable a dependent volume set in an I/O group)
```

3. **Mount the shadow volume** `/dev/dsk/c1t3d0s4`**.**

   You now have a master volume-dependent point-in-time snapshot of the data. If the volume set is enabled in a non-existent I/O group, the I/O group will be created.

Using Overflow Volumes

You can initialize an overflow volume for use with compact dependent shadow volumes, then attach the overflow volume to selected compact dependent shadow volume sets. You can also detach an overflow volume from a compact dependent shadow volume set.

▼ To Initialize an Overflow Volume

Before you can attach an overflow volume to a shadow volume set, the overflow volume must be initialized. See also "To Initialize an Overflow Volume" on page 96.

- **Initialize an overflow volume by entering:**

```
# iiadm -O /dev/rdsk/c1t4d0s6
```

You can now attach shadow volume sets with compact dependent shadow volumes to the newly initialized overflow volume. Any number of compact dependent shadows can be attached to an overflow volume as long as it has sufficient space.

This option supports the optional `-n` parameter, no prompting, which will directly perform the requested action without prompting. This option is typically used during script processing.

Be sure that you want to initialize the overflow volume, especially if you are using the `-n` option.

Note – When using the `-n` option, the `-n` must precede the `-O` option on the command line. For example: `-nO` or `-n -O`. See "-n Option to the -c, -A, -O, and -u Command Options" on page 77.

▼ To Attach Overflow Volumes to Shadow Volume Sets

A dependent shadow volume set with a compact dependent shadow volume should have an overflow volume attached to it so that any writes to the volume set, if it gets full, can be redirected to the overflow volume. See also "To Attach an Overflow Volume" on page 82.

- **Attach an overflow volume to a volume set by entering:**

```
# iiadm -A /dev/rdsk/c1t4d0s6 /dev/rdsk/c1t3d0s4
```

- **Attach an overflow volume to all compact dependent shadow volume sets in an I/O group by entering:**

```
# iiadm -g io-groupname -A /dev/rdsk/c1t4d0s6
```

This option supports the optional `-n` parameter, no prompting, which will directly perform the requested action without prompting. This option is typically used during script processing.

If the volume you are attaching has not yet been initialized (see "To Initialize an Overflow Volume" on page 96), it will be before it is attached so be certain that you want to initialize the volume, especially when using the `-n` option.

Note – When using the `-n` option, the `-n` must precede the `-A` option on the command line. For example: `-nA` or `-n -A`.

▼ To Detach Overflow Volumes from Shadow Volume Sets

Use this option if you want to detach an overflow volume from a compact dependent shadow volume set. See also "To Detach an Overflow Volume" on page 85.

- **Detach an overflow volume from a shadow volume set by entering:**

```
# iiadm -D /dev/rdsk/c1t3d0s4
```

- **Detach an overflow volume from all compact dependent shadow volume sets in an I/O group by entering:**

```
# iiadm -g io-groupname -D
```

If the overflow volume contains any data associated with any shadow volume set that you are trying to detach, the command will fail.

Moving Shadow Volume Sets

Shadow volume sets can be moved into and out of I/O groups for ease of management. All shadow volume sets within an I/O group respond to commands sent to the I/O group.

▼ To Move Shadow Volume Sets into I/O Groups

Use the `-m` option to move shadow volume sets into I/O groups for ease of management. Move a shadow volume set out of any I/O group by entering two double-quote marks instead of the I/O group name. See also "To Move a Volume Set" on page 95. If you move the shadow volume set into a non-existent I/O group, the point-in-time copy software will create that I/O group.

- **Move the shadow volume set into an I/O group by entering:**

```
# iiadm -g io-groupname -m /dev/rdsk/c1t3d0s4
```

- **Move the shadow volume set out of an I/O group by entering:**

```
# iiadm -g "" -m /dev/rdsk/c1t3d0s4
```

Setting Copy Parameters

Volume copy can be *throttled* by setting the copy parameters so that copy operations do not adversely affect ongoing business operations by consuming system resources.

▼ To Get and Set Copy Parameters

Use the `-P` option to set copy parameters after enabling your shadow volume sets and collecting them in I/O groups for ease of management. When an update or a copy is in progress, two settings determine how much impact the copy or update has on availability of system resources to other applications.

The first setting, *units*, is the number of chunks of data to copy (100 through 60000) before delaying to allow access to other applications.

The second setting, *delay*, is the length of time, in system clock ticks (2 through 10000), to wait before sending the next series of data chunks.

Get the current settings of these parameters with the `-P` option. The point-in-time copy software defaults to minimum settings. See also "To Set Copy Parameters" on page 97.

- **Set copy parameters for a specific volume set by entering:**

```
# iiadm -P delay units /dev/rdsk/c1t3d0s4
```

- **Set copy parameters for all volume sets in an I/O group by entering:**

```
# iiadm -g io-groupname -P delay units
```

- **Get copy parameters for a specific volume set by entering:**

```
# iiadm -P /dev/rdsk/c1t3d0s4
```

- **Get copy parameters for all volume sets in an I/O group by entering:**

```
# iiadm -g io-groupname -P
```

Updating, Copying, and Waiting

To bring the master volume and the shadow volume into synchronization, you can perform either an update or a copy. An update writes only the blocks of data that have changed since the last update or copy as recorded in the bitmap. A copy operation performs a full volume copy, which takes longer than an update. You can update the master volume from the shadow volume or the shadow volume from the master volume. If necessary, you can cause the point-in-time copy software to wait for a copy or update to complete before executing the next command.

The update, copy, and wait command options also support process ID locking, or PID locking, see "PID Locking a Shadow Volume Set" on page 51.

Note – Take the same precautions using copy and update options as you would using the dd(1M) command to copy between raw devices. That is, unmount the target of the copy and update operation first. You can unmount the source device also, to stop write operations to it and ensure a consistent shadow copy, but this source unmount is not required. After issuing the copy or update, remount the device(s) and resume the workload; you do not have to wait for the operation to complete.

Note – A remote mirror volume set must be in logging mode for the point-in-time copy software to successfully perform an enable, copy, update, or reset operation on a remote mirror volume. If not, the point-in-time copy operation fails and the remote mirror software reports that the operation is denied.

Note – If you do not use the -n option to the update command, you are prompted to confirm updates for any update from a shadow volume to its master volume. See "-n Option to the -c, -A, -O, and -u Command Options" on page 77.

▼ To Update Shadow Volume Sets

Use the `-u` option to perform fast resynchronizations, or updates, of a shadow volume set at much faster speeds than most full volume copy or mirroring operations. See also "To Update a Volume Set" on page 102.

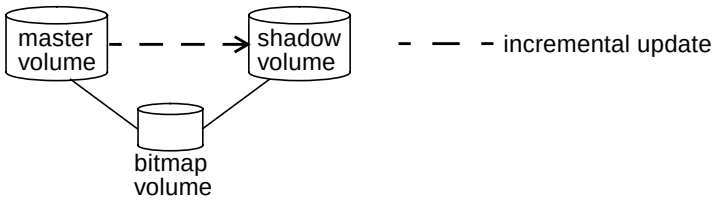

FIGURE 3-2 Updating the Shadow Volume

Update the Shadow Volume From the Master Volume

1. **Unmount the shadow volume** `/dev/rdsk/c1t3d0s4.`
2. **Update the shadow volume by entering:**

```
# iiadm -u s /dev/rdsk/c1t3d0s4
(update the shadow from the master in the specified volume set)
# iiadm -g io-groupname -u s
(update all volume group shadows associated with the specified I/O group)
```

3. **Mount the shadow volume.**

Update the Master Volume From the Shadow Volume

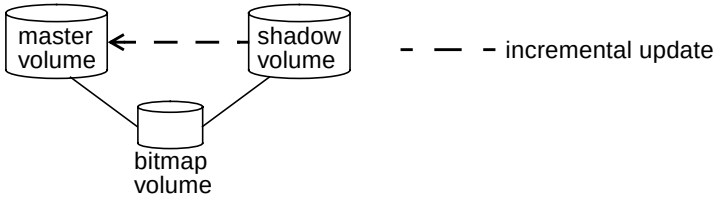

FIGURE 3-3 Updating the Master Volume

1. **Unmount the master volume** `/dev/rdsk/c1t3d0s0.`

2. **Update the master volume by entering**:

Note – The specified volume in this example is the shadow volume, even though you are updating the master volume. This is correct; always specify the set's shadow volume when performing on a shadow volume set.

```
# iiadm -u m /dev/rdsk/c1t3d0s4
(update the master from the shadow in the specified volume set)
# iiadm -g io-groupname -u m
(update all volume group masters from their shadows in the specified I/O group)
```

3. **Mount the master volume.**

▼ To Perform Full Copy Operations

You typically perform a full volume copy only when you need two physically-separate volumes, or if a volume has become corrupted, or if a volume has failed. See also "To Copy Volumes" on page 83.

Copy All Data From the Master Volume to the Shadow Volume

1. **Unmount the shadow volume** `/dev/rdsk/c1t3d0s4`**.**
2. **Copy all data from the master volume to the shadow volume by entering:**

```
# iiadm -c s /dev/rdsk/c1t3d0s4
(copy all data from master volume to shadow volume in the specified volume set)
# iiadm -g io-groupname -c s
(copy all data from all masters in an I/O group to their shadows)
```

3. **Mount the shadow volume.**

FIGURE 3-4 Copy Master Volume to Shadow Volume

Copy All Data From the Shadow Volume to the Master Volume

FIGURE 3-5 Copy Shadow Volume to Master Volume

1. **Unmount the master volume** `/dev/rdsk/c1t3d0s0`**.**
2. **Copy all data from the shadow volume to the master by entering:**

```
# iiadm -c m /dev/rdsk/c1t3d0s4
(copy all data from shadow volume to master volume in the specified volume set)
# iiadm -g io-groupname -c m
(copy all data from all shadows in an I/O group to their masters)
```

3. **Mount the master volume.**

▼ To Wait for Completion of Copy or Update

When you start a copy or an update and you want to issue another command involving shadow volume sets that are part of the copy or update, issue a wait command then your next command. The point-in-time copy software will wait until the copy or update is complete before proceeding with the next command. See also "To Wait for a Copy or Update to Complete" on page 104.

The wait command option also supports the unlock PID option, `-n`. If the locking process has terminated without unlocking a shadow volume set, you can unlock it with the `-n` option to the wait command option. See "PID Locking a Shadow Volume Set" on page 51.

- **Wait for a copy or update to a specific shadow volume set to complete by entering:**

```
# iiadm -w /dev/rdsk/c1t3d0s4
```

- **Wait for copies or updates to all volume sets in an I/O group to complete by entering:**

```
# iiadm -g io-groupname -w
```

Waiting a shadow volume set monitors *all* processes executing on the shadow volume set when the wait is issued and returns when all of the processes have completed.

PID Locking a Shadow Volume Set

When you execute a copy or update command on a shadow volume set and you include the `-p` option, you lock that shadow volume set so that no other process can affect it. Executing the wait command option, `-w`, with the `-p` command option against a shadow volume set clears the PID lock if and only if your process is the locking process.

Locking a Shadow Volume Set

For example:

- This command line updates a shadow volume from its master volume and PID locks the shadow volume set to the process that issues this command line:

```
# iiadm -p -u s /dev/rdsk/c1t3d0s4
```

- This command line copies a shadow volume and PID locks the shadow volume set to the process that issues this command line:

```
# iiadm -p -c s /dev/rdsk/c1t3d0s4
```

When you attempt to PID lock a shadow volume set, three things can happen:

If the shadow volume set is not currently PID locked, the operation will conclude successfully.

If the shadow volume set is currently PID locked and you are the original locking process, the operation will conclude successfully, although no change will have taken place.

If the shadow volume set is currently locked and you are not the original locking process, the operation will fail and an error message will be issued.

Unlocking a Shadow Volume Set

There are two circumstances in which you perform commands to unlock a PID lock: when you are finished with the shadow volume set and are no longer concerned that it might change while a command is executing, or when a process has terminated without releasing a PID lock.

For example, to unlock a shadow volume set that your process locked:

- This command line waits for completion of all operations against a shadow volume set and unlocks the PID lock of the volume set. You must be the originating process in order for this command to execute successfully.

```
# iiadm -p -w /dev/rdsk/c1t3d0s4
```

In the event that a process is terminated before it unlocks a PID lock on a volume set, the wait command with the `-n` option can be used to unlock it.

For example, to unlock a shadow volume set whose locking process is defunct:

- This command line waits until all commands against a shadow volume set have terminated, then unconditionally unlocks the PID lock. You do not have to be the originating process in order for this command to execute successfully.

```
# iiadm -n -p -w /dev/rdsk/c1t3d0s4
```

Note – The `-n` option to the wait command, `-w`, has no inherent security and can be used, by root, even if an initiating process is still current. Be cautious in your use of this command option.

Listing Volumes

These command options enable you to list various shadow volume sets and volumes as required.

▼ To List Shadow Volume Sets

You can use the `-l` option to list all configured shadow volume sets, including suspended and offlined sets, or all such shadow volume sets in a specific I/O group. See also "To List Volume Sets" on page 93.

- **List all configured shadow volume sets by entering:**

```
# iiadm -l
```

- **List all configured shadow volume sets in an I/O group by entering:**

```
# iiadm -g io-groupname -l
```

Note – Entering `iiadm` with no options lists the same information as entering `iiadm -l`.

▼ To List Overflow Volumes

Use the `-L` option to list all overflow volumes. See also "To List Overflow Volumes or Groups" on page 94.

- **List all overflow volumes by entering:**

```
# iiadm -L
```

▼ To List I/O Groups

Use the `-L` option to list all I/O groups. See also "To List Overflow Volumes or Groups" on page 94.

- **List all I/O groups by entering:**

```
# iiadm -g -L
```

Displaying Status

You can display the status of shadow volume sets and of overflow volumes. Displaying status enables you to see the progress of copy and update operations and the status of overflow volumes.

▼ To Display Shadow Volume Set Status

Use the `-i` option to check the status of a shadow volume set or of all shadow volume sets in an I/O group. Use this option after issuing a copy or an update command to see if the copy or update has finished execution. Also see "To Display Volume Set Status" on page 90.

- **Display the status of a specific shadow volume set by entering:**

```
# iiadm -i /dev/rdsk/c1t3d0s4
```

- **Display the status of all shadow volume sets in an I/O group by entering:**

```
# iiadm -g io-groupname -i
```

▼ To Display Overflow Volume Status

Use the `-Q` option to display the status of overflow volumes, the total overflow volume capacity, how much capacity is in use, and how much capacity is available. See also "To List Overflow Volumes or Groups" on page 94.

- **Display the status of a specific overflow volume by entering:**

```
# iiadm -Q /dev/rdsk/c1t4d0s6
```

Note – If the overflow volume is not attached to any compact dependent shadow volumes and you use the `iiadm -Q` option, you will get an error message.

Aborting Copy or Update Commands

Update operations and copy operations can consume system bandwidth. You can abort these operations during busy times and continue them later.

▼ To Abort a Copy or an Update

Use the `-a` option to abort a copy or update that is in process. The state of the copy or update will be saved so that you can complete the operation at a later time. See also "To Abort a Copy or Update" on page 81.

- **Abort the copy or update to a specific shadow volume set by entering:**

```
# iiadm -a /dev/rdsk/c1t3d0s4
```

- **Abort all copy or update operations to all shadow volume sets in an I/O group by entering:**

```
# iiadm -g io-groupname -a
```

Reissue the copy or update command for the affected volume set to continue.

Note – The master and shadow volumes of an independent volume set are complete, consistent volumes that can be used independently (hence the name), after you disable the volume set. Thus prior to allowing a disable operation to complete, both the master and shadow volumes must be left in a consistent state.

If a copy or update operation is in progress, you will be unable to disable the volume set using the `iiadm -d` command option. To proceed, you must abort the copy or update operation using the `iiadm -a` command option, the volumes will be left inconsistent, and a disable command option would now be allowed.

Aborting a shadow volume set causes all copy or resynchronization activity to stop. A point-in-time update is required to resume internal processing.

Resetting

Shadow volume sets that are offline for any reason can be put back on line with the reset command.

Note – If a bitmap volume goes offline, the *delay* and *units* parameters are reset to the default values when the volume is reset with the `iiadm -R` command.

▼ To Reset Volume Sets

Note – A remote mirror volume set must be in logging mode for the point-in-time copy software to successfully perform an enable, copy, update, or reset operation on a remote mirror volume. If not, the point-in-time copy operation fails and the remote mirror software reports that the operation is denied.

An offlined shadow volume set can be brought back online with the `-R` option after the reason for the offline status has been cleared. Note that the volume set, while still associated, is not considered to contain valid shadow data. See also "To Reset a Volume Set or I/O Volume Group" on page 100.

- **Reset an offlined shadow volume set by entering:**

```
# iiadm -R /dev/rdsk/c1t3d0s4
```

- **Reset all offlined shadow volume sets in an I/O group by entering:**

```
# iiadm -g io-groupname -R
```

Caution – The `iiadm -g group -R` command *must not be used* with multiple shadows of single masters because all the resets occur at the same time with unpredictable results.

If the master volume of more than one shadow volume (multiple shadows) goes offline and, upon repair, must be returned online with the `iiadm -R` command, follow this procedure:

1. **Reset all independent shadow volume sets, except the shadow volume set that you want to update the master volume with.**

```
# iiadm -R shadow  shadow2 ... shadowN
```

2. **Wait for all these resets to finish with the `-w` command.**

```
# iiadm -w shadow  shadow2 ... shadowN
```

3. **Reset the final shadow volume set that contains the shadow volume you want to update the master volume with.**

```
# iiadm -R shadow
```

4. **Wait for the copy to complete.**

```
# iiadm -w shadow
```

5. **Reset all dependent shadow volume sets..**

```
# iiadm -R shadow  shadow2 ... shadowN
```

TABLE 3-1 Reset Command Results for Various Volume Types

Offline volume type	After issuing a reset, the point-in-time copy software:
Dependent shadow volume	Automatically updates the shadow volume from the master volume, using volume differences stored by the bitmap volume
Independent shadow volume	Automatically copies the master volume to the shadow volume
Bitmap	Resynchronizes the volume pair by copying the master volume to the shadow volume

Offlining a volume in a shadow volume set causes any I/O attempted on that volume to fail. The point-in-time copy driver intervenes, disallowing access, for example, to a shadow volume, and preventing writes and bitmap changes on a master.

Resetting the volume after it has been offlined reestablishes I/O to the previously offlined volume of the shadow volume set. If the master volume had been offlined, for example, bitmap management restarts on the previously established bitmap.

Resetting volumes is usually required when storage array maintenance is required: for example, when a disk on which part of a shadow volume set resides needs replacing.

Disabling

Disabling a shadow volume set ends the association of the volumes within the shadow volume set. The association of the data in the shadow volume set cannot be reestablished except by enabling the shadow volume set again, which causes a full copy of the master to the shadow in an independent shadow volume set, or a dependent shadow volume set. In either case, the shadow volume set is new and the data status at the time of disable is lost. Disable a volume set only if you are finished using it as a shadow volume set.

▼ To Disable a Shadow Volume Set

See also "To Disable Volume Sets" on page 84 and "Interaction With svadm" on page 33.

- **Disable the point-in-time copy software for a volume set by entering:**

```
# iiadm -d /dev/rdsk/c1t3d0s4
```

- **Disable all volume sets in an I/O group by entering:**

```
# iiadm -g io-groupname -d
```

▼ To Disable an Independent Shadow Volume Set During a Copy or Update

The point-in-time copy software requires that the master and shadow volumes of an independent shadow volume set be in a consistent state before the shadow volume set is disabled. Therefore, the point-in-time copy software will not allow you to disable an independent shadow volume set while a copy or update operation is in progress because the shadow and master volumes would be inconsistent.

Any volume that is a dependent shadow volume must be unmounted before attempting a disable operation. This includes an independent shadow volume that is dependent due to an incomplete copy or update operation.

If you must disable an independent volume set while a copy or update is in progress, *and you are not concerned with the consistency of the target volume*:

1. **Unmount the shadow volume set.**
2. **Abort the copy or update.**

   See "To Abort a Copy or an Update" on page 56.

```
# iiadm -a /dev/rdsk/c1t3d0s4
```

3. **Disable the shadow volume set with the `iiadm -d` command option.**

```
# iiadm -d /dev/rdsk/c1t3d0s4
```

The shadow volume set is now disabled.

With a shadow volume set configured with a dependent shadow volume, any I/O being processed to the shadow volume is halted, and the volume set is disabled.

Note – The target (master or shadow) volume of the copy or update operation is left in an inconsistent state.

Exporting, Importing, and Joining Shadows

The functionality implemented via the three `iiadm` command options, `-E` (export), `-I` (import) and `-J` (join), allow for a shadow volume on a dual-ported device to be exported to a secondary host while still under point-in-time copy software control. This capability allows shadow volume processing by its associated applications to be off-loaded to a secondary host without impacting the primary host's master volume or its associated applications.

Retaining control of the shadow volume while imported on the secondary host maintains bitmap integrity to facilitate fast resynchronization via copy or update processing at a later time.

While the shadow volume is exported to a secondary host, read and write access to the master volume is tracked by the point-in-time copy software. Read and write access to the shadow volume on the secondary host is tracked by point-in-time copy using a second bitmap.

Once secondary host processing has completed, the shadow volume and second bitmap can be disabled on the secondary host and joined to the original master volume such that the master, shadow, and bitmap consistency is reconstructed. After completing the join processing, the shadow volume set is now in the same state as it would have been if the secondary host processing of the shadow volume had been performed by the primary host.

Note – You might not need to copy the bitmap to the original host if the bitmap is on a dual-ported drive.

Note – The bitmap from Host B that was copied across to be joined can be removed from Host A once the join command is complete.

Caution – Once a shadow is rejoined to its original master on Host A, Host B should not use the shadow volume even if it is still accessible.

▼ To Export, Import, and Join a Shadow Volume

The following is a tabular outline that describes how to export, import, and join a shadow volume.

See also "To Export a Shadow Volume" on page 88.

See also "To Import a Shadow Volume" on page 91.

See also "To Join a Shadow Volume" on page 92.

- **Create Instant Image shadow volume on primary node and start application using shadow volume.**

Primary Host	Secondary Host	Comments
master volume		existing *master* volume, mounted, valid data
create *shadow* and *bitmap1* volumes		should be the same redundancy (RAID) as the *master*
shadow volume		same size as *master* on a dual-ported device
bitmap1 volume		size based on *master* volume set
enable independent shadow set		`iiadm -e ind` *master shadow* \ *bitmap1*
mount *shadow* volume		`mount` *shadow mount_point*
start application using *shadow*		

- **Switch application and shadow volume to secondary host.**

Primary Host	Secondary Host	Comments
stop application using *shadow*		existing *master* volume, mounted, valid data
unmount *shadow*		`umount` *mount_point*
export *shadow* volume		`iiadm -E` *shadow*
create *bitmap2*		same size as *bitmap1*, on dual-port device
copy *bitmap1* to *bitmap2*		`cp` *bitmap1 bitmap2*
deport *shadow/bitmap2*		required for VxVM or SDS(SLVM)
	import *shadow/bitmap2*	required for VxVM or SDS(SLVM)
	import *shadow* volume	`iiadm -I` *shadow bitmap2*
	mount *shadow* volume	`mount` *shadow mount_point*
	start application using *shadow*	

- **Switch application and shadow volume to primary host.**

Primary Host	Secondary Host	Comments
	stop application using *shadow*	
	unmount *shadow* volume	`umount` *mount_point*
	disable *shadow* volume	`iiadm -d` *shadow*
	deport *shadow/bitmap2*	required for VxVM or SDS(SLVM)
import *shadow/bitmap2*		required for VxVM or SDS(SLVM)
join *shadow* volume		`iiadm -J` *shadow bitmap2*
mount *shadow* volume		`mount` *shadow mount_point*
start application using *shadow*		

Displaying Help and Software Versions

The point-in-time copy software can be made to display help information as well as software version numbers of installed modules.

▼ To Display Help Information

Help information is displayed in two ways: in context with an incorrectly entered command line, and in response to the `-h` option. Bad command lines get specific help for the command in question. The `-h` option causes the entire help content to be sent to `stderr`.

- **Display the entire help text by entering:**

```
# iiadm -h
```

▼ To Display Software Version Numbers

The version numbers of installed point-in-time copy software modules is accessible through the `-v` option. See also "To Print Software Versions" on page 103.

- **Display the software version numbers by entering:**

```
# iiadm -v
```

CHAPTER 4

Configuration Considerations

This chapter covers the following topics:

Point-in-Time Software with Remote Mirror Software

Sun StorEdge Availability Suite 3.1 remote mirror software allows the replication, or mirroring, of volumes hosted on a Solaris system across any TCP/IP network. Remote mirror software is used to ensure volume level backups at physically remote locations.

Remote mirror software, like point-in-time copy software, synchronizes volumes as a part of providing mirroring capabilities. After a synchronization has been performed, remote mirror volumes, which are termed the primary and secondary volumes, are remotely replicated. This means that the two volumes are kept up-to-date with each other. Additional information is available: see "Related Documentation" on page xii. A few of the highlights of using point-in-time software and remote mirror software together are described below.

To help ensure the highest level of data integrity and system performance on both sites during normal operations, point-in-time copy software is recommended for use in conjunction with remote mirror software.

Using point-in-time copy software with remote mirror software, a point-in-time copy can be replicated to a physically remote location, providing a consistent copy of the volume as part of an overall disaster recovery plan. Depending upon the configuration of the shadow volume set, such a copy can be kept relatively up-to-date without noticeable impact to normal processing.

For example, a point-in-time copy of a remote mirror primary volume can be transferred to the secondary site. Applications can remain open and active at the primary site while the point-in-time copy is established. This works well if the secondary volume is able to be out of sync with the primary volume by some small time delta. The advantage in this approach is that the overhead involved in remotely mirroring the primary data will be reduced if the point-in-time copy is mirrored instead. Keeping the secondary site slightly out of sync with the primary also allows the verification of the correctness of the primary data prior to replicating it to the secondary site.

Using point-in-time copy software with remote mirror software, a point-in-time copy of a remote mirror secondary volume can be established prior to starting synchronization of a secondary volume from the primary site (the site the primary volume is hosted from). Protection against double failure is provided by using the point-in-time copy software to create a point-in-time copy of the replicated data at the secondary site before beginning resynchronization. If a subsequent failure should occur during resynchronization, the point-in-time copy can be used as a fallback position, and resynchronization can be resumed when the subsequent

failure issues have been resolved. Once the secondary site is fully synchronized with the primary site, the point-in-time copy software volume set can be disabled, or put to other uses, such as remote backup, remote data analysis, or other functions required at the secondary site.

Interaction Between Point-in-Time Copy software and Remote Mirror software in the Sun StorEdge Data Service I/O stack

The point-in-time copy software I/O performed internally during an enable, copy, or update operation can alter the contents of the shadow volume without any new I/O coming down the Solaris I/O stack. When this happens, the I/O is not intercepted in the SV layer. If the shadow volume is also a remote mirror volume, the remote mirror software will not see these I/O operations either. In this situation, the data modified by the I/O will not be replicated to the target remote mirror volume.

To allow this replication to occur, the point-in-time copy software can be configured to offer the remote mirror software the changed bitmap. If the remote mirror software is in logging mode, it will accept the bitmap. If the bitmap is accepted, the remote mirror software will OR the point-in-time copy software bitmap with its own bitmap for that volume, which adds the point-in-time copy software changes to its own list of changes to be replicated to the remote node. If the remote mirror software is in replication mode for the volume, it will reject the bitmap from the point-in-time copy software. This, in turn, will fail the enable, copy, or update operation. Once remote mirror logging has been reenabled, the point-in-time copy software operation can be reissued.

Note – A remote mirror volume set must be in logging mode for the point-in-time copy software to successfully perform an enable, copy, update, or reset operation on a remote mirror volume. If not, the point-in-time copy operation fails and the remote mirror software reports that the operation is denied.

Using A Point-in-Time Copy Shadow Volume Set to Back Up A Remote Mirror Resync

The remote mirror software, with the `sndradm -I` command, enables the system administrator to configure a point-in-time shadow volume set to be used just prior to a remote mirror software resync operation. The remote mirror secondary volume is the master volume of the point-in-time shadow volume set. Just before the remote mirror resync, the point-in-time shadow volume set is enabled. If the remote mirror resync fails for any reason, the secondary volume, as the master in its point-in-time shadow volume set, can be restored by performing a point-in-time copy software shadow-to-master update.

Note – The volumes specified for use as the point-in-time shadow volume set (shadow and bitmap) must not be in use for any other purpose. Loss of the remote mirror secondary volume's data can occur if the resync fails and the point-in-time volumes have been concurrently used for another purpose.

It is possible to set up configurations in which this situation can occur, particularly in both multi-hop remote mirror replication and remote mirror disaster recovery scenarios. The system administrator must assure that this does not happen. See the *Sun StorEdge Availability Suite 3.1 Remote Mirror Software Administration and Operation Guide* for further information.

Point-in-Time Copy Software with Sun Cluster 3.0

Point-in-time copy software volumes can be hosted in the Sun Cluster 3.0 Update 1 environment and the Sun Cluster 3.0 Update 2 environment. This allows point-in-time copy replications to failover, or continue unaffected, if a node hosting point-in-time copy shadow volume sets crashes. Failing over involves placing the volumes of the affected node under the control of another node in the cluster, and continuing the replication when the new node takes control. This process is automated by Sun Cluster as part of its control of volume management subsystems.

An important component of successful point-in-time copy software failover is how the shadow volume sets are configured in a Sun Cluster resource group. A resource group is a grouping of items in a Sun Cluster which are interrelated in such a way as to make it impossible to failover a single member of the group without failing over all members of the group. That is, members of a resource group are dependent upon one another when a node in the cluster is failed over. Detailed information about resource groups is available in the Sun Cluster documentation.

Note – Sun Cluster 3.0 Update 1 is also known as the Sun Cluster 3.0 07/01 release; Sun Cluster 3.0 Update 2 is also known as the Sun Cluster 3.0 12/01 release.

Instant Image Failover

When the point-in-time copy software is running in a cluster and the node it is running on fails, the Sun Cluster software detects the failure and initiates failover. Conceptually, failover involves restarting the processes which were running on the failing node on another node without losing any information. This information is application dependent and outside the control of Sun Cluster. Sun Cluster coordinates the movement of the associated file systems, shadow volume sets, volumes, networking, and configuration data.

In the case of point-in-time copy software, this means that control of the volumes being referenced (i.e. master volume, shadow volume, and bitmap volume) must be moved to the new node. Then, operation is restarted at the point that it left off.

The point-in-time copy software must be configured such that the master volume, the shadow volume, the bitmap volume, and the overflow volume are part of the same volume manager resource group. This is important because all members of the resource group must be available at the point at which the point-in-time copy software comes up in the boot sequence.

Because of its position in the kernel I/O stack, failing over the point-in-time copy software is similar to failing over a volume manager. The Sun StorEdge software and the Sun Cluster software work together to ensure that I/O processing on point-in-time copy volumes is enabled at the correct point in the failover process on the new node, and that processing on in-transit I/O is completed. The bitmap volumes are utilized to continue with operations on the new node. The bitmap volumes for point-in-time copy volumes running in a Sun Cluster must be disk-based, not memory based.

Additional Performance Considerations

Several performance considerations exist for Sun StorEdge Availability Suite 3.1 point-in-time copy software and this list is not comprehensive. Consider the following when configuring a system for use with point-in-time copy software:

- Configure to avoid hot spots by not putting multiple volumes on a single spindle.
- Disk speeds of both the master volume and the shadow volume will affect the performance of the master volume during independent shadow volume set synchronizations.
- The performance of the master volume is affected by accesses to a dependent shadow volume.

CHAPTER 5

Command-Line Interface

This chapter describes the point-in-time copy software's CLI and provides the syntax of the CLI commands. The CLI is designed to control a single shadow volume set, an I/O group consisting of more than one shadow volume set, or selected volume sets or I/O groups within a specific Sun Cluster Resource Group. The CLI can be invoked interactively or incorporated into a shell script. This chapter includes:

Command-Line Interface

The primary user interface to the point-in-time copy software is the `/usr/opt/SUNWesm/sbin/iiadm` command.

The following usage applies to all examples and explanations in this chapter.

Name	Definition
master	The specific master volume referenced in the command
shadow	The specific volume set, whose name defaults to the name of its associated shadow volume
shadowvol	The name of a specific shadow volume
bitmap	The name of a specific bitmap volume
cluster	The name, or *tag*, of a specific Sun Cluster
io-groupname	The specific I/O group, whose name is given upon creation by naming it in a `move` (to group) or an `enable` command
overflow	The specific overflow volume referenced in the command

Command Modifiers

The `iiadm` CLI performs only one action per command invocation, as such you cannot combine multiple options, except with the `-g` and `-C` command modifiers. The order in which command modifiers are entered is important: they must be entered after the `iiadm` command and before any other command options or variables, such as names. Note also that the command option is the last non-variable entry on a command line, which prevents confusion with volume names.

`iiadm -g`

This command modifier extends most, but not all, existing commands to act on I/O groups. No specific volume set names are needed because the command is carried out against all volume sets within the I/O group specified.

Syntax

`iiadm -g` *io-groupname* `-l`

This command line lists (`-l`) all volume sets in the I/O group, *io-groupname*.

iiadm -C

This command modifier enhances all existing commands to act on all volume sets associated with the specified Sun Cluster tag. In a cluster environment, enabling a shadow volume set causes it to be tagged with the current cluster tag. Enable volume sets so that master, shadow, bitmap, and overflow volumes are in the same disk group, physical and logical, so that failover will work correctly. Commands executed in a cluster environment are executed against all volume sets associated with the current cluster. If you need to execute commands against a cluster other than the current one, the `-C` option enables you to do that.

Syntax

`iiadm -C` *cluster* `-l`

This command line lists (`-l`) all shadow volume sets associated with the specified Sun Cluster whose tag is *cluster.*

`iiadm -C` *cluster* `-g` *io-groupname* `-l`

This command line lists all shadow volume sets in the specified I/O group in the cluster whose tag is *cluster.*

Caution – *Do not* install the Sun StorEdge Version 3.0/3.1 core and data services software on servers in a Sun Cluster 3.0 environment.

- The Version 3.0 and 3.1 software is *not* coexistent with the Sun Cluster 3.0 environment.
- The Version 3.0 and 3.1 core and data services software is cluster aware in the Sun Cluster 3.0 Update 1 and Update 2 environments and provides high availability for the Sun StorEdge data services.

-n Option to the -c, -A, -O, and -u Command Options

This command modifier, when used with a copy or update command, skips the printing of a query to `stderr` that asks if you are sure you want to update the master volume. This is particularly useful in scripts where you don't want the process to halt waiting for an operator answer. See the full command descriptions for details and warnings about using the `-n` option.

Syntax

`iiadm -n -c m` *shadow*

`iiadm -n -u m` *shadow*

Command Options

When you enter an `iiadm` command, you must enter an option, telling the Instant Image software what to do, and usually one or more volume, set, and group names, telling the Instant Image software where to execute the command.

With few exceptions, only one option can be entered on a command line. Here are some examples of command lines illustrating the enable option, `-e`.

`iiadm -e ind` *master shadowvol bitmap*

`iiadm -g` *io-groupname* `-e dep` *master shadowvol bitmap*

`iiadm -C` *cluster* `-e ind` *master shadowvol bitmap*

Help Information

If an unsupported action item is entered, `iiadm` displays a brief summary of the help text to `stderr`. The help message text can also be called up with the `iiadm -h` option.

If more than one action item or an incorrectly specified action item is entered, `iiadm` displays the specific error message to `stderr`. Here is an example of an incorrectly specified action.

```
# iiadm -d
iiadm: option requires an argument -- d
iiadm: unrecognized argument

Brief summary:
             -e {ind|dep} master_vol shadow_vol bitmap_vol
             -[cu {s|m}] volume_set
             -[irs] all
             -[adDEilPRw] volume_set
             -g group_name [options]
             -f config_file [options]
             -C cluster_tag [options]
             -[hilLrsv]
             -[IJ] bitmap volume_set
             -[OQ] overflow_vol
             -P {delay} {units} volume_set
```

Configuration Parameters

There are four configuration parameters that directly influence the operation of the Instant Image software. You can use the `vi` editor to modify these parameters.

Note – When changing any of the parameters described in this section, you must shut down and restart your system in order for the changes to take effect.

---

ii_bitmap

This parameter defaults to `1` upon installation of the software. Possible settings are: `0` and `1`. When you enable volume groups, they have this value associated with them. Edit `/usr/kernel/drv/ii.conf`.

If set to `0`, the bitmaps are maintained in memory only and their contents are stored in their respective bitmap volumes across shutdowns. However, if the system crashes, there is a possibility of resulting inconsistent data. This is especially the case if the system crashes during a shadow-to-master update in a compact dependent shadow set.

If set to `1`, the bitmaps is maintained in the bitmap volume.

ii_throttle_units

This variable holds the default value for *units* in the copy parameter command, `iiadm -P`. You can set the default value (from 100 to 60000) for all newly enabled sets by modifying the `ii_throttle_units` setting in `/usr/kernel/drv/ii.conf`.

ii_throttle_delay

This variable holds the default value for *delay* in the copy parameter command, `iiadm -P`. You can set the default value (from 2 to 10000) for all newly enabled sets by modifying the `ii_throttle_units` setting in `/usr/kernel/drv/ii.conf`.

ii_debug

This variable defaults to a value of `0`, but can be set to `1` or `2`, which generates debugging messages to the console. None of the settings changes the operation of the point-in-time copy software.

Option List

All the command options are listed in alphabetical order in the following pages of this chapter.

To Abort a Copy or Update

Use this option when you need to abort any copy or update before it executes to completion. Data remains valid, but the update or copy to the target volume is incomplete. The target volume is now a dependent copy of the source volume. At a later time, you can re-issue the update or copy option to resynchronize the volumes.

Note – See "Cautions: Enable, Copy, and Update" on page 32

This option aborts any current copy or update operation on the specified volume set or on all volume sets within an I/O group.

`iiadm -a`

Syntax

`iiadm -a` *shadow*

`iiadm -g` *io-groupname* `-a`

To Attach an Overflow Volume

This option attaches an overflow volume to a volume set with a compact dependent shadow or to all volume sets within an I/O group that have compact dependent shadows.

Note – An overflow volume must be initialized before it is attached to a volume set. See "To Initialize an Overflow Volume" on page 96.

Use this option after you enable a volume set with a compact dependent shadow. You can also use this command to attach an overflow volume to an I/O group containing volume sets with compact dependent shadows.

This option supports the optional `-n` parameter, no prompting, which will directly perform the requested action without prompting. This option is typically used during script processing.

If the volume you are attaching has not yet been initialized (see "To Initialize an Overflow Volume" on page 96), it will be before it is attached so be certain that you want to initialize the volume, especially when using the `-n` option.

Note – When using the `-n` option, the `-n` must precede the `-A` option on the command line. For example: `-nA` or `-n -A`.

`iiadm -A`

Syntax

`iiadm -A` *overflow shadow*

`iiadm -g` *io-groupname* `-A` *overflow*

`iiadm -nA` *overflow shadow*

`iiadm -g` *io-groupname* `-nA` *overflow*

To Copy Volumes

Note – See "Cautions: Enable, Copy, and Update" on page 32

The copy option writes all the data from either the master volume of the volume set to the shadow volume or from the shadow volume to the master volume. You can also copy all masters to all shadows or all shadows to all masters within an I/O group.

If you do a shadow-to-master copy, the system prints a query that asks if a shadow-to-master copy is what you really want. You can stop this message from being printed with the `-n` command option.

You can place the shadow volume set under PID lock by using the `-p` option. See also "PID Locking a Shadow Volume Set" on page 51.

Use this option to synchronize the master and shadow volumes with a full volume copy. If you do not need a full copy, use the update option instead.

`iiadm -c`

Syntax

`iiadm -c { s | m }` *shadow*

`iiadm -g` *io-groupname* `-c { s | m }`

`iiadm -p -c { s | m }` *shadow*

`iiadm -g` *io-groupname* `-p -c { s | m }`

Options

`s`	update from master to shadow
`m`	update from shadow to master

Note – A remote mirror volume set must be in logging mode for the point-in-time copy software to successfully perform an enable, copy, update, or reset operation on a remote mirror volume. If not, the point-in-time copy operation fails and the remote mirror software reports that the operation is denied.

To Disable Volume Sets

Note – See "Interaction With svadm" on page 33.

The disable option disables a volume set or disables all volume sets in an I/O group. If you disable an independent volume set, the shadow volume contains valid data and remains accessible. If you disable a dependent volume set, the shadow does not contain valid data, the first 64KB of the volume are erased, and the volume becomes inaccessible for data retrieval.

Use this option to disable the point-in-time copy software for specific volume sets or all volume sets in specific I/O groups so that it no longer maintains the shadow volumes' point-in-time snapshots or the associated bitmaps.

Use this option to terminate the volume set association when the volume set is no longer needed, such as when you want an independent shadow copy to use for testing. Once disabled, the volume set is no longer usable except as stand-alone copies, and cannot be re-enabled.

`iiadm -d`

Syntax

`iiadm -d` *shadow*

`iiadm -g` *io-groupname* `-d`

To Detach an Overflow Volume

Use this option to detach an overflow volume from a volume set or from all volume sets within an I/O group.

This option executes only if the volume set has no data on the overflow volume. One such instance is immediately after a point-in-time snapshot of the master is taken.

`iiadm -D`

Syntax

`iiadm -D` *shadow*

`iiadm -g` *io-groupname* `-D`

To Enable Volume Sets

Note – See "Cautions: Enable, Copy, and Update" on page 32

Note – Do not create bitmaps on cylinder 0 because the point-in-time copy software does a raw write and destroys the VTOC for that device.

Note – You must use a volume for a bitmap. File-based bitmaps are not supported in the Instant Image software Version 3.0.

Note – The remote mirror and point-in-time copy software does not support metatrans devices (also known as trans metadevices) created by the Sun Solstice DiskSuite and Solaris Volume Manager software.

The enable option enables an Instant Image volume set, composed of the specified master volume, shadow volume, and bitmap volume, then causes a point-in-time snapshot of the master volume, either dependent or independent, to be taken on the shadow volume.

You can enable the volume set as a member of an I/O group by specifying the name of the I/O group. If the I/O group exists, the volume set is attached to it; if the I/O group does not exist, it is created and the volume set is attached to it. The volume set is enabled as a compact dependent shadow volume if the shadow volume you specify is smaller than the master volume. You can, and should, attach an overflow volume later with the `-A` option to any compact dependent shadow volume set.

Use this option to initially create Instant Image volume sets. You can create the volume sets with or without assigning them to an I/O group. At a later time, you can move any volume set into any I/O group with the move option.

`iiadm -e`

Syntax

`iiadm -e { ind | dep }` *master shadowvol bitmap*

`iiadm -g` *io-groupname* `-e { ind | dep }` *master shadowvol bitmap*

Options

`ind`	Independent shadow; all master volume data is copied to the shadow volume.
`dep`	Dependent shadow; a master volume-dependent, point-in-time snapshot of the master volume, which is accessible through the shadow volume.

Note – A remote mirror volume set must be in logging mode for the point-in-time copy software to successfully perform an enable, copy, update, or reset operation on a remote mirror volume. If not, the point-in-time copy operation fails and the remote mirror software reports that the operation is denied.

To Export a Shadow Volume

Use this option to export a shadow volume or all the shadow volumes in an I/O group so that another host can import them. One instance when you would want to do this is to continue business transactions on a secondary host while the primary host is unavailable.

`iiadm -E`

Syntax

`iiadm -E` *shadowvol*

`iiadm -g` *io-groupname* `-E`

To Display Command Usage

Use this option to display a summary of usage for all point-in-time copy software options and modifiers.

When you enter a command improperly, `iiadm` sends an abbreviated command usage summary to `stderr`. Use the `-h` option to display the usage summary of all `iiadm` options.

`iiadm -h`

Syntax

```
iiadm -h
```

To Display Volume Set Status

Use this command when you want to display the status of volume sets. Specify the name of the volume set to check its status, or enter `all` to check all volume sets. A typical use of this option is to check the status of a copy or update operation before using the target volume for any other operations. See also "To Wait for a Copy or Update to Complete" on page 104.

`iiadm -i`

Syntax

`iiadm -i { `*shadow*` | all }`

`iiadm -g `*io-groupname*` -i`

Options

shadow	display the status of the named volume set
`all`	display the status of all volume sets

Sample output:

```
# iiadm -i /dev/vx/rdsk/shadowdg/shadowvol1
/dev/vx/rdsk/masterdg/mastervol1: (master volume)
/dev/vx/rdsk/shadowdg/shadowvol1: (shadow volume)
/dev/vx/rdsk/rootdg/bitvol1: (bitmap volume)
Independent copy (this is an independent shadow volume set)
Volume size:      200
Percent of bitmap set: 0 (the bitmap is clean)
```

To Import a Shadow Volume

Use this option to import a previously exported shadow volume using a specified bitmap. Use this bitmap later to rejoin the shadow volume to its master volume. This is typically done by a secondary host importing the shadow volume exported by the primary host.

`iiadm -I`

Syntax

`iiadm -I` *shadowvol bitmap*

To Join a Shadow Volume

Use this command to join the shadow volume to its master volume using a specific bitmap. See “Exporting, Importing, and Joining Shadows” on page 62 for the sequence for joining a shadow volume to its original master volume.

iiadm -J

Syntax

`iiadm -J` *shadowvol bitmap*

Options

bitmap bitmap volume

Note – You may not need to copy the bitmap to Host A if the bitmap is on a dual-ported drive.

Caution – Once a shadow is rejoined to its original master on Host A, Host B should not use the shadow volume even if it is still accessible.

To List Volume Sets

This option lists either all volume sets or all volume sets in an I/O group. All configured volumes sets, including offlined and suspended volume sets, are shown.

`iiadm -l`

Syntax

`iiadm -l`

`iiadm -g` *io-groupname* `-l`

Sample output:

```
# iiadm -l
ind /dev/vx/rdsk/masterdg/mastervol5
/dev/vx/rdsk/shadowdg/shadowvol5 /dev/vx/rdsk/rootdg/bitvol5
dep /dev/vx/rdsk/masterdg/mastervol4
/dev/vx/rdsk/shadowdg/shadowvol4 /dev/vx/rdsk/rootdg/bitvol4
ind /dev/vx/rdsk/masterdg/mastervol3
/dev/vx/rdsk/shadowdg/shadowvol3 /dev/vx/rdsk/rootdg/bitvol3
dep /dev/vx/rdsk/masterdg/mastervol2
/dev/vx/rdsk/shadowdg/shadowvol2 /dev/vx/rdsk/rootdg/bitvol2
ind /dev/vx/rdsk/masterdg/mastervol1
/dev/vx/rdsk/shadowdg/shadowvol1 /dev/vx/rdsk/rootdg/bitvol1
```

Note – Entering `iiadm` with no options lists the same information as entering `iiadm -l`.

To List Overflow Volumes or Groups

This option lists either all overflow volumes or all I/O groups.

`iiadm -L`

Syntax

`iiadm -L`

`iiadm -g -L`

`Sample output:`

```
# iiadm -g -L
groupone
# iiadm -L
/dev/vx/rdsk/rootdg/overflowvol
```

To Move a Volume Set

Use this option to move one or more volume sets into an I/O group. The volume set can be a member of an existing I/O group or not a member. The move operation does not involve data operations so there is no need to quiesce the application before moving the volume set. You can move the volume set into the NULL I/O group, meaning no I/O group, using two double-quote marks instead of an I/O group name.

Most point-in-time copy software options, such as the copy option, can be used with the group command and help to simplify operations. Use the move option to place volume groups into I/O volume groups so they can be operated on at the same time with single commands.

`iiadm -m`

Syntax

`iiadm -g` *io-groupname* `-m` *shadow-1 shadow -2 ... shadow-n*

`iiadm -g “” -m` *shadow*

To Initialize an Overflow Volume

This option initializes an overflow volume for use with volume sets that were enabled with compact dependent shadow volumes.

Once initialized, the overflow volume can be attached to any number of compact dependent shadow volumes, depending on available space.

If existing overflow volume space is insufficient, use this option after you enable a volume set with a compact dependent shadow volume, then attach the overflow volume to the volume set to receive any data overflows.

This option supports the optional `-n` parameter, no prompting, which will directly perform the requested action without prompting. This option is typically used during script processing.

Note – When using the `-n` option, the `-n` must precede the `-O` option on the command line. For example: `-nO` or `-n -O`.

`iiadm -O`

Syntax

`iiadm -O` *overflow*

`iiadm -n -O` *overflow*

To Set Copy Parameters

Use this option to tune the copy process so that the copy does not excessively consume system resources. This *throttled copy* is tuned by setting the maximum number of chunks of data (100 through 60000) to be sent before pausing and by setting the length of the pause, in system clock ticks (2 through 10000), before sending the next series of chunks of data. In this way, the copy proceeds to completion without preventing other processes from using the system. The software defaults to minimum settings for both parameters.

Get copy parameters for a specific volume set with the `iiadm -P shadow` command. Get copy parameters for an I/O group with the `iiadm -g` *io-groupname* `-P` command.

The point-in-time copy software, at installation, defaults to the minimum settings for both parameters: *delay* and *units*. Use this option to speed up copies if your system has idle time during copy operations. Use this command to slow down copies if other applications require more system resources during copy operations.

Note – If a bitmap volume goes offline, the *delay* and *units* parameters are reset to the default values when the volume is reset with the `iiadm -R` command.

`iiadm -P`

Syntax

`iiadm -P` *delay units shadow*

`iiadm -P` *shadow*

`iiadm -g` *io-groupname* `-P` *delay units*

`iiadm -g` *io-groupname* `-P`

Options

delay number of system clock ticks (2-10000) between copy operations

units number of chunks of data (100-60000) sent before pausing for the set delay

Sample output:

```
# iiadm -P /dev/vx/rdsk/shadowdg/shadowvol3
volume: /dev/vx/rdsk/shadowdg/shadowvol3
copy delay: 2
copy unit: 100
# iiadm -g groupone -P
volume: /dev/vx/rdsk/shadowdg/shadowvol2
copy delay: 2
copy unit: 100
volume: /dev/vx/rdsk/shadowdg/shadowvol4
copy delay: 3
copy unit: 1000
```

To Display Overflow Volume Status

This option displays the status of the specified overflow volume.

Note – In order for this command option to work correctly, the overflow volume must be attached to at least one compact dependent shadow volume.

Use this option to determine:

- The number of volume sets attached to the overflow volume.
- The number of volume sets currently using the overflow volume.
- The total number of chunks in the overflow volume.
- The total number of used chunks.
- The total number of available chunks.

`iiadm -Q`

Syntax

`iiadm -Q` *overflow*

Sample output:

```
# iiadm -Q /dev/vx/rdsk/rootdg/overflowvol
Total number of attached shadows: 1
Number of currently attached shadows: 1
Total number of chunks: 159
Number of chunks ever allocated: 1
Number of unused chunks: 158
```

To Reset a Volume Set or I/O Volume Group

This option places specific volume sets that are offline or all volume sets in an I/O group that are offline to the online state.

Use this option to return volume sets to the online state after the condition that caused them to be marked offline has been corrected. After executing this option, the best state available for the volume set will be reestablished. A point-in-time snapshot will be created or, in the case of independent copy volume sets where the bitmap was offline, the master and shadow volumes will be marked as different.

Note – If a bitmap volume goes offline, the *delay* and *units* parameters are reset to the default values when the volume is reset with the `iiadm -R` command.

`iiadm -R`

Syntax

`iiadm -R` *shadow*

`iiadm -g` *io-groupname* `-R`

Note – A remote mirror volume set must be in logging mode for the point-in-time copy software to successfully perform an enable, copy, update, or reset operation on a remote mirror volume. If not, the point-in-time copy operation fails and the remote mirror software reports that the operation is denied.

▼ To Reset a Master Volume with Multiple Shadows

Caution – The `iiadm -g` *group* `-R` command *must not* be used with multiple shadows of single masters because all the resets occur at the same time with unpredictable results.

If the master volume with multiple shadow volumes is placed offline, return it to the online state with the `iiadm -R` command, as follows:

1. **Reset all independent shadow sets, except the shadow set that you want to update the master volume with.**

```
# iiadm -R shadow shadow2 ... shadowN
```

2. **Wait for all reset operations to finish.**

```
# iiadm -w shadow shadow2 ... shadowN
```

3. **Reset the volume set containing the shadow volume you want to update the master volume with.**

```
# iiadm -R shadow
```

4. **Wait for the copy to complete.**

```
# iiadm -w shadow
```

5. **Reset all dependent shadow sets.**

```
# iiadm -R shadow shadow2 ... shadowN
```

To Update a Volume Set

Note – See "Cautions: Enable, Copy, and Update" on page 32

Use this option to update a volume set or all volume sets in an I/O group, either from master to shadow or from shadow to master.

If you do a shadow-to-master update, the system prints a query that asks if a shadow-to-master update is what you really want. You can stop this message from being printed with the `-n` command option.

You can place the shadow volume set under a PID lock with the `-p` option. See also "PID Locking a Shadow Volume Set" on page 51.

Use this option whenever you want to update the point-in-time shadow copy of the master volume or to update the master from the shadow. The update option incrementally updates the volumes with only changed data, as tracked in the bitmaps, and is very quickly completed, as compared to a full volume copy, which takes longer. If you need a full volume copy, use the copy option.

`iiadm -u`

Syntax

`iiadm -u { s | m }` *shadow*

`iiadm -g` *io-groupname* `-u { s | m }`

`iiadm -p -u { s | m }` *shadow*

`iiadm -g` *io-groupname* `-p -u { s | m }`

Options

`s`	update from master to shadow
`m`	update from shadow to master

Note – A remote mirror volume set must be in logging mode for the point-in-time copy software to successfully perform an enable, copy, update, or reset operation on a remote mirror volume. If not, the point-in-time copy operation fails and the remote mirror software reports that the operation is denied.

To Print Software Versions

Use this option to print the version of the major software modules when you need to know the software versions that are installed. One such time is during a software update.

iiadm -v

Syntax

`iiadm -v`

Sample output:

```
# iiadm -v
InstantImage version 3.n.N
```

To Wait for a Copy or Update to Complete

This option causes the system to wait for the point-in-time copy software to complete any in-progress copy or update operation on a volume set or on all volume sets in an I/O group before allowing another command to execute on that volume set.

Use this command when you need to be sure the copy or update operation has completed before invoking another command that affects that volume set.

- Use this command after enabling a shadow volume set with the independent option or after you have issued a copy or update command.
- Use this command before you disable a shadow volume set that could have a copy or update operation in progress.

You can also use this command with the `-p` option to remove a process ID lock (PID lock) from a volume set if you are the originating process. If you are root, you can use the `-n` option with the `-p` option to remove a PID lock from a shadow volume set even if the originating process has terminated.

This command is useful when used as part of a script.

`iiadm -w`

Syntax

`iiadm -w` *shadow*

`iiadm -g` *io-groupname* `-w`

`iiadm -p -w` *shadow*

`iiadm -p -n -w` *shadow*

APPENDIX **A**

Software Statistics

This chapter describes the operation of the `dsstat` utility. The `dsstat` utility collects and reports I/O statistics for the Sun StorEdge Availability Suite 3.1 software components.

The main sections of this appendix are:

Remote Mirror and Point-in-Time Copy Software Statistics: The dsstat Command

The `/usr/opt/SUNWscm/sbin/dsstat`(1M) command displays I/O statistics for the remote mirror, point-in-time copy, and cache software. See the `dsstat`(1SCM) for more information.

Syntax

`dsstat -m` *mode* [`-r` *report–options*][`-d` *display-options*][`-s` *volume-sets*] [`-f` | `-F`] [-z] [*interval* [*count*]]

See TABLE A-1 for the command options.

See TABLE A-2 for a description of the display fields.

Command Options

TABLE A-1 `dsstat` Command Options

Option	Description
`-m` *mode*	Specifies the mode of operation for `dsstat`. This is a required option. The mode choices are: • `ii` for point-in-time copy statistics • `sndr` for remote mirror statistics • `cache` for `SUNWnvm` (Fast Write Cache) statistics If `-m` is not specified, the default is for the statistics for all modes to be displayed. If more than one mode is displayed, field headers are output for each report cycle, overriding the `[-f|-F]` option.
`-r` *report options*	Specifies the volume components to be displayed. Each item is represented by a single character. You can select multiple items. The report options vary depending on the *mode* selected. Choices are: For *mode* = `ii` (default = `'msbo'`) • `m` for master volume statistics • `s` for shadow volume statistics • `b` for bitmap volume statistics • `o` for overflow volume statistics (if attached) For *mode* = `sndr` (default = `'bn'`) • `b` for bitmap volume statistics • `n` for network volume statistics For *mode* = `cache`: There are no *report options* for `cache` mode.
`-h`	Prints a detailed usage message to `stderr`. See "Output of the `dsstat -h` command." on page 113.

TABLE A-1 `dsstat` Command Options

`-d` *display options*	Specifies the statistics to be displayed. The types of statistics are represented by a single character. You can select multiple types. The display options vary depending on the *mode* selected. The choices are: For *mode* = `ii` (default = `'sf'`) • `r` for detailed read statistics • `w` for detailed write statistics • `t` for timing statistics • `s` for summary statistics • `f` for volume type/status flags For *mode* = `sndr` (default = `'spf'`) • `r` for detailed read statistics • `w` for detailed write statistics • `t` for timing statistics • `s` for summary statistics • `f` for volume type/status flags • `p` for percentage of volume requiring sync • `a` display alternate (secondary) names for volumes **Note:** Statistics displayed in `sndr` mode default to the host from which the command was issued. Use the `-a` option to see the alternate host. For *mode* = `cache` (default = `'sf'`) • `r` for detailed read statistics • `w` for detailed write statistics • `d` for destaged data statistics • `c` for write cancellation statistics • `s` for summary statistics • `f` for cache behavior flags
`-s` *volume sets*	Filters the display output to include only the specified volume sets. Where *volume sets* is a comma-delimited list of volume names.
`-f | -F`	• `-f` Outputs a field header for every report cycle. • `-F` Outputs a field header once, when reporting begins.
`-z`	Suppress report lines that have zero values (no activity).
interval [*count*]	If you do not enter either an *interval* or a *count*, you will get a single report for a one-second interval. If you enter only an *interval*, you will get continuous reports for *interval*-second periods of time. If you enter an *interval* and a *count*, you will get *count* number of reports at *interval*-second periods of time.

Display Fields

TABLE A-2 `dsstat` Display Fields

Fields	Description
`set name`	Volume set specified by `-s` option
`t`	Valid volume type for point-in-time copy software: • `I` - independent shadow volume set • `D` - dependent shadow volume set Valid volume type for remote mirror software: • `P` - Primary host of the volume set • `S` - Secondary host of the volume set Valid cache modes for cache: • `C` - Cache reads and writes • `D` - Disk reads and writes
`s`	Valid volume status for point-in-time copy software: • `C for` copy in progress • `- for` no copy in progress Valid volume status for remote mirror software: • `L` - Changes to this volume are being logged • `SY` - Sync in progress: sending data • `RS` - Reverse sync in progress: sending data • `SN` - Sync needed • `RN` - Reverse sync needed • `VF` - Volume failed • `BF` - Bitmap failed • `FF` - FCAL failed
`sn`	Percentage of volume requiring sync
`rtps`	Number of reads
`rkps`	Kilobytes read
`wtps`	Number of writes
`wkps`	Kilobytes written
`svt`	Service time per operation
`tps`	Sum of `rtps` and `wtps`
`kps`	Sum of `rkps` and `wkps`
`r`	Cache read behavior

TABLE A-2 `dsstat` Display Fields

Fields	Description
`w`	Cache write behavior
`ckps`	Kilobytes read from cache
`dkps`	Kilobytes read from disk
`hit`	Read hits during interval
`ds/s`	Kilobytes destaged from cache
`cn/s`	Number of write cancellations

Note – Note: Unless otherwise specified, all fields are per-second averages based on the data collected during the specified interval. If you specify 5 second intervals, the data you see is a per-second average of the data collected over the last 5 second interval.

Examples

- Report Cache statistics, with detailed breakdowns of read and writes to cache / disk. Generate reports at 5 second intervals. Limit reporting to the set `c1t1d0s0`.

Note – Only the trailing sixteen characters of a volume name appear in a report. For example: `/dev/rdsk/c1t1d0s0` is displayed as `ev/rdsk/c1t1d0s0`.

```
# dsstat -m cache -d rw -s /dev/rdsk/c1t1d0s0 5

                          - read -              - write -
    volume              ckps   dkps    hit    ckps   dkps    hit
    ev/rdsk/c1t1d0s0       0      0   0.00       0      0   0.00
    ev/rdsk/c1t1d0s0       3   2396   0.13     983    763 100.00
    ev/rdsk/c1t1d0s0    2399    799  75.00    2815   2686 100.00
    ev/rdsk/c1t1d0s0    3200    800  80.00    2755   2908 100.00
    ev/rdsk/c1t1d0s0    3999    799  83.33    2809   2868 100.00
    ev/rdsk/c1t1d0s0    4800    800  85.71    2867   2931 100.00
```

- Report master and shadow statistics for point-in-time copy software, using summary output, including volume type / status flags. Generate reports at five second intervals.

```
# dsstat -m ii -r ms 5

                               - master -       - shadow -
    set name          t  s     kps  tps  svt     kps  tps  svt
    ev/rdsk/c0t1d0s0  I  -       0    0    0       0    0    0
    ev/rdsk/c0t1d0s0  I  -    9047  219    3    9040  219    7
    ev/rdsk/c0t1d0s0  I  -   13548  317    3    9760  243    6
    ev/rdsk/c0t1d0s0  I  -    5946  155    3    9684  227    8
    ev/rdsk/c0t1d0s0  I  -   16539  417    2    9242  225    7
```

- Report network statistics for remote mirror software, using detailed breakdowns of read and write statistics, timing, and volume type/status flags. Limit reporting to the set `c1t1d0s0`. Generate reports at five second intervals.

```
# dsstat -m sndr -r n -d rwtpf -s /dev/rdsk/c1t1d0s0 5

                                          - network -
    set name          t  s     sn    rkps rtps   wkps wtps  svt
    ev/rdsk/c1t1d0s0  P  L  77.27       0    0      0    0    0
    ev/rdsk/c1t1d0s0  P SY  75.07       0    0   2317   72   11
    ev/rdsk/c1t1d0s0  P SY  71.67       0    0   3443  108    9
    ev/rdsk/c1t1d0s0  P SY  69.37       0    0   2426   76   12
    ev/rdsk/c1t1d0s0  P SY  66.68       0    0   2765   86   11
```

- Output of the `dsstat -h` command.

```
# dsstat -h
dsstat [-m <mode>] [-f | -F] [-z] [-s <sets>] [-r <flags>] [-d <flags>] \
[<interval> [<count>]]
        -d <flags> Specifies the statistics to be displayed

           For 'cache' mode
              Valid <flags> are 'rwfsdc', default <flags> are 'sf'
              r=read, w=write, f=flags, s=summary,  d=destaged,
              c=write cancellations

           For 'ii' mode;
              Valid <flags> are 'rwtfs', default <flags> are 'sf'
              r=read, w=write, t=timing, f=flags, s=summary

           For 'sndr' mode;
              Valid <flags> are'rwtfpsa', default <flags> are 'spf'
              r=read, w=write, t=timing, f=flags, p=percentages,
              s=summary a=alternate volume names

        -f  prints field headers once for each iteration

        -F  prints field headers once, at the start of reporting

        -h  prints detailed usage message

        -m <mode> where mode is, 'cache', 'ii', or 'sndr'
           Multiple mode switches may be used.

        -r <flags> specifies components to be reported

           For 'cache' mode, this option is not used.

           For 'ii' mode;
              Valid <flags> are 'msbo', default <flags> are 'msbo'
              m=master, s=shadow, b=bitmap, o=overflow

           For 'sndr' mode;
              Valid <flags> are 'nb', default <flags> are 'nb'
              n=network, b=bitmap

        -s <sets> outputs specified sets
            Where <sets> is a comma delimited list of set names

        -z  suppress reports with zero value (no activity)

        <interval> is the number of seconds between reports

        <count> is the number of reports to be generated

#
```

Glossary

atomic Term used to indicate that a number of actions occur simultaneously. For instance, actions taken on grouped shadow volume sets occur at the same time, or atomically.

bitmap volume For every 32K block of a *master volume* that is part of a *shadow volume set*, a bit in a bitmap volume is maintained which indicates if the data at the block has changed with respect to its associated *point-in-time copy*.

changed Term applied to data blocks whose contents have been altered since the last synchronization.

compact dependent shadow volume A *shadow volume* that is smaller than its associated *master volume*. The use of the term compact is intended to convey that less storage is allocated, not that the data in the blocks is compacted or compressed in any way.

delay variable Part of the `iiadm -P` command. This variable sets the number of system clock ticks to wait before sending the next *units variable* set of data chunks.

dependent A *shadow volume set* can be *enabled* as dependent, in which case no *master volume* to *shadow volume* copy is performed. After the shadow volume set has been enabled, the shadow volume cannot stand alone. It is dependent on the master volume to service reads.

enable In the Sun StorEdge Availability Suite 3.1 point-in-time copy software, you use the enable command to associate the constituent volumes of a *shadow volume set* with one another.

export With point-in-time copy software, you export a *shadow volume* to make it available for another host to use.

fast synchronization See *update*.

full synchronization See *full volume copy*.

full volume copy To copy the full contents of the master volume to the shadow volume or the shadow volume's full contents to the master volume.

hot-backup A mode of operation available with some database applications wherein there is no need to quiesce a *shadow volume set* prior to a *point-in-time copy.*

import A host can import, for its own use, a *shadow volume* that has been exported by another host. The original host maintains the *shadow volume set* relationship for later *joining* of the shadow volume to its original shadow volume set. The importing host must maintain a record of changes to the imported shadow in a bitmap.

independent A *shadow volume set* can be *enabled* as independent, in which case a full volume copy is performed, *master volume* to *shadow volume.* After the copy, the shadow volume can stand alone, or independently.

join A *shadow volume* that has been *exported* can be joined to its original *shadow volume set* with the join command.

master volume The volume containing the original data.

overflow volume A volume that is attached to a *compact dependent shadow volume* to accept writes that are in excess of the compact dependent shadow volume's capacity.

point-in-time copy See *point-in-time snapshot*.

point-in-time snapshot A view of a data volume's contents relative to a point in time. While the data volume itself may continue changing with time, the point-in-time snapshot does not.

remote mirror The remote mirror software provides for remote replication of data volumes over many line protocols including TCP/IP. Remote mirror software is a part of Sun StorEdge Availability Suite software.

resynchronize A synchronization which occurs between volumes in a *shadow volume set* that have been previously *synchronized*. This can be a full volume copy or an *update*.

scoreboard Another name for *bitmap volume*.

shadow volume set The minimum configuration of volumes for use by the point-in-time copy software, consisting of a *master volume*, a *shadow volume*, and a *bitmap volume*. Optionally, a shadow volume set may include an *overflow volume.*

shadow volume The volume containing a point-in-time copy of the data on its associated *master volume*.

synchronize To make the *shadow volume* and the *master volume* of a *shadow volume set* match one another. This can involve a full volume copy or only the clearing of a bitmap, depending on the type of shadow volume set.

unchanged Term applied to data blocks whose contents have not been altered since the last synchronization.

units variable Part of the `iiadm -P` command. This variable sets the number of data chunks to send before pausing for *delay variable* time to allow other applications to have a share of system resources.

update An update is a *synchronization* of the *master volume* and the *shadow volume* in which only blocks of data that have *changed* since the last synchronization are written. Also called a *fast synchronization*.

virtual shadow This term is applied to an application's view of the *shadow volume* in a dependent shadow volume set. Some blocks of data may reside on the *master volume* and some may reside on the shadow volume. To a reading or writing application, it appears as a single volume: a virtual shadow.

Index

D

E

I