



Sun StorageTek™ Availability Suite 4.0 Point-in-Time Copy Software Administration Guide

Sun Microsystems, Inc.
www.sun.com

Part No. 819-6149-10
April 2006, Revision A

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

Copyright 2006 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

This document and the product to which it pertains are distributed under licenses restricting their use, copying, distribution, and decompilation. No part of the product or of this document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, AnswerBook2, docs.sun.com, Sun StorageTek, Solstice DiskSuite, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and in other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and in other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

U.S. Government Rights—Commercial use. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2006 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, Californie 95054, États-Unis. Tous droits réservés.

Sun Microsystems, Inc. possède les droits de propriété intellectuelle relatifs à la technologie décrite dans ce document. En particulier, et sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs des brevets américains listés sur le site <http://www.sun.com/patents>, un ou les plusieurs brevets supplémentaires ainsi que les demandes de brevet en attente aux États-Unis et dans d'autres pays.

Ce document et le produit auquel il se rapporte sont protégés par un copyright et distribués sous licences, celles-ci en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Tout logiciel tiers, sa technologie relative aux polices de caractères, comprise, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit peuvent dériver des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux États-Unis et dans d'autres pays, licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, AnswerBook2, docs.sun.com, Sun StorageTek, Solstice DiskSuite, et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface utilisateur graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox dans la recherche et le développement du concept des interfaces utilisateur visuelles ou graphiques pour l'industrie informatique. Sun détient une licence non exclusive de Xerox sur l'interface utilisateur graphique Xerox, cette licence couvrant également les licenciés de Sun implémentant les interfaces utilisateur graphiques OPEN LOOK et se conforment en outre aux licences écrites de Sun.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DÉCLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES DANS LA LIMITE DE LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE À LA QUALITÉ MARCHANDE, À L'APTITUDE À UNE UTILISATION PARTICULIÈRE OU À L'ABSENCE DE CONTREFAÇON.



Contents

Preface xi

- 1. Introduction to the Point-in-Time Copy Software** 1
 - Uses for the Point-in-Time Copy Software 2
 - Point-in-Time Copy Software Architecture 3
 - Point-in-Time Copy Software and the Kernel 3
 - Point-in-Time Copy Software and the Data Services I/O Stack 5
 - Shadow Volume Sets 6
 - Independent Copy Operations 7
 - Creating an Independent Shadow Volume 7
 - Accessing the Independent Shadow Volume 8
 - Resynchronizing the Shadow and Master Volumes 9
 - Dependent Copy Operations 10
 - Creating a Dependent Shadow Volume 10
 - Accessing the Dependent Shadow Volume 11
 - Resynchronizing the Master Volume to the Shadow Volume 11
 - Resynchronizing the Shadow Volume to the Master Volume 11
 - Compact Dependent Shadow Volumes 12
 - Overflow Volumes for Compact Dependent Shadow Volumes 13
 - Bitmap Management 14

Multiple Shadows of a Single Master	21
Export Shadow	21
VTOC Information	22
2. Operation Considerations	23
Point-in-Time Copy Operations	24
Understanding System Startup and Shutdown	25
Quiescing a Master Volume	26
Point-in-Time Copies of Mounted Volumes	27
How the Delay Units Function Affects Volume Copy Operations	28
Export, Import, and Join of Dual-Ported or SAN-Accessible Shadow Volumes	29
Grouping of Volume Sets	30
Data Services Logging File	32
Operational Notes	33
Cautions: Enable, Copy, and Update Operations	33
Using the CLI for Copy and Update Operations	33
Length of Volume Names	33
Shadowing the Root File System	33
Shadowing Encapsulated Volumes	34
Interaction With <code>svadm</code>	34
Creating and Configuring Sun StorageTek Volume Sets	34
3. Configuration Considerations	35
Using the Point-in-Time Copy Software With the Remote Mirror Software	36
Interaction in the Sun StorageTek Data Service I/O Stack	37
Using a Point-in-Time Copy Shadow Volume Set to Back Up a Remote Mirror Resync	38
Using Point-in-Time Copy Volume Sizing With the Remote Mirror Software	38
Point-in-Time Copy Software in a Sun Cluster 3.1/3.2 Environment	39

Failovers	39
Additional Performance Considerations	40
4. Using the Point-in-Time Copy Software	41
Task Summary: Getting Started	42
Increasing the Storage Volume Limit	43
▼ To Increase the Storage Volume Limit	43
Sizing Requirements for Volume Sets	44
Enabling Volume Sets	45
▼ To Enable an Independent Volume Set	45
▼ To Enable a Dependent Volume Set	46
Using Overflow Volumes	47
▼ To Initialize an Overflow Volume	47
▼ To Attach Overflow Volumes to Shadow Volume Sets	48
▼ To Detach Overflow Volumes From Shadow Volume Sets	49
Moving Shadow Volume Sets	50
▼ To Move Shadow Volume Sets Into I/O Groups	50
Setting Copy Parameters	51
▼ To Get and Set Copy Parameters	51
Updating, Copying, and Waiting Operations	52
▼ To Update the Shadow Volume From the Master Volume	53
▼ To Update the Master Volume From the Shadow Volume	53
Performing Full Copy Operations	54
▼ To Copy All Data From the Master Volume to the Shadow Volume	54
▼ To Copy All Data From the Shadow Volume to the Master Volume	54
▼ To Wait for Completion of Copy or Update	55
PID Locking a Shadow Volume Set	56
Locking a Shadow Volume Set	56
Unlocking the PID on a Shadow Volume Set	57

Listing Shadow Volume Sets and Volumes	58
▼ To List Shadow Volume Sets	58
▼ To List Overflow Volumes	58
▼ To List I/O Groups	59
Displaying Status	60
▼ To Display Shadow Volume Set Status	60
▼ To Display Overflow Volume Status	60
Aborting Copy or Update Commands	61
▼ To Abort a Copy or an Update Operation	61
Resetting	62
▼ To Reset Volume Sets	62
▼ To Reset the Master Volume of Multiple Shadow Volumes	63
Disabling a Shadow Volume Set	65
▼ To Disable a Shadow Volume Set	65
▼ To Disable an Independent Shadow Volume Set During a Copy or Update Operation	66
Exporting, Importing, and Joining Shadows in a Standalone Environment	67
▼ To Export, Import, and Join a Shadow Volume	68
Exporting, Importing, and Joining Shadows in a Sun Cluster OE	70
Configuring a Point-in-Time Copy Set in a Sun Cluster OE	70
▼ To Configure a Point-in-Time Copy Set in a Sun Cluster OE	71
Point-in-Time Copy Set Considerations in a Sun Cluster OE	76
Redundancy	76
Implicit Join Operation	76
Incomplete Export / Import / Join Sequence	77
Displaying Help and Software Version Numbers	77
▼ To Display Help Information	77
▼ To Display Software Version Numbers	78
dsbitmap Bitmap Sizing Utility	79

dsbitmap 79
Syntax 79
Usage for dsbitmap 80
Examples for dsbitmap 80

5. Command-Line Interface 81

Command Modifiers 81

iiadm -g 82

Syntax 82

iiadm -C 83

Syntax 83

-n Option to the -c and -u Command Options 83

Syntax 83

Command Options 84

Help Information 85

Configuration Parameters 86

ii_bitmap 86

ii_throttle_units 86

ii_throttle_delay 86

ii_debug 86

Option List 87

To Abort a Copy or Update Operation 87

iiadm -a 87

Syntax 87

To Attach an Overflow Volume 88

iiadm -A 88

Syntax 88

To Copy Volumes 89

iiadm -c 89

Syntax	89
Options	89
To Disable Volume Sets	90
<i>iiadm -d</i>	90
Syntax	90
To Detach an Overflow Volume	91
<i>iiadm -D</i>	91
Syntax	91
To Enable Volume Sets	92
<i>iiadm -e</i>	92
Syntax	92
Options	93
To Export a Shadow Volume	94
<i>iiadm -E</i>	94
Syntax	94
To Display Command Usage	95
<i>iiadm -h</i>	95
Syntax	95
To Display Volume Set Status	96
<i>iiadm -i</i>	96
Syntax	96
Options	96
To Import a Shadow Volume	97
<i>iiadm -I</i>	97
Syntax	97
To Join a Shadow Volume	98
<i>iiadm -J</i>	98
Syntax	98

Options	98
To List Volume Sets	99
<i>iiadm -l</i>	99
Syntax	99
To List Overflow Volumes or Groups	100
<i>iiadm -L</i>	100
Syntax	100
To Move a Volume Set	101
<i>iiadm -m</i>	101
Syntax	101
To Initialize an Overflow Volume	102
<i>iiadm -O</i>	102
Syntax	102
To Set Copy Parameters	103
<i>iiadm -P</i>	103
Syntax	103
Options	103
To Display Overflow Volume Status	105
<i>iiadm -Q</i>	105
Syntax	105
To Reset a Volume Set or I/O Volume Group	106
<i>iiadm -R</i>	106
Syntax	106
▼ To Reset a Master Volume With Multiple Shadows	106
To Update a Volume Set	108
<i>iiadm -u</i>	108
Syntax	108
Options	108

To Print Software Versions 109

`iiadm -v` 109

Syntax 109

To Wait for a Copy or Update Operation to Complete 110

`iiadm -w` 110

Syntax 110

A. dsstat and Storage Cache Statistics 111

Syntax 111

Examples 114

Report Cache Statistics 114

Report Master and Shadow Volume Point-in-Time Copy Software
Statistics 115

Report Network Statistics for the Remote Mirror Software 116

Glossary 119

Index i

Index xiv

Preface

The *Sun StorageTek Availability Suite 4.0 Point-in-Time Copy Software Administration 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.

The Sun StorageTek Availability Suite 4.0 Point-in-Time Copy software is a *point-in-time snapshot* facility that runs in a Solaris OS. 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.

Before You Read This Book

In order to fully use the information in this document, you must have thorough knowledge of the Solaris™ Operating System (Solaris OS).

How This Book Is Organized

This book contains the following chapters:

[Chapter 1](#) describes the Sun StorageTek Availability Suite 4.0 Point-in-Time Copy software in functional detail.

[Chapter 2](#) discusses various operational considerations for the Sun StorageTek Availability Suite 4.0 Point-in-Time Copy software.

[Chapter 3](#) discusses configuration considerations when using Sun StorageTek Availability Suite 4.0 Point-in-Time Copy software with other data services software.

[Chapter 4](#) 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 5](#) describes the Point-in-Time Copy software's CLI and provides the syntax of the CLI commands.

[Appendix A](#) describes usage of the `dsstat` command, which enables you to track the I/O statistics of the Point-in-Time Copy software.

[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 the following for this information:

- Software documentation that you received with your system
- Solaris Operating System documentation, which is at <http://docs.sun.com>

Typographic Conventions

Typeface*	Meaning	Examples
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>% You have mail.</code>
AaBbCc123	What you type, when contrasted with on-screen computer output	<code>% su</code> <code>password:</code>
<i>AaBbCc123</i>	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be superuser to do this. To delete a file, type <code>rm filename</code> .
[]	In syntax, brackets indicate that an argument is optional.	<code>scmadm [-d sec] [-r n[:n][,n]...] [-z]</code>
{arg arg}	In syntax, braces and pipes indicate that one of the arguments must be specified.	<code>sndradm -R b {p s}</code>

* The settings on your browser might differ from these settings.

Shell Prompts

Shell	Prompt
C shell	<i>machine-name%</i>
C shell superuser	<i>machine-name#</i>
Bourne shell and Korn shell	\$
Bourne shell and Korn shell superuser	#

Related Documentation

Application	Title	Part Number
Latest release information	<i>Sun StorageTek Availability Suite 4.0 Software Release Notes</i>	819-6152
Installation and Configuration	<i>Sun StorageTek Availability Suite 4.0 Software Installation and Configuration Guide</i>	819-6147
Troubleshooting	<i>Sun StorageTek Availability Suite 4.0 Software Troubleshooting Guide</i>	819-6151
Remote Mirror software system administration	<i>Sun StorageTek Availability Suite 4.0 Remote Mirror Software Administration Guide</i>	819-6148
Sun cluster usage	<i>Sun Cluster and Sun StorageTek Availability Suite 4.0 Software Integration Guide</i>	819-6150

Accessing Sun Documentation

You can view, print, or purchase a broad selection of Sun documentation, including localized versions, at:

<http://www.sun.com/documentation>

Third-Party Web Sites

Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Contacting Sun Technical Support

If you have technical questions about this product that are not answered in this document, go to:

<http://www.sun.com/service/contacting>

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can submit your comments by going to:

<http://www.sun.com/hwdocs/feedback>

Please include the title and part number of your document with your feedback:

Sun StorageTek Availability Suite 4.0 Point-in-Time Copy Software Administration Guide,
part number 819-6149-10

Introduction to the Point-in-Time Copy Software

This chapter describes the Sun StorageTek Availability Suite 4.0 Point-in-Time Copy software in functional detail. First, the uses for the software are described, and then the software's architecture is explained. The chapter continues to discuss the allowable volume set configurations 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:

- ["Uses for the Point-in-Time Copy Software" on page 2](#)
- ["Point-in-Time Copy Software Architecture" on page 3](#)
- ["Shadow Volume Sets" on page 6](#)
- ["Independent Copy Operations" on page 7](#)
- ["Dependent Copy Operations" on page 10](#)
- ["Compact Dependent Shadow Volumes" on page 12](#)
- ["Overflow Volumes for Compact Dependent Shadow Volumes" on page 13](#)
- ["Bitmap Management" on page 14](#)
- ["Multiple Shadows of a Single Master" on page 21](#)
- ["Export Shadow" on page 21](#)
- ["VTOC Information" on page 22](#)

Uses for the Point-in-Time Copy Software

The Point-in-Time Copy software, running in the Solaris OS, provides applications with continuous access to volume data and offers secondary applications un-intrusive access to a point-in-time copy of the same volume data. 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 master to shadow or from shadow to master.

The Point-in-Time Copy software supports 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 Point-in-Time Copy software 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

The Sun StorageTek Availability Suite software is cluster aware in the Sun™ Cluster 3.1/3.2 environments, and provides high availability (HA).



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

Point-in-Time Copy Software Architecture

Sun StorageTek Availability Suite 4.0 Point-in-Time Copy software is a *point-in-time snapshot* facility that runs on the Solaris OS. 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. Instantly after a *point-in-time copy* is created, you have immediate read and write access to both the original master and shadow copy volumes.

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

The Point-in-Time Copy software tracks the differences between the master and shadow volumes caused by write operations from the time when the set is enabled. This capability allows the data on the two volumes to move forward in time independently of each other. Applications can access both volumes and modify the data on them independently.

Because the software tracks differences between the volumes, the volumes can be quickly updated after the first Point-in-Time copy is enabled. A fast resynchronization 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 enabled, copied, or updated on the shadow volume set, the applications using the shadow volume set can immediately resume processing. The point-in-time copy is established or re-established when the command-line interface (CLI) prompt returns or when the next shell script command is read.

Point-in-Time Copy Software and the Kernel

Sun StorageTek data services are implemented as layered pseudo-drivers in the Solaris kernel I/O stack. These drivers rely on the `nsctl` (Network StorageTek Control) framework to support this layering and provide runtime control. Point-in-Time Copy software is implemented as an `nsctl` I/O filter module, which enables integration with other Sun StorageTek data services (specifically Remote Mirror Copy software). The architecture of the Point-in-Time Copy software in the kernel I/O stack is shown in [FIGURE 1-1](#).

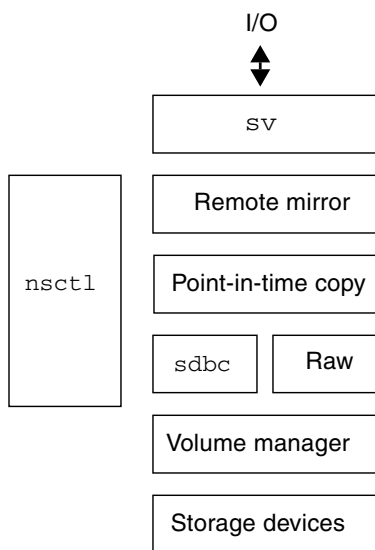


FIGURE 1-1 Point-in-Time Copy Software in the Sun StorageTek Services I/O stack.

The Point-in-Time Copy software works by being in the Solaris I/O data path. I/O commands and data enter and exit the Point-in-Time Copy software through the Sun StorageTek storage volume (*sv*) software. Mediated by *nsctl*, the data optionally flows through the Remote Mirror software and the Point-in-Time Copy software to the storage device block cache (*sdbc*) drivers and then to its destination either on the storage device (for write operations) or in application or kernel memory (for read operations).

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 redundant array of independent disks (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 that access 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 and move it to its destination on the storage device.

The I/O commands targeted to shadow volume sets are intercepted by the `sv` driver and routed through the Sun StorageTek I/O stack prior to being passed on to the storage device driver or the volume manager. The `sv` layer is a very thin layer in the Solaris I/O stack and operates by interposing itself 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 StorageTek service I/O stack. The `sv` layer routes them through the Sun StorageTek 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 device back to user space.

Shadow Volume Sets

The master volume of a shadow volume set is the volume from which you intend to create a point-in-time copy. The master volume is the source of the data that is 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*. 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. A complete duplicate of the master volume is started on the independent shadow volume when the point-in-time copy is initiated.

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 Point-in-Time Copy 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 full synchronization has been completed.

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 that have not been written since the point-in-time copy was established.

Additional details of how independent and dependent shadow volume sets behave is detailed in [“Independent Copy Operations” on page 7](#) and [“Dependent Copy Operations” on page 10](#). For details about compact dependent shadow volumes, see [“Compact Dependent Shadow Volumes” on page 12](#).



Caution – When creating shadow volume sets, do not create shadow or bitmap volumes using partitions that include cylinder 0. Data loss might occur. See [“VTOC Information” on page 22](#).

Independent Copy Operations

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

1. The data at the block that 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, the software skips over this block. In this way an independent copy is established on the shadow volume.

Once the background copy is completed, 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 been completed on a shadow volume set by copying only those blocks that have been modified since the full copy. Update copies are described in [“Resynchronizing the Shadow and Master Volumes” on page 9](#).

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 or SAN-Accessible 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. The shadow volume can then be 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 one of the first two approaches is taken, bitmap management continues, which allows:

- In the case of an exported shadow volume, a later *join* of the exported shadow with its original shadow volume set

- In the case where Point-in-Time Copy software control is maintained, a later update, or fast, copy can be performed

Joins are explained in [“Export, Import, and Join of Dual-Ported or SAN-Accessible Shadow Volumes” on page 29](#). *Update* point-in-time copies are explained in [“Resynchronizing the Shadow and Master Volumes” on page 9](#).

If an independent shadow volume is accessed by another host with the export and import commands, a secondary 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 still 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 information about where the modification originated is available.

Resynchronizing the Shadow and Master Volumes

The term *resynchronization* is used to describe a synchronization that occurs between volumes in a shadow volume set that were 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 7](#).

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

Dependent Copy Operations

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

- The proportion of data changed between taking Point-in-Time Copies is infrequent.
- The overhead of accessing the shadow volume through the master volume is not prohibitive in terms of performance for the master or the shadow.
- The overhead in I/O CPU and bandwidth use engendered by 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, the bitmap volume begins tracking changes made on the master volume. Enabling a shadow volume set with a dependent shadow volume does not initiate a background synchronization process. All data that has remained unmodified on the master volume since the point-in-time copy was created is accessed on the master volume itself.

1. 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.
2. 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.
3. Then, the new block data is written to the master volume, and the associated bit in the bitmap volume is marked as 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 cannot be accessed if the master volume and bitmap volume are not available.

Accessing the Dependent Shadow Volume

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. 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 the bitmap to determine if the data has been modified. If it hasn't, data from the block that 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 that 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. 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. Resynchronization 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 an update synchronization. In an update synchronization, only the blocks marked as changed (with a bitmap value of 1) are copied to the target of the copy. In the case of a dependent shadow volume, this includes 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 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 percent, or, more specifically, the percentage difference in size between the master and shadow volumes (see below).
- 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 between planned point-in-time copies. For many applications, entire areas of storage are modified rarely relative to their neighbors.

For example, if you know that at most 10 percent of the blocks on the master volume are changed between point-in-time copies, you can allocate a compact dependent shadow volume at 10 percent of the size of the master volume.

1. The Point-in-Time Copy software keeps track of the updated data blocks using an index in the bitmap.
2. Blocks written to the master are first copied to the next available block in the compact dependent shadow volume.
3. 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 shadow volume fails. To protect against such a failure, you can designate *overflow volumes* for a compact dependent shadow volume set.

Note – If a compact dependent shadow volume set overflows due to size or due to an unexpectedly large volume of write operations, 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 enables you to recover data. *However, any subsequent write operations will force the shadow volume offline.*

Overflow Volumes for Compact Dependent Shadow Volumes

You can designate an overflow volume for one or more compact dependent shadow volumes. If a compact dependent shadow volume exceeds its limits (that is, 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 volume, or a block on the overflow volume.

When a volume is initialized as an overflow volume, information is written to a header area on the volume that the Point-in-Time Copy software uses to track how the volume is being used. For example, an overflow volume keeps track of the number of dependent shadow volumes that 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 to create point-in-time copies. For every 32 Kbyte block of a master volume that is part of a shadow volume set, a bit is maintained that 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*, 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 after a point-in-time copy has been established. In the figure, each 32 Kbyte block on the master and shadow volumes is represented by a cell. The contents of the cell (for example, AAA) represents the data in the 32 Kbyte blocks on the volume. For each block that differs from the master since the point-in-time copy was established, a bit in the bitmap volume is set to 1. This indicates that the data on the storage has changed since the point-in-time copy was made.

Master	Physical shadow	Bitmap
AAA	AAA	0
BBB	BBB	0
CCC	Nnn	1
DDD	DDD	0
EEE	EEE	0
FFF	FFF	0
GGG	Mmm	1
HHH	HHH	0
III	III	0
JJJ	JJJ	0

FIGURE 1-2 Independent Shadow Volume Set After Creation of 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 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 as changed (1) in the bitmap.

Master	Virtual shadow	Physical shadow	Bitmap
AAA	AAA	???	0
BBB	BBB	???	0
CCC	Nnn	Nnn	1
DDD	DDD	???	0
EEE	EEE	???	0
FFF	FFF	???	0
GGG	Mmm	Mmm	1
HHH	HHH	???	0
III	III	???	0
JJJ	JJJ	???	0

FIGURE 1-3 Dependent Shadow Volume Set After Creation of 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 that 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 percent.
- Storage conservation is important.
- A dependent shadow is suitable.

With compact dependent shadow volumes, an index is maintained for every changed block being tracked in the bitmap volume. It is an index to the block in the compact volume of the data as it existed when 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. 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 taken off line.
- The source volume becomes the only valid copy of the volume.

To prevent this occurrence, you can designate overflow volumes 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 after a point-in-time copy has been established.

Master	Virtual shadow	Physical shadow/block#		Bitmap/index	
AAA	AAA	Mmm	0	0	-
BBB	BBB	Nnn	1	0	-
CCC	Nnn	???	2	1	1
DDD	DDD	???	3	0	-
EEE	EEE	???	4	0	-
FFF	FFF			0	-
GGG	Mmm			1	0
HHH	HHH			0	-
III	III			0	-
JJJ	JJJ			0	-

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

To prevent the problems 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 is filled up, the following occurs:

- The I/O fails.
- The target volume is taken off line.
- 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 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 – A single compact dependent shadow volume can be configured to use only one overflow volume. Many single compact dependent shadow volumes can have overflow volumes configured on the same physical volume,



Caution – Do not create bitmaps on cylinder 0 because the Point-in-Time Copy software does a raw write and destroys the virtual table of contents (VTOC) for that device.

Master	Virtual shadow	Physical shadow	Overflow		Bitmap / index	
ZZZ	AAA	Mmm	0	Header	1	2
BBB	Qqq	Nnn	1	Rrr	1	[2]
CCC	Nnn	AAA	2	Qqq	1	1
DDD	DDD	Yyy	3	???	0	-
EEE	EEE	Vvv	4	???	0	-
FFF	Yyy		5	???	1	3
GGG	Mmm		6	???	1	0
HHH	Vvv		7	???	1	4
III	Rrr		8	???	1	[1]
JJJ	JJJ		N	???	0	-

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

Multiple Shadows of a Single Master

Sun StorageTek Availability Suite 4.0 Point-in-Time Copy software enables creation of multiple point-in-time copies 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 own unique 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. For the shadow to be exported, it must reside on a dual-ported or SAN-accessible 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 performed by a host that is separate from the master volume's host.

VTOC Information

The Solaris system administrator must be knowledgeable about the virtual table of contents (VTOC) that is created on raw devices by the Solaris operating system.

The creation and updating of a physical disk's VTOC is a standard function of the Solaris operating system. Software applications like the Sun StorageTek Availability Suite, the growth of storage virtualization, and the appearance of SAN-based controllers have made it easy for an uninformed Solaris system administrator to inadvertently allow a VTOC to be altered. Altering the VTOC increases the possibility of data loss.

Remember these points about the VTOC:

- A VTOC is a software-generated virtual table of contents based on the geometry of a device and written to the first cylinder of that device by the Solaris `format(1M)` utility.
- Various software components such as `dd(1M)`, backup utilities, Point-in-Time Copy software, and Remote Mirror software can copy the VTOC of one volume to another volume if that volume includes cylinder 0 in its mapping.
- If the VTOC of the source and destination volumes are not 100 percent identical, some type of data loss might occur.

This data loss might not be initially detectable, but can be detected later when other utilities are used, like `fsck(1M)`.

When first configuring and validating volume replication, save copies of all affected device's VTOCs using the `prtvtoc(1M)` utility. The `fmthard(1M)` utility can be used to restore them later, if necessary.

- When using volume managers like VxVM and SVM, copying between individual volumes created under these volume managers is safe. VTOC issues are avoided because the VTOC is excluded from volumes created by these volume managers.
- When formatting individual partitions on a raw device, for all partitions except the backup partition, make sure the partitions do not map cylinder 0, which contains the VTOC. When using raw partitions as volumes, you are the volume manager and you need to exclude the VTOC from partitions that you configure.
- When formatting the backup partition of a raw device, make sure that the physical geometries of the source and destination devices are identical. (Partition 2, by default, maps all cylinders under the backup partition.) If identical device sizing is not possible, make sure that the source backup partition is smaller than the destination partition and that the destination partition does not map cylinder 0.

Operation Considerations

This chapter consists of the following main topics:

- [“Point-in-Time Copy Operations” on page 24](#)
- [“Understanding System Startup and Shutdown” on page 25](#)
- [“Quiescing a Master Volume” on page 26](#)
- [“Point-in-Time Copies of Mounted Volumes” on page 27](#)
- [“How the Delay Units Function Affects Volume Copy Operations” on page 28](#)
- [“Export, Import, and Join of Dual-Ported or SAN-Accessible Shadow Volumes” on page 29](#)
- [“Grouping of Volume Sets” on page 30](#)
- [“Data Services Logging File” on page 32](#)
- [“Operational Notes” on page 33](#)

Note – Sun suggests that the same disk management method (either software volume management or raw disk partitioning) be used with the Remote Mirror software and the Point-in-Time Copy software on the primary site and the secondary site volumes.

The Sun StorageTek Availability Suite software replicates data at the block level, and the block counts for a given size are different for a disk slice and a volume. Due to this difference, a remote mirror replication or reverse sync, or a point-in-time full independent copy or reverse copy, might fail if the target size is smaller than the source size.

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 examples use these volume names:

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

All commands are accessed on the following path:

```
/usr/sbin/iiadm
```

Point-in-Time Copy software operations include, but are not limited to, these tasks:

- Enabling shadow volume sets for selected volumes.
- Using fast resynchronization, or update, commands to quickly update volumes when necessary.
- Using the copy command to perform full volume copies.
- You typically perform a full volume copy only when 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.
- Disabling shadow volume sets, if you choose to do so. You can keep shadow volume sets enabled indefinitely.

Note – If a remote mirror volume is a *target* of Point-in-Time Copy update or copy, 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.

Understanding System Startup and Shutdown

Point-in-Time Copy makes use of the Service Management Facility (SMF) for the startup and shutdown of its software. SMF provides an infrastructure that augments the traditional Unix startup scripts, `init` run levels, and configuration files. The traditional run-level control script, which started and stopped Point-in-Time Copy software, has been replaced with an SMF service that is enabled during system startup and disabled at system shutdown.

When the Point-in-Time service is brought online during system startup, the volumes of previously configured shadow volume sets are resumed. During shutdown, the Point-in-Time service is brought offline, suspending the volumes of previously configured shadow volume sets.

Note – The commands for suspending and resuming shadow volume sets are not made available to the user. The sets may be suspended and resumed, however, by disabling and enabling the Point-in-Time Copy SMF service using the `dscfgadm` utility. For more information, refer to the *Sun StorageTek Availability Suite 4.0 Software Installation and Configuration Guide*.

Unlike the run-control scripts, which were started sequentially, many SMF services may start concurrently during system startup, resulting in decreased boot times. To ensure that the Point-in-Time Copy SMF service is started at the appropriate time in the boot process, its service description specifies formalized dependency relationships with other services.

The Availability Suite services under SMF control and the current state is shown below. See `man dscfgadm` for additional information.

```
# dscfgadm -i
SERVICE          STATE             ENABLED
nws_scm           online           true
nws_sv            online           true
nws_ii            online           true
nws_rdc           online           true
nws_rdcsyncd     online           true

Availability Suite Configuration:
Local configuration database: valid
cluster configuration database: valid
cluster configuration location: /dev/did/rdisk/d16s7
```

Quiescing a Master Volume

In Solaris operating systems, 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 be 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 be unwritable).

Most file systems, databases, and applications create or update an item on a disk in an I/O operation that involves 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 or 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 might cause partial data to be rolled back or intentionally discarded.

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

Therefore, a point-in-time copy or update should not be performed against a source volume (master or shadow) without quiescing or stopping all application I/O and flushing any file system caches, such as `lockfs(1M)`, associated with the volume. This quiesced or stopped state is required only 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 using a logging file system, or in Oracle®, that 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 StorageTek Availability Suite 4.0 Point-in-Time Copy software provides an attractive complement to a hot-backup facility. Instead of putting the database in hot backup mode for the minutes or hours needed to perform disk or tape replication, the Point-in-Time Copy software requires this mode for only milliseconds or seconds.

Point-in-Time Copies of Mounted Volumes

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 in a 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 the shadow volume, by design, is not.

When a point-in-time copy is established in this way, when the target volume is first mounted, the software detects that a currently dismounted volume has mounted metadata. The software 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” on page 26](#)).

The target of a point-in-time copy operation, which is usually the shadow volume, must not be mounted. If the target is mounted, the application accessing the target volume will read inconsistent and changing data.

Note – If a remote mirror volume is a *target* of Point-in-Time Copy update or copy, 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 the set is not in logging mode, 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 operation, 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 Operations” on page 10](#)). 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 types of variables: the *units* variables and the *delay* variables. Copy units are expressed in 32 Kbyte chunks (for example: 50 Mbyte = 1600); delay units are expressed in milliseconds. The loop performs the specified copy units worth of I/O, and then sleeps for copy delay milliseconds, until the synchronization is complete.

By adjusting the copy units and copy delay values, a system administrator can tune the impact that background synchronization imposes on the system. 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 103](#).

Export, Import, and Join of Dual-Ported or SAN-Accessible 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 enable retention of point-in-time copy information 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 Point-in-Time 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 the volume 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. The new set includes the exported shadow volume as its shadow volume. A new bitmap volume is on the partner host. While enabled, any write operations 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 using an OR operation to compare 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 contains only zeros, and this OR operation leaves the bitmap on the original host unchanged. The bitmap volume from the partner host is no longer needed after the join command is complete, and the volume can be reused.

If a write did occur on the partner host, the OR operation sets the bitmap for the associated block to 1 (or changed). Using the bitmap volume, you can now create a point-in-time copy using the update copy.

See [“Exporting, Importing, and Joining Shadows in a Standalone Environment” on page 67](#) and [“Exporting, Importing, and Joining Shadows in a Sun Cluster OE” on page 70](#) for details.

Grouping of Volume Sets

The Point-in-Time Copy software enables you to place shadow volume sets into I/O groups.

Groups are helpful in administering multiple volume sets in much the same way that a script would be. With an I/O group, a single CLI command can be executed on every member of the group.

I/O groups enable 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 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.

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

[TABLE 2-1](#) is a usage summary for options that support grouping (-g g). The legend for the usage summary is as follows:

ind - independent volume set

dep - dependent volume set

all - all configured volumes

m - master volume

s - shadow volume

v - shadow volume (reference name)

o - overflow volume

b - bitmap volume

TABLE 2-1 Usage Summary for Options That Support Grouping

Option	Description
-g g -e ind m s b	Group enable independent master shadow bitmap
-g g -e dep m s b	Group enable dependent master shadow bitmap
-g g -d	Disable group
-g g -u s	Update shadow for all volumes in group
-g g -u m	Update master for all volumes in group
-g g -c s	Copy to shadow for all volumes in group
-g g -c m	Copy to master for all volumes in group
-g g -a	Abort copy for all volumes in group
-g g -w	Wait for all volumes in group
-g g -i	Display status of all volumes in group
-g g -l	List all volumes in group
-g g -L	List all groups
-g g -m v v	Move one or more volumes into group
-g "" -m v	Remove volume from group
-g g -R	Reset all volumes in group
-g g -A o	Attach overflow to all volumes in group
-g g -D	Detach overflow from all volumes in group
-g g -E	Export shadow volume for all volumes in group
-g g -P d u	Set copy delay and units for all volumes in group
-g g -P	Get copy delay and units for all volumes in group

Data Services Logging File

The Sun StorageTek Availability Suite 4.0 Point-in-Time Copy software, like all data services software, generates entries in the data services log file, `/var/adm/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 logging 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 with Point-in-Time Copy software messages is shown here.

```
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/rdisk/rootdg/ii_10mb_0
Feb 06 16:15:16 ii: Enabled /dev/vx/rdisk/rootdg/ii_10mb_0
/dev/vx/rdisk/rootdg/ii_1mb_0 /dev/vx/rdisk/rootdg/ii_mb_0 (dependent)
Feb 06 16:15:17 sv: enabled /dev/vx/rdisk/rootdg/ii_1mb_0
Feb 07 08:14:43 ii: Disabled /dev/vx/rdisk/rootdg/ii_1mb_0
Feb 07 08:15:05 sv: enabled /dev/vx/rdisk/rootdg/ii_10mb_0
Feb 07 08:15:05 ii: Enabled /dev/vx/rdisk/rootdg/ii_10mb_0
/dev/vx/rdisk/rootdg/ii_1mb_0 /dev/vx/rdisk/rootdg/ii_mb_0 (dependent)
Feb 07 08:15:05 sv: enabled /dev/vx/rdisk/rootdg/ii_1mb_0
Feb 07 08:15:19 ii: Create overflow succeeded /dev/vx/rdisk/rootdg/ii_9mb_0
Feb 07 08:15:28 ii: Attach /dev/vx/rdisk/rootdg/ii_1mb_0
/dev/vx/rdisk/rootdg/ii_9mb_0
Feb 07 08:19:59 ii: Start update /dev/vx/rdisk/rootdg/ii_1mb_0 to shadow
Feb 07 08:20:02 ii: Finish update /dev/vx/rdisk/rootdg/ii_1mb_0 to shadow
Feb 07 08:21:21 ii: Disabled /dev/vx/rdisk/rootdg/ii_1mb_0
Feb 07 08:21:27 sv: enabled /dev/vx/rdisk/rootdg/ii_10mb_0
Feb 07 08:21:27 ii: Enabled /dev/vx/rdisk/rootdg/ii_10mb_0
/dev/vx/rdisk/rootdg/ii_1mb_0 /dev/vx/rdisk/rootdg/ii_mb_0 (dependent)
Feb 07 08:21:27 sv: enabled /dev/vx/rdisk/rootdg/ii_1mb_0
Feb 07 08:21:38 ii: Attach /dev/vx/rdisk/rootdg/ii_1mb_0
/dev/vx/rdisk/rootdg/ii_9mb_0
Feb 07 08:22:42 ii: Disabled /dev/vx/rdisk/rootdg/ii_1mb_0
```

Operational Notes

Following are some operational notes.

Cautions: Enable, Copy, and Update Operations

Keep the following cautionary items in mind when performing an enable, copy, or update operation:

- Although these operations are checked for Solaris mounted file systems, do not perform enable, copy, or update operations on other types of shadow volumes. To prevent a system panic from occurring, you must unmount the shadow volume before invoking an enable, copy, or update operation.
- 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 copy is established or reestablished, (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. 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 shadow volume group when using the copy or update commands.

Length of Volume Names

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

Shadowing the Root File System

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

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. Using the `iiadm -d` command option to disable a volume set automatically removes 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.

Creating and Configuring Sun StorageTek Volume Sets

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
- Performing any operation that changes the Sun StorageTek services and related shadow volume set configuration



Caution – When creating shadow volume sets, do not create shadow or bitmap volumes using partitions that include cylinder 0. Data loss might occur. See [“VTOC Information” on page 22](#).

When configuring a volume set, do not use the same volume set as a point-in-time copy shadow volume and as a remote mirror secondary volume. If you attempt to configure a volume set for two purposes, the data contained on the volume might not be valid for the application that accesses the volume.

Configuration Considerations

This chapter discusses Point-in-Time Copy software configuration issues.

This chapter covers the following topics:

- [“Using the Point-in-Time Copy Software With the Remote Mirror Software” on page 36](#)
- [“Point-in-Time Copy Software in a Sun Cluster 3.1/3.2 Environment” on page 39](#)
- [“Additional Performance Considerations” on page 40](#)

Using the Point-in-Time Copy Software With the Remote Mirror Software

Sun StorageTek Availability Suite 4.0 Remote Mirror software enables 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. After a synchronization has been performed, the primary and secondary remote mirror volumes are remotely replicated. This means that the two volumes are kept up-to-date with each other. For additional information, see [“Related Documentation” on page xiv](#). A few of the highlights of using Point-in-Time Copy software and Remote Mirror software together are described in this section.

To ensure the highest level of data integrity and system performance on both sites during normal operations, use Point-in-Time Copy software in conjunction with Remote Mirror software.

When using the Point-in-Time Copy software with the 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 on 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 can 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 is reduced if the point-in-time copy is mirrored instead. Keeping the secondary site slightly out of sync with the primary site also enables verification of the primary data prior to replicating the data on the secondary site.

Using Point-in-Time Copy software with Remote Mirror software, you can create a point-in-time copy of a remote mirror secondary volume before beginning synchronization of the secondary volume from the primary site. Protection against double failure is provided by the point-in-time copy of the replicated data. If a subsequent failure occurs during resynchronization, the point-in-time copy can be used as a fallback position. 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) at the secondary site.

Interaction in the Sun StorageTek Data Service I/O Stack

The I/O operation that the Point-in-Time Copy software performs 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 storage volume (SV) layer. If the shadow volume is also a remote mirror volume, the Remote Mirror software does not see these I/O operations. In this situation, the data modified by the I/O is not replicated on 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 accepts the bitmap. If the bitmap is accepted, the Remote Mirror software 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 rejects the bitmap from the Point-in-Time Copy software. This, in turn, causes the enable, copy, or update operation to fail. Once remote mirror logging has been reenabled, the Point-in-Time Copy software operation can be reissued.

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 shadow-to-master update.



Caution – 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.



Caution – It is possible to set up configurations in which data loss can occur, particularly in both multihop remote mirror replication and remote mirror disaster recovery scenarios. The system administrator must assure that this does not happen. See the *Sun StorageTek Availability Suite 4.0 Remote Mirror Software Administration Guide* for further information.

Using Point-in-Time Copy Volume Sizing With the Remote Mirror Software

In a point-in-time copy snapshot, the shadow volume is made to look exactly like the master volume, including matching the number of blocks. If the master volume is larger or smaller than the shadow volume, the shadow volume appears to be resized at the moment the snapshot is taken. Physically, the shadow has not changed size, but the point-in-time copy kernel module always reports that the shadow is the same size as the master.

This can present several problems with the Remote Mirror software, which does not expect the size of the volume to change:

- If you plan to use a remote mirror primary or secondary volume as a point-in-time copy shadow volume, the master and the shadow of that point-in-time copy volume set must be exactly the same size. If they are different, you see an error when attempting to enable the point-in-time copy volume set with the `iiadm -e` command.

- You might also see this error during a `disable iiaadm -d` command. If the shadow volume of the existing point-in-time copy volume set is not the same size as the master volume, and a remote mirror volume set was enabled using that shadow volume as the primary or secondary, the Point-in-Time Copy software stops you from performing the disable operation.

Point-in-Time Copy Software in a Sun Cluster 3.1/3.2 Environment

Point-in-Time Copy software volumes can be hosted in the Sun Cluster 3.1 environment and the Sun Cluster 3.2 environment. Clustering allows point-in-time copy replications to fail over, or continue unaffected, if a node hosting a point-in-time copy shadow volume set 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 the Sun Cluster environment as part of its control of volume management.

A successful Point-in-Time Copy software fail over requires proper configuration of the shadow volume sets in a Sun Cluster resource group. A resource group is a grouping of items in a Sun Cluster that are interrelated in such a way as to make it impossible to fail over 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 fails over. Detailed information about resource groups is available in the Sun Cluster documentation.

Failovers

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 that were running on the failing node on another node, without losing any information. This information is application dependent and outside the control of the Sun Cluster environment. The environment 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, control of the volumes being referenced (master volume, shadow volume, and bitmap volume) must be moved to the new node. Then operations restart at the point where they left off.

The Point-in-Time Copy software must be configured such that the master volume, the bitmap volume, and the overflow volume are part of the same volume manager device group. All members of the device group must be available at the point at which the Point-in-Time Copy software comes up in the boot sequence. The shadow volume can be in a different device group, in order to afford the use of export, import, and join in a Sun Cluster.

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 StorageTek 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 environment must be disk-based, not memory-based.

Additional Performance Considerations

Consider the following when configuring a system for use with Point-in-Time Copy software:

- Configure to avoid hot spots. Do not put multiple volumes on a single spindle.
- Disk speeds of both the master volume and the shadow volume 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.
- If the Sun StorageTek Availability Suite software is used with a file system, tuning the number of *SV* threads often produces better performance.

Several performance considerations exist for the Sun StorageTek Availability Suite 4.0 Point-in-Time Copy software and this list is not comprehensive.

When a filesystem flushes its cache, it generates many parallel writes. *SV*'s default setting of 32 threads could produce a bottleneck. The maximum number of threads allowed is 1024.

Note – Each thread consumes 32k of memory.

The tuneable, *sv_threads*, is in `/usr/kernel/drv/sv.conf`. Because *sv.conf* values are read when a module loads, changes to the *sv_threads* value will not take effect until you restart the Availability Suite services using `dscfgadm`.

Using the Point-in-Time Copy Software

This chapter explains how to 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:

- [“Task Summary: Getting Started” on page 42](#)
- [“Increasing the Storage Volume Limit” on page 43](#)
- [“Sizing Requirements for Volume Sets” on page 44](#)
- [“Enabling Volume Sets” on page 45](#)
- [“Using Overflow Volumes” on page 47](#)
- [“Moving Shadow Volume Sets” on page 50](#)
- [“Setting Copy Parameters” on page 51](#)
- [“Updating, Copying, and Waiting Operations” on page 52](#)
- [“PID Locking a Shadow Volume Set” on page 56](#)
- [“Listing Shadow Volume Sets and Volumes” on page 58](#)
- [“Displaying Status” on page 60](#)
- [“Aborting Copy or Update Commands” on page 61](#)
- [“Resetting” on page 62](#)
- [“Disabling a Shadow Volume Set” on page 65](#)
- [“Exporting, Importing, and Joining Shadows in a Standalone Environment” on page 67](#)
- [“Exporting, Importing, and Joining Shadows in a Sun Cluster OE” on page 70](#)
- [“Displaying Help and Software Version Numbers” on page 77](#)
- [“dsbi tmap Bitmap Sizing Utility” on page 79](#)

Task Summary: Getting Started

This table briefly lists the steps and tasks for getting started with the Point-in-Time Copy software.

TABLE 4-1 Getting Started Task Summary

Step	For Instructions (Command-Line Interface)
1. Choose volumes to use.	“Sizing Requirements for Volume Sets” on page 44
2. Enable shadow volume sets consisting of master volume, shadow volume, and bitmap volume.	“Enabling Volume Sets” on page 45
3. Initialize and attach overflow volumes to compact dependent shadow volume sets	“Using Overflow Volumes” on page 47
4. Collect volume sets in I/O groups for ease of management.	“To Move Shadow Volume Sets Into I/O Groups” on page 50
5. Perform copy, update, and other operations on the volume sets.	Chapter 4 , including: “Updating, Copying, and Waiting Operations” on page 52 “Performing Full Copy Operations” on page 54
6. Check volume set status.	“To Display Shadow Volume Set Status” on page 60

Increasing the Storage Volume Limit

The Sun StorageTek Availability Suite software has a default limit of 4096 storage volumes for use with the software. For example, if you use Point-in-Time Copy software only, you can have 1365 volume sets, each consisting of a master, shadow, and bitmap volume. If you use both the Remote Mirror and the Point-in-Time Copy products, the number of volume sets is shared between these two products.

If you want to increase the storage volume limit, edit the `/usr/kernel/drv/nsctl.conf` file and change the number associated with `nsc_max_devices`.

The following procedure describes how to increase the storage volume default limit.

▼ To Increase the Storage Volume Limit

1. **Log in as the root user.**
2. **Open the `/usr/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. **Restart the Availability Suite services as follows:**

```
# dscfgadm -d
# dscfgadm -e
```

Sizing Requirements for Volume Sets

Before you can enable shadow volume sets, you must create the volumes with your volume manager. The Point-in-Time Copy software supports VERITAS Volume Manager, Solaris Logical Volume Manager (SLVM) software, and raw volumes.

Shadow volume sets can be organized in I/O groups so that commands can be carried out across 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. The following are the sizing requirements for volume sets:

- A master volume can be any size.
- A shadow volume, independent or dependent, must be at least as large as the master volume.
- A compact dependent shadow volume can be smaller than the master volume but its size should easily exceed the total number of expected writes to both the master and shadow volumes during the time when the compact dependent shadow volume is in use. The Point-in-Time Copy software supports a shared overflow volume for storing changes that exceed the configured size of the compact dependent shadow volume. If the overflow volume is full, or not specified, a change cannot be recorded in the compact dependent shadow volume, forcing the volume to become inconsistent. The volume is subsequently taken offline.
- 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 Kbyte per 1 Gbyte of master volume size (rounded up to the nearest whole Gbyte), plus an additional 24 Kbyte for overhead.

For example, to shadow a 3-Gbyte master volume, the bitmap size must be $(3 \times 8 \text{ Kbyte}) + 24 \text{ Kbyte}$, or 48 Kbyte in size. A 50-Gbyte master volume requires a 424-Kbyte bitmap volume.
 - For compact dependent shadow volume sets:

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

For example, to shadow a 3-Gbyte master volume, the bitmap size must be $(3 \times 264 \text{ Kbyte}) + 24 \text{ Kbyte}$, or 816 Kbyte in size. A 50-Gbyte master volume in a compact dependent shadow volume set requires a 13224-Kbyte 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 might be wasted. If you enable a shadow volume set with a bitmap that is too small, the enable command fails with an error message. See also [“dsbitmap Bitmap Sizing Utility” on page 79](#).

Enabling Volume Sets

Use a volume for the bitmap.

When creating shadow volume sets, do not create shadow or bitmap volumes using partitions that include cylinder 0 because data loss might occur. See [“VTOC Information” on page 22](#).

A remote mirror volume set must be in logging mode (only if a remote mirror volume is a *target* of Point-in-Time Copy update/copy) 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 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 is performed. Both the master volume and the shadow volume are available at once, but the shadow volume is treated as a dependent shadow until the full copy is complete. See also [“To Enable Volume Sets” on page 92](#).

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

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

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

If the shadow volume set is enabled in a nonexistent I/O group, the I/O group is created.

▼ To Enable a Dependent Volume Set

Use the `-e dep` option to enable a dependent shadow volume set. No copy operation takes place and the volume set is available at once. See also [“To Enable Volume Sets” on page 92](#).

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

```
# iiadm -e dep /dev/rdisk/c1t3d0s0 /dev/rdisk/c1t3d0s4 /dev/rdisk/c1t2d0s5
#-- to enable a dependent volume set
# iiadm -g io-groupname -e dep /dev/rdisk/c1t3d0s0 /dev/rdisk/c1t3d0s4 \
/dev/rdisk/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 nonexistent I/O group, the I/O group is created.

Using Overflow Volumes

You can initialize an overflow volume for use with compact dependent shadow volumes, and 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 102](#).

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

```
# iiaadm -O /dev/rdisk/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, which directly performs the requested action without prompting. This option is typically used during script processing.



Caution – 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 and -u Command Options” on page 83](#).

▼ To Attach Overflow Volumes to Shadow Volume Sets

A dependent shadow volume set with a compact dependent shadow volume can have an overflow volume attached to it so that any writes to the volume set can be redirected to the overflow volume when necessary. See also [“To Attach an Overflow Volume” on page 88](#).

Note – You can attach only one overflow volume to a compact dependent shadow volume.

- Attach an overflow volume to a volume set by typing:

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

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

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

This option supports the optional `-n` parameter, which directly performs 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 102](#)), it will be initialized before it is attached. 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 procedure if you want to detach an overflow volume from a compact dependent shadow volume set. See also [“To Detach an Overflow Volume” on page 91](#).

- Detach an overflow volume from a shadow volume set by typing:

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

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

```
# 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 one or more shadow volume sets into I/O groups for ease of management. Move a shadow volume set out of any I/O group by typing two quotation marks instead of the I/O group name. See also [“To Move a Volume Set” on page 101](#). If you move the shadow volume set into a nonexistent I/O group, the Point-in-Time Copy software creates that I/O group.

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

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

- Move multiple shadow volume sets into an I/O group by typing:

```
# iiadm -g io-groupname -m /dev/rdisk/c1t3d0s4 /dev/rdisk/c1t4d0s4
```

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

```
# iiadm -g "" -m /dev/rdisk/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.

▼ To Get and Set Copy Parameters

Use the `-P` option to set copy parameters after you have enabled your shadow volume sets and collected 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, *delay*, is the length of time in system clock ticks (2 through 10000) to wait before sending the next series of data chunks.

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

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 103](#).

- Set copy parameters for a specific volume set by typing:

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

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

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

- Get copy parameters for a specific volume set by typing:

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

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

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

Updating, Copying, and Waiting Operations

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 was 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 (PID) locking see [“PID Locking a Shadow Volume Set” on page 56](#).

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 devices and resume the workload. You do not have to wait for the operation to be complete.

A remote mirror volume set must be in logging mode (only if a remote mirror volume is a *target* of Point-in-Time Copy update/copy) 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` and `-u` Command Options” on page 83](#).

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 108](#).

▼ To Update the Shadow Volume From the Master Volume

1. Unmount the shadow volume, `/dev/rdisk/c1t3d0s4`.
2. Update the shadow volume by typing:

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

3. Mount the shadow volume.

▼ To Update the Master Volume From the Shadow Volume

1. Unmount the master volume, `/dev/rdisk/c1t3d0s0`.
2. Update the master volume by typing:

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

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

3. Mount the master volume.

Performing Full Copy Operations

You typically perform a full volume copy when you need two physically separate volumes, if a volume has become corrupted, or if a volume has failed. See also [“To Copy Volumes” on page 89](#).

▼ To Copy All Data From the Master Volume to the Shadow Volume

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

```
# iiadm -c s /dev/rdisk/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.**

▼ To Copy All Data From the Shadow Volume to the Master Volume

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

```
# iiadm -c m /dev/rdisk/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 want to issue a command involving shadow volume sets that are part of an ongoing copy or update operation, issue a wait command and then issue your next command. The Point-in-Time Copy software waits until the copy or update is complete before proceeding with the next command. See also [“To Wait for a Copy or Update Operation to Complete” on page 110](#).

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 by using the `-n` option to the wait command option. See [“PID Locking a Shadow Volume Set” on page 56](#).

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

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

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

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

The wait command monitors *all* processes executing on the shadow volume set when the command 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 only if your process is the locking process.

Locking a Shadow Volume Set

You can lock a shadow volume set when you update or copy a shadow volume.

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 concludes successfully.
- If the shadow volume set is currently PID locked and you are the original locking process, the operation concludes successfully, although no change has taken place.
- If the shadow volume set is currently locked and you are not the original locking process, the operation fails and an error message is issued.

Unlocking the PID on 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, type:

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

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

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, type:

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

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



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

Listing Shadow Volume Sets and Volumes

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

▼ 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 99](#).

- List all configured shadow volume sets by typing:

```
# iiadm -l
```

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

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

Note – Typing `iiadm` with no options lists the same information as typing `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 100](#).

- List all overflow volumes by typing:

```
# 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 100](#).

- List all I/O groups by typing:

```
# 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 operation is complete. Also see [“To Display Volume Set Status” on page 96](#).

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

```
# iiaadm -i /dev/rdisk/c1t3d0s4
```

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

```
# iiaadm -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 100](#).

- Display the status of a specific overflow volume by typing:

```
# iiaadm -Q /dev/rdisk/c1t4d0s6
```

Note – If the overflow volume is not attached to any compact dependent shadow volumes and you use the `iiaadm -Q` option, you see 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 Operation

Use the `-a` option to abort a copy or update that is in process. The state of the copy or update is saved so that you can complete the operation at a later time. See also [“To Abort a Copy or Update Operation” on page 87](#).

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

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

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

```
# 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, 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 cannot 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 are left inconsistent, so a disable command option is subsequently 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 online with the reset command. If a bitmap volume goes offline, the *delay* and *units* parameters revert to the default values when the volume is reset with the `iiadm -R` command.

TABLE 4-2 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. If the master volume had been offline, for example, bitmap management restarts on the previously established bitmap.

Resetting volumes is usually required when storage array maintenance is required, for example, when you must replace a disk on which part of a shadow volume set resides.

▼ To Reset Volume Sets

Note – If a remote mirror volume is a *target* of Point-in-Time Copy update or copy, 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 the set is not in logging mode, the Point-in-Time copy operation fails and the Remote Mirror software reports that the operation is denied.

An offline 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 106.

- **Reset an offline shadow volume set by typing:**

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

- **Reset all offline shadow volume sets in an I/O group by typing:**

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



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

▼ To Reset the Master Volume of Multiple Shadow Volumes

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 use to update the master volume.**

```
# 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 use for updating the master volume.**

```
# 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
```

Disabling a Shadow Volume Set

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. The shadow volume set is in effect new and the data status at the time of the disable operation 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 90](#) and [“Interaction With svadm” on page 34](#).

- **Disable the Point-in-Time Copy software for a volume set by typing:**

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

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

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

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

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 does not allow you to disable an independent shadow volume set while a copy or update operation is in progress.

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 Operation”](#) on page 61.

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

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

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

The shadow volume set is now disabled.

If a shadow volume set is 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 in a Standalone Environment

The functionality implemented by 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 enables shadow volume processing by associated applications to be off-loaded to a secondary host without affecting 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 being 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 the Point-in-Time Copy software using a second bitmap.

Once secondary host processing is complete, the shadow volume and second bitmap can be disabled on the secondary host and joined to the original master volume. The master, shadow, and bitmap consistency is reconstructed. After completing the join processing, the shadow volume set is 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 volume 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 94.](#)

See also [“To Import a Shadow Volume” on page 97.](#)

See also [“To Join a Shadow Volume” on page 98.](#)

1. Create a shadow volume on the primary host and start the application using that shadow volume.

Primary Host	Secondary Host	Comments
<i>Master</i> volume		Existing <i>master</i> volume, mounted, valid data
Create <i>shadow</i> and <i>bitmap1</i> volumes		Should be the same redundancy (RAID) as the <i>master</i>
<i>shadow</i> volume		Same size as <i>master</i> on a dual-ported device
<i>bitmap1</i> volume		Size based on <i>master</i> volume set
Enable independent shadow set		<code>iiadm -e ind master shadow\ bitmap1</code>
Mount <i>shadow</i> volume		<code>mount shadow mount_point</code>
Start application using <i>shadow</i>		

2. Switch the application and shadow volume to the secondary host.

Primary Host	Secondary Host	Comments
Stop application using <i>shadow</i>		Existing <i>master</i> volume, mounted, valid data
Unmount <i>shadow</i>		<code>umount mount_point</code>
Export <i>shadow</i> volume		<code>iiadm -E shadow</code>
Create <i>bitmap2</i>		same size as <i>bitmap1</i> , on dual-port device
Copy <i>bitmap1</i> to <i>bitmap2</i>		<code>dd bitmap1 bitmap2</code>
Deport <i>shadow/bitmap2</i>		Required for VxVM or SDS(SLVM)
	Import <i>shadow/bitmap2</i>	Required for VxVM or SDS(SLVM)

Primary Host	Secondary Host	Comments
	Import <i>shadow</i> volume	<code>iiadm -I shadow bitmap2</code>
	Mount <i>shadow</i> volume	<code>mount shadow mount_point</code>
	Start application using <i>shadow</i>	

3. Switch the application and shadow volume to the primary host.

Primary Host	Secondary Host	Comments
	Stop application using <i>shadow</i>	
	Unmount <i>shadow</i> volume	<code>umount mount_point</code>
	Disable <i>shadow</i> volume	<code>iiadm -d shadow</code>
	Deport <i>shadow/bitmap2</i>	Required for VxVM or SDS(SLVM)
Import <i>shadow/bitmap2</i>		Required for VxVM or SDS(SLVM)
Join <i>shadow</i> volume		<code>iiadm -J shadow bitmap2</code>
Mount <i>shadow</i> volume		<code>mount shadow mount_point</code>
Start application using <i>shadow</i>		

Exporting, Importing, and Joining Shadows in a Sun Cluster OE

The Point-in-Time Copy software provides the functionality to allow for an independent shadow volume on a multi-ported storage device to be exported to a secondary node within a Sun Cluster Operating Environment (OE), while still under Point-in-Time Copy software control. This capability of Exporting / Importing / Joining of a Point-in-Time Copy shadow set enables shadow volume processing by associated applications to be off-loaded to a secondary node within a Sun Cluster OE, without affecting the primary node's master volume or its associated applications. Since the Point-in-Time Copy software retains control of the shadow volume while it is imported on a secondary node within a Sun Cluster OE, point-in-time consistency is retained to facilitate fast resynchronization via update processing at a later time.

On a per Point-in-Time Copy set basis, each volume set requires a minimum of three raw global device partitions or volume manager controlled volumes (for example, Solaris Volume Manager or VERITAS Volume Manager). For the E/I/J functionality to work correctly, the shadow volume must be on a different global device or volume manager controlled device group than its associated master and bitmap volumes. This allows the shadow volume's device group to be switched between various nodes in a Sun Cluster and to be used as an exportable shadow volume.

Configuring a Point-in-Time Copy Set in a Sun Cluster OE

The steps to create a highly available Point-in-Time Copy volume set are listed in the following section.

There is no convention required for the Resource Group Manager (RGM) resource groups or resource types, although a planned and well thought out naming scheme, spanning the volume manager, if one is used, and the Sun Cluster resource groups and types, will be beneficial later on if troubleshooting is required.

The setup creates a Point-in-Time Volume set on Sun Cluster nodes, node-A and node-B, with the exportable shadow volume available on node-C.

▼ To Configure a Point-in-Time Copy Set in a Sun Cluster OE

Note – This example uses two SVM device groups: “oracle” and “backup”, where “oracle” is the master volume's device group and “backup” is the exportable shadow volume's device group.

This example is based on configuring the following Point-in-Time Copy set:

```
# iiadm -ne ind /dev/md/oracle/rdisk/d1 /dev/md/backup/rdisk/d1 \  
/dev/md/oracle/rdisk/d2
```

Note – Do not invoke the above, or similar, `iiadm` command at this time in the following sequence of steps. The device attributes of the constituent volumes of an enabled Point-in-Time Copy set are such that without Sun Cluster Resource Group Manager (RGM) control, the association of a master and bitmap volume, allowing an exportable shadow volume, makes all of the associated device groups no longer highly available. This fact could impact the high availability of the Sun Cluster as it pertains to these associated volumes.

1. Create a resource group that will contain the HAStoragePlus resource type associated with the Point-in-Time Copy set.

This resource group should specify two or more nodes within the Sun Cluster, or be left blank if all nodes in the Sun Cluster are capable of supporting the Point-in-Time Copy set as a highly available resource.

```
# scrgadm -a -g Availability_Suite_RG -h node-A,node-B[,node-  
C,...]
```

Or, for all nodes in the Sun Cluster, as long as the Availability Suite software has been installed and configured on each:

```
# scrgadm -a -g Availability_Suite_RG
```

Note – Additional resource types for other Sun Cluster High Availability (HA) data services or applications may be added to this same resource group at your discretion.

2. Assure that the `SUNW.HAStoragePlus` resource type is registered. If not register it.

```
# scrgadm -p | grep "Res Type name:" | grep HAStoragePlus
# scrgadm -a -t SUNW.HAStoragePlus
```

3. Add an `HAStoragePlus` resource type to the previously created resource group.

The `HAStoragePlus` resource type will be used to specify two Sun Cluster device groups, one representing the master-bitmap volume pair, and one for the exportable shadow volume. The ordering of these device group is important; the exportable shadow volume must be the last one specified.

Note – The `HAStoragePlus` resource type allows for its `GlobalDevicePaths` parameter to be either the full device path specification of a Sun Cluster device or the name of a Sun Cluster device group. The first format is used in this example, so that no doubt exists as to which devices are being associated. In doing so, the example will be setting a `GlobalDevicePath` with both a master and bitmap volume, each of which is in the same device group. As such, one of the device path specifications is redundant, and will be ignored.

```
# scrgadm -a -g Availability_Suite_RG -j \
Availability_Suite_RES -t SUNW.HAStoragePlus -x \
GlobalDevicePaths=/dev/md/oracle/rdisk/d1,\
/dev/md/oracle/rdisk/d2,/dev/md/backup/rdisk/d1 -x \
AffinityOn=False
```

The Sun Cluster resource type `SUNW.HAStoragePlus`, supports a configuration option `AffinityOn`, with the default value being `True`. This setting, along with the fact that the `GlobalDevicePaths` qualifier contains two device groups, with one of the device groups being the exportable shadow volume, implies that if the exportable shadow volume is currently in use on a secondary Sun Cluster node, this resource group will have a strong affinity to move to the node, regardless of the node list specified in step 2 above.

For example, in a three (or more) node Sun Cluster configuration, where two nodes have the system resources to support an HA enterprise class application like Oracle and the third Sun Cluster node is a low-end backup system, if the exportable shadow volume is in use on this third system, the HA application will move to the third Sun Cluster node, even if there are not system resources capable of supporting its execution. This is the justification for setting `AffinityOn=False` in the example above.

4. Bring the resource group online. Then, assure that the resource group is located on the Sun Cluster node where the Point-in-Time Copy enable command will be invoked.

```
# scswitch -Z -g Availability_Suite_RG
# scswitch -z -g Availability_Suite_RG -h node-A
```

5. Enable the Point-in-Time Copy set, taking note of the new option `-n` to enable exportable shadows.

```
# iiadm -ne ind /dev/md/oracle/rdsk/d1 /dev/md/backup/rdsk/d1 \
/dev/md/oracle/rdsk/d2
```

6. Validate that the Point-in-Time Copy set is available on this node.

```
# iiadm -i /dev/md/backup/rdsk/d1
# scstat -g
# scstat -D
```

7. Switch the resource group from this node to each of the other configured node(s), and validate the set.

```
# scswitch -z -g Availability_Suite_RG -h node-B
# telnet node-B
<login to root account>
# iiadm -i /dev/md/backup/rdsk/d1
# scstat -g
# scstat -D
# ^D {logout}
```

8. This Point-in-Time Copy volume set is now highly available and usable as a resource group to which other highly available applications (HA-NFS, HA-ORACLE, and so forth) can now be added.

For example:

```
# scrgadm -a -g Availability_Suite_RG -j nfs_res -t SUNW.nfs
# scswitch -e -j nfs
```

To use the Point-in-Time shadow volume on another node within the Sun Cluster, it must be exported from its associated set and disabled as a device path within its HAStoragePlus resource type.

9. Prior to being exported from a Point-in-Time Copy set, the II set must be fully independent, which is confirmed by `wait (iiadm -w)`, returning.

```
# iiadm -w /dev/md/backup/rdisk/d1
```

10. Export the II shadow volume from its associated Point-in-Time Copy set.

```
# iiadm -E /dev/md/backup/rdisk/d1
# iiadm -i /dev/md/backup/rdisk/d1
```

11. The Point-in-Time exportable shadow volume can be switched to another node in the Sun Cluster.

```
# scswitch -z -D backup -h node-C
```

Or the Point-in-Time Copy set can be switched to another node in the Sun Cluster.

```
# scswitch -z -g Availability_Suite_RG -h node-C
```

12. Then, validate the correct behavior.

```
# telnet node-C
<login to root account>
# iiadm -i /dev/md/backup/rdisk/d1
# scstat -g
# scstat -D
```

13. The Point-in-Time Shadow volume is now accessible independently from the Point-in-Time Copy set, off-host, yet the original Point-in-Time set is still active on the other Sun Cluster node.

If the shadow volume will be accessed in a read/write mode, a secondary bitmap should be used to import the shadow locally on this node, so that subsequent fast-resynchronization operations (`iiadm -u`) can be done, versus the full-synchronization (`iiadm -c`).

Note – The exportable shadow MUST be enabled with the `-C local` tag, so that the system can differentiate between the highly available Point-in-Time Copy set and the locally accessible exportable shadow, which have the exact same name.

```
# iiadm -C local -I /dev/md/backup/rdsk/d1 /dev/md/backup/rdsk/d2
# iiadm -i /dev/md/backup/rdsk/d1
```

Note – From this node, you will see the imported shadow volume and the shadow volume's Point-in-Time Volume set as suspended on this node and active on node-A (or node-B).

14. While this imported shadow volume is active on this node, test steps #4 and #5 to validate that the original Point-in-Time Copy set is still highly available.

Remember that the original Point-in-Time Copy set is not configured to be highly available on node-C, since an attempt to do so will fail as long as the shadow volume is imported on this node.

15. When you are done using the imported shadow volume on this node (if it was decided to import it), disable the locally accessible imported shadow volume, switch it back to the node where the Point-in-Time Volume set is active, and enable the resource in the resource group.

```
# iiadm -C local -d /dev/md/backup/rdsk/d1
```

16. Take the resource offline and back online, forcing the exportable shadow volume back to the Sun Cluster node, where the rest of the Point-in-Time Copy set is enabled.

```
# scswitch -n -j Availability_Suite_RES
# scswitch -e -j Availability_Suite_RES
```

Note – From this node, you will still see the shadow volume's Point-in-Time volume set as suspended on this node, active on node-A (or node-B). The imported shadow volume is no longer listed.

```
# iiadm -i /dev/md/backup/rdsk/d1
# ^D {logout, back to node-A }
```

17. Now join the shadow volume (and possible changes) back with the original Point-in-Time Copy set.

```
# iiadm -J /dev/md/backup/rdisk/d1 /dev/md/backup/rdisk/d2
```

18. The Point-in-Time Copy set is back in its original state, as though the shadow volume has never been exported.

Point-in-Time Copy Set Considerations in a Sun Cluster OE

This section discusses issues involved with Point-in-Time Copy sets in a Sun Cluster OE.

Redundancy

To provide high availability to the data contained on a Point-in-Time Copy set when using global devices, it is assumed that the master, shadow and bitmap volumes are on redundant storage, since there is no means today to provide for host-based data service redundancy on raw global devices. If controller based redundancy is not available, then a Sun Cluster supported volume manager must be used. For performance reasons, it is recommended under all scenarios that bitmap volumes are NOT placed on RAID-5 volumes (either host-based or controller-based), since the nature of bitmap I/O processing within a Point-in-Time Copy volume set can be I/O intensive.

Implicit Join Operation

The implicit join operation is a new feature of the Availability Suite software, when using the Export, Import, Join functionality in a Sun Cluster OE. By design, a Point-in-Time Copy set with its shadow volume exported, can not coexist on the same node as the Point-in-Time Copy set containing the imported shadow volume.

From an operational point of view, having the master volume and shadow volume on two different Sun Cluster node is the sole reason for using Export, Import, Join functionality in a Sun Cluster OE. If you wanted both master and shadow volumes on the same node, you would have avoided using Export, Import, Join in the first place.

If a Sun Cluster voluntary or involuntary failover event decided to move the master and associated bitmap volume to the Sun Cluster node containing the imported shadow volume, the aforementioned design issue would prevent the failover from completing successfully. To address this concern, the Availability Suite software detects this failover condition and performs an implicit join operation, merging the two Point-in-Time Copy sets back into one. This operation should have no impact on the master or shadow volume, other than the fact that both volumes are now in the same Point-in-Time Copy set, on the same node in the Sun Cluster.

Incomplete Export / Import / Join Sequence

The operational procedures for Export, Import, Join are based on the assumption that all three steps will be performed. Due to circumstances in system availability outside of Availability Suite, after exporting the shadow volume from the set, it may not be possible to perform the Import step at this time, yet it is desirable to Join the exported shadow volume back into the Point-in-Time Copy set. In other words, it may be desirable at times to perform an Export, Join sequence, with no Import step. To perform a Join operation, there is still the requirement for a secondary bitmap volume, but since this secondary bitmap volume was NOT used during a recent Import operation, it contains stale data or uninitialized data. Prior to performing a Join operation, it is required to copy (using the Solaris `dd` utility) the current bitmap volume over the contents of the secondary bitmap volume, so that the secondary bitmap volume's data is in a known state. The failure to perform this manual initialization step may result in the Join operation failing, or when state data is used, it may cause an inconsistency between what is actually on the shadow volume and the current state as recorded in the bitmap.

Displaying Help and Software Version Numbers

The Point-in-Time Copy software 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. Command-line errors 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 content by typing:

```
# iiaadm -h
```

▼ To Display Software Version Numbers

The version numbers of installed Point-in-Time Copy software modules are accessible through the `-v` option. See also [“To Print Software Versions” on page 109](#).

- Display the software version numbers by typing:

```
# iiaadm -v
```

dsbitmap Bitmap Sizing Utility

The `dsbitmap` utility is installed along with the Sun StorageTek Availability Suite software. Use it to calculate the required size of a bitmap for a Point-in-Time Copy shadow volume set or a Remote Mirror volume set.

The `dsbitmap` utility is typically used by the system administrator during the initial stages of configuring Sun StorageTek Availability Suite software. The utility determines the required bitmap volume sizes, and then checks if the bitmap volumes that have been created are suitable.

`dsbitmap`

This utility enables you to determine the required size of the volume for a Point-in-Time Copy bitmap or a Remote Mirror bitmap. If you include an optional proposed bitmap volume in the command, its suitability as a bitmap volume for the proposed data volume is tested.

Syntax

To obtain the size of a Point-in-Time Copy bitmap, use this command:

```
dsbitmap -p data-volume [bitmap-volume]
```

To obtain the size of a Remote Mirror bitmap, use this command:

```
dsbitmap -r data-volume [bitmap-volume]
```

Usage for dsbitmap

```
# dsbitmap -h
usage: dsbitmap -h
       dsbitmap { -p | -r } data_volume [bitmap_volume]
       -h : This usage message
       -p : Calculate size of Point in Time bitmap
       -r : Calculate size of Remote Mirror bitmap
```

Examples for dsbitmap

Remote Mirror volumes display both memory and disk queue sizes.

```
# dsbitmap -r /dev/md/rdisk/d100
Remote Mirror bitmap sizing

Data volume (/dev/md/rdisk/d100) size: 2064384 blocks
Required bitmap volume size:
  Sync replication: 9 blocks
  Async replication with memory queue: 9 blocks
  Async replication with disk queue: 73 blocks
```

Point-in-Time Copy volumes display sizes for independent, dependent, and compact dependent sizes.

```
# dsbitmap -p /dev/md/rdisk/d100
Point in Time bitmap sizing

Data volume (/dev/md/rdisk/d100) size: 2064384 blocks
Required bitmap volume size:
  Independent shadow: 62 blocks
  Full size dependent shadow: 62 blocks
  Compact dependent shadow: 798 blocks
```


Command-Line Interface

This chapter describes the Point-in-Time Copy software's command-line interface (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 Modifiers" on page 81](#)
- ["Help Information" on page 85](#)
- ["Command Options" on page 84](#)
- ["Option List" on page 87](#)

Command Modifiers

The primary user interface to the Point-in-Time Copy software is the `/usr/sbin/iiadm` command.

The following terms apply to all examples and explanations in this chapter.

Name	Definition
<i>master</i>	The specific master volume referenced in the command
<i>shadow</i>	The specific volume set, whose name defaults to the name of its associated shadow volume
<i>shadowvol</i>	The name of a specific shadow volume
<i>bitmap</i>	The name of a specific bitmap volume

Name	Definition
<i>cluster</i>	The name, or <i>tag</i> , of a specific cluster
<i>io-groupname</i>	The specific I/O group, whose name is created when using a <code>move (to group)</code> or an <code>enable</code> command
<i>overflow</i>	The specific overflow volume referenced in the command

The `iiadm` CLI performs only one action per command invocation. You cannot combine multiple options, except with the `-g` and `-C` command modifiers. Command modifiers 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`

The `-g` 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

The `-C` 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 device group, physical and logical, so that failover works 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 *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*.

Note – The core and data services software is cluster aware in the Sun Cluster 3.2 environment and provides high availability for the Sun StorageTek data services.

-n Option to the -c and -u Command Options

The `-n` 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 while 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 type an `iiadm` command, you must include an option telling the software what to do. Usually, you also type one or more volume, set, and group names, that tell the software where to execute the command.

With few exceptions, only one option can be included 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 you type an unsupported action item, `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.

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
```

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

Configuration Parameters

There are four configuration parameters that directly influence the operation of the Point-in-Time Copy software. You can use `vi` or another editor to modify these parameters.

Note – When changing any of the parameters described in this section, you must restart the Availability Suite services using `dscfadm` 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 the `/usr/kernel/drv/ii.conf` file to change this value.

If set to 0, the bitmaps are maintained only in memory 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 are 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 the `/usr/kernel/drv/ii.conf` file.

`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 the `/usr/kernel/drv/ii.conf` file.

`ii_debug`

This variable defaults to a value of 0, but can be set to 1 or 2, each of 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 here in alphabetical order.

To Abort a Copy or Update Operation

Use this option when you need to abort any copy or update operation before it completes execution. 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 reissue the update or copy option to resynchronize the volumes.

Note – See [“Cautions: Enable, Copy, and Update Operations”](#) on page 33.

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.

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, which directly performs 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 102](#)), it will be initialized before it is attached. 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 Operations”](#) on page 33

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 by using 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 56.

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 – If a remote mirror volume is a *target* of Point-in-Time Copy update or copy, 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 the set is not in logging mode, 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 34.

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 64 KB 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 for all volume sets in specific I/O groups so that the software no longer maintains the shadow volumes’ point-in-time snapshots or the associated bitmaps.

Use this option to terminate a volume set association that 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 standalone copies. The set 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 Operations”](#) on page 33



Caution – When creating shadow volume sets, do not create shadow or bitmap volumes using partitions that include cylinder 0. Data loss might occur. See [“VTOC Information”](#) on page 22.

The enable option enables a point-in-time copy volume set, composed of the specified master volume, shadow volume, and bitmap volume, and then takes a point-in-time snapshot of the master volume. The shadow volume, dependent or independent, houses the point-in-time copy.

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, the software creates and attaches the volume set 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.

Note – When you are using compact dependent shadow volume sets we suggest that you attach a shared overflow volume right after the sets creation, so as to address a potential out of volume space issue now, instead of later when the compact dependent shadow volume may overflow.

Use this option to initially create 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 – If a Remote Mirror volume is a *target* of Point-in-Time Copy update or copy, 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 the set is not in logging mode, 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. For example, you might do this 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 help content for all Point-in-Time Copy software options and modifiers.

When you type 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 type `all` (or leave blank) 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 Operation to Complete”](#) on page 110.

```
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/rdisk/shadowdg/shadowvol1
/dev/vx/rdisk/masterdg/mastervol1: (master volume)
/dev/vx/rdisk/shadowdg/shadowvol1: (shadow volume)
/dev/vx/rdisk/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)
```

Sample output showing a compact dependent shadow volume:

```
# iiadm -i
/dev/md/rdisk/d3000: (master volume)
/dev/md/rdisk/d3011: (shadow volume)
/dev/md/rdisk/d3021: (bitmap volume)
Dependent copy, compacted shadow space
Volume size:        262144
Shadow chunks total: 2048 Shadow chunks used: 0
Percent of bitmap set: 0
                    (bitmap 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 in a Standalone Environment” on page 67](#) 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 might 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 all volume sets or all volume sets in an I/O group. All configured volume sets, including offline and suspended volume sets, are shown.

```
iiadm -l
```

Syntax

```
iiadm -l
```

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

Sample output:

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

Note – Typing `iiadm` with no options lists the same information as typing `iiadm -l`.

To List Overflow Volumes or Groups

This option lists 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. 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, by using two quotation 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 to simplify operations. The move option places volume sets into I/O volume groups so that all volumes in the group can be operated on at the same time by a single command.

```
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, which directly performs 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 copy operations do not excessively consume system resources. A throttled copy is tuned by setting the maximum number of chunks of data (100 through 60000) to be sent between pauses and by setting the length of the pause in system clock ticks (2 through 10000). 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 by using the `iiadm -P shadow` command. Get copy parameters for an I/O group by using the `iiadm -g io-groupname -P` command.

At installation, the Point-in-Time Copy software defaults to the minimum settings for both *delay* and *units* parameters. 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 revert 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 to 10000) between copy operations

units Number of chunks of data (100 to 60000) sent before pausing for the set delay

Sample output:

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


To Display Overflow Volume Status

This option displays the status of the specified overflow volume.

Note – 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/rdisk/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 returns specific offline volume sets, or all offline volume sets in an I/O group, 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 is reestablished. A point-in-time snapshot is created or, in the case of independent copy volume sets where the bitmap was offline, the master and shadow volumes are marked as different.

Note – If a bitmap volume goes offline, the *delay* and *units* parameters revert 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 – If a remote mirror volume is a *target* of Point-in-Time Copy update or copy, 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 the set is not in logging mode, 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. 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 use to update the master volume.

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

2. Wait for all reset operations to finish.

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

3. Reset the volume set that contains the shadow volume you want to use to update the master volume.

```
# 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 Operations”](#) on page 33.

Use this option to update a volume set or all volume sets in an I/O group. The update can be 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 appearing by using 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 56.

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. An update is very quickly completed, as compared to a full volume copy. 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 – If a remote mirror volume is a *target* of Point-in-Time Copy update or copy, 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 the set is not in logging mode, 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 numbers of the major software modules. For example, you might need to know the software versions before you perform a software update.

```
iiadm -v
```

Syntax

```
iiadm -v
```

Sample output:

```
# iiadm -v  
PointInTime version 11.10
```

To Wait for a Copy or Update Operation 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. No other commands can execute on that volume set until the copy or update operation is complete.

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 (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 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
```

dsstat and Storage Cache Statistics

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

The `/usr/sbin/dsstat` command displays I/O statistics for the Remote Mirror, Point-in-Time Copy, and Cache software. See the `dsstat` man page for more information.

Unless you specify otherwise, all fields displayed show per-second averages based on the data collected during the interval. For example, if you specify 5-second intervals, the data displayed is a per-second average of the data collected during the last 5-second interval.

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.

TABLE A-1 dsstat Options

Option	Description
<code>-m mode</code>	<p>Specifies the statistics reporting mode, where <i>mode</i> is one of the following:</p> <ul style="list-style-type: none">• <code>cache</code>• <code>ii</code>• <code>sndr</code> <p>This option determines the options you can use with the <code>-r report-option</code>. You cannot use the <code>-r report-option</code> if you specify <code>-m cache</code>.</p> <p>If <code>-m</code> is not specified, the default is for the statistics for all modes to be displayed. You can enter a comma-separated list of modes as well as multiple <code>-m</code> switches. The following commands are equivalent:</p> <pre># dsstat -m sndr,ii # dsstat -m sndr -m ii</pre>
<code>-r report-options</code>	<p>No <i>report-options</i> are available for the <code>-m cache</code> mode.</p> <p>If you specify <code>-m ii</code>, you can use this option as follows. If you do not specify <code>-r report-options</code>, the default display is <code>-r msbo</code>.</p> <ul style="list-style-type: none">• <code>-r m</code> shows master volume information.• <code>-r s</code> shows shadow volume information.• <code>-r b</code> shows bitmap volume information.• <code>-r o</code> shows overflow volume information (if this volume type is attached). <p>If you specify <code>-m sndr</code>, you can use this option as follows. If you do not specify <code>-r report-options</code>, the default display is <code>-r bn</code>.</p> <ul style="list-style-type: none">• <code>-r b</code> shows bitmap volume information (this command shows only the bitmap of the host you are issuing the command from).• <code>-r n</code> shows network volume information.
<code>-d display-options</code>	<p>Specifies the types of information to show. You can specify one or more of the following options on the command line. (If you specify <code>-d t</code>, the <code>r</code> and <code>w</code> options are ignored.)</p> <p>If you specify <code>-m cache</code>, the available options are as follows. If you do not specify this option, the default display is <code>-d sf</code>.</p> <ul style="list-style-type: none">• <code>-d r</code> shows detailed read information.• <code>-d w</code> shows detailed write information.• <code>-d d</code> shows destaged data information.• <code>-d c</code> shows cancelled write information.• <code>-d s</code> shows summary of information.• <code>-d f</code> shows cache behavior flags.

TABLE A-1 dsstat Options (Continued)

	<p>If you specify <code>-m sndr</code> or <code>-m ii</code>, the available options are as follows. If you do not specify this option, the default display for <code>-m ii</code> is <code>-d sf</code> and for <code>-m sndr</code> is <code>-d spf</code>.</p> <ul style="list-style-type: none">• <code>-d r</code> shows detailed read information.• <code>-d w</code> shows detailed write information.• <code>-d t</code> shows timing information.• <code>-d s</code> shows summary of information.• <code>-d f</code> shows volume type and status flags.• <code>-d p</code> shows (<code>-m sndr</code> only) percentage of volume requiring a synchronization.• <code>-d a</code> shows (<code>-m sndr</code> only) shows the secondary volume names.
<code>-s volume-set</code>	<p>Shows information about the specified <i>volume-set</i> only. <i>volume-set</i> can also be a comma-delimited list of volume names.</p>
<code>-f</code>	<p>Displays a field header for every report cycle.</p>
<code>-F</code>	<p>Displays a field header once, when reporting begins.</p>
<code>-z</code>	<p>Does not report or display lines that have a value of zero (indicates no activity).</p>
<code>interval [count]</code>	<p>Specifies how often the report is updated. The default interval is one second. <i>count</i> specifies how many reports to display. If <i>count</i> is not specified, report output continues at the specified <i>interval</i> time until you interrupt it by a break (^C).</p> <p>If you do not specify <i>interval</i> and <i>count</i>, the command shows a single report for a one second interval.</p> <p>Unless you specify otherwise, all fields displayed show per second averages based on the data collected during the interval. For example, if you specify 5-second intervals, the data displayed is a per-second average of the data collected during the last 5-second interval.</p>

Examples

TABLE A-2 describes the fields displayed.

Report Cache Statistics

- Report cache statistics (`dsstat -m cache`)
- Show detailed information for read and write statistics (`-d rw`)
- Show information for the volume `c1t35d0s6` (`-s /dev/rdisk/c1t35d0s6`)
- Generate reports at five-second intervals (`5`)

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

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

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

Report Master and Shadow Volume Point-in-Time Copy Software Statistics

- Report master and shadow volume Point-in-Time Copy software statistics (dsstat -m ii -r ms)
- Generate reports at five-second intervals (5).

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

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

Report Network Statistics for the Remote Mirror Software

- Reports network statistics for the Remote Mirror software (-m sndr, -r n)
- Shows information for the secondary volume c1t35d0s6 (-s /dev/rdsk/c1t35d0s6)
- Shows detailed information for read and write statistics, timing, and volume type or status flags (-d rwtpf)
- Generates reports at five-second intervals (5)

```
# 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

TABLE A-2 dsstat Display Fields

Field	Description
set name	Volume set specified by -s option
t	Volume type for remote mirror volumes: P primary host volume S secondary host volume Volume type for the point-in-time copy volumes: I independent shadow volume set D dependent shadow volume set Valid cache modes: C cache reads and writes D disk reads and writes
s	Volume status for the Remote Mirror software: L volume is in logging mode; changes are being logged R replicating changes to secondary SY forward synchronization is in progress RS reverse synchronization is in progress SN forward synchronization needed RN reverse synchronization needed VF volume failed BF bitmap failed Volume status for the point-in-time copy volumes: C copy in progress - no copy in progress
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

Field	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

Glossary

- atomic** (adj.) 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** (n.) For every 32 Kbyte block of a master volume that is part of a shadow volume set, a bitmap volume maintains a bit that indicates if the data at the block has changed with respect to its associated point-in-time copy.
- changed** (adj.) Term applied to data blocks whose contents have been altered since the last synchronization.
- compact dependent shadow volume** (n.) A shadow volume that is smaller than its associated master volume. Compact conveys that less storage is allocated, not that the data in the blocks is compacted or compressed in any way.
- data service** Highly Available (HA) applications within the Sun Cluster environment are known as data services. The term data service is used to describe a third-party application that has been configured to run on a cluster rather than on a single server. A data service includes the application software and Sun Cluster software that starts, stops, and monitors the application. This terminology should not be confused with the concept of the Availability Suite software data service called Point-in-Time Copy.
- delay variable** (n.) 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** (adj.) Describes a shadow volume set 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 read operations.
- enable** (v.) In the Sun StorageTek Availability Suite 4.0 Point-in-Time Copy software, to use the `enable` command to associate the constituent volumes of a shadow volume set with one another.

- export** (v.) With Point-in-Time Copy software, to make a shadow volume available for another host to use.
- exportable shadow** A point-in-Time Copy set's shadow volume, in a Sun Cluster OE, that is in a different device group than the set's master and bitmap volumes. The ability to create a set with this configuration is available only when using the `iiadm -n` option, when initially creating the Point-in-Time volume set.
- fast synchronization** See *update*.
- full volume copy** (v.) 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** (n.) 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** (v.) To move a shadow volume that has been exported by another host into a new host for the host's own use. 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** (adj.) 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** (v.) To attach a shadow volume that has been exported to its original shadow volume set by using the `join` command.
- master volume** (n.) The volume containing the original data.
- overflow volume** (n.) A volume that is attached to a compact dependent shadow volume to accept write operations that are in excess of the compact dependent shadow volume's capacity.
- point-in-time copy** (n.) A view of a data volume's contents relative to a point in time. While the data volume itself might continue changing with time, the point-in-time copy does not.
- remote mirror** (n.) Software that provides for remote replication of data volumes over many line protocols, including TCP/IP. Remote Mirror software is part of Sun StorageTek Availability Suite software.
- resynchronize** (v.) To perform synchronization between volumes in a shadow volume set that have been previously synchronized. This can be a full volume copy or an *update*.
- scoreboard** (n.) Another name for bitmap volume.

scoreboarding	(v.) A technique that occurs when the Point-in-Time Copy software uses a bitmap volume to create point-in-time copies. For every 32 Kbyte block of a master volume that is part of a shadow volume set, a bit is maintained that indicates if the data at the block has changed with respect to its associated point-in-time copy.
shadow volume	(n.) A volume containing a point-in-time copy of the data on an associated master volume.
shadow volume set	(n.) 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 might include an overflow volume.
synchronize	(v.) 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	(adj.) Term applied to data blocks whose contents have not been altered since the last synchronization.
units variable	(n.) Part of the <code>iiadm -P</code> command. This variable sets the number of data chunks to send before pausing for delay variable time to allow other applications to share system resources.
update	(v.) 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	(n.) An application's view of the shadow volume in a dependent shadow volume set. Some blocks of data might reside on the master volume and some might reside on the shadow volume. To a reading or writing application, it appears as a single volume, a virtual shadow.

Index

A

- aborting copies or updates, 61
- attaching overflow volumes, 88

B

- bitmap management, 14
- bitmap sizing
 - dsbitmap utility, 79
- bitmap volumes, 81

C

CLI

- command modifiers, 81
- command options, 84
- copy and update operations, 33
- clock ticks, 103
- command modifiers
 - clustering
 - iiadm -C, 83
 - grouping
 - iiadm -g, 82
- command option
 - aborting copies or updates
 - usage, 61
 - attaching overflow volumes
 - usage, 48
 - copying volume sets
 - usage, 54
 - detaching overflow volumes
 - usage, 49
 - disabling volume sets
 - usage, 65

- displaying help text
 - usage, 77
- displaying software versions
 - usage, 78
- displaying volume set status
 - usage, 60
- enabling dependent volume sets
 - usage, 46
- enabling independent volume sets
 - usage, 45
- exporting shadow volumes
 - usage, 68
- initializing overflow volumes
 - usage, 47
- listing I/O groups
 - usage, 59
- listing overflow volume status
 - usage, 60
- listing overflow volumes
 - usage, 58
- listing volume sets
 - usage, 58
- moving volume sets
 - usage, 50
- resetting volume sets
 - usage, 62
- setting copy parameters
 - usage, 51
- waiting for copies or updates
 - usage, 55

command options

- aborting
 - syntax, 87

- attaching overflow volumes
 - syntax, 88
- copying
 - syntax, 89
- detaching overflow volumes
 - syntax, 91
- disabling volume sets
 - syntax, 90
- displaying command usage
 - syntax, 95
- displaying overflow volume status
 - syntax, 105
- displaying volume set status
 - syntax, 96
- enabling volume sets
 - syntax, 92
- exporting shadow volumes
 - syntax, 94
- help
 - iiadm -h, 85
- importing shadow volumes
 - syntax, 97
- initializing overflow volumes
 - syntax, 102
- joining shadow volumes
 - syntax, 98
- listing I/O groups
 - syntax, 100
- listing overflow volumes
 - syntax, 100
- listing volume sets
 - syntax, 99
- moving volume sets
 - syntax, 101
- printing software versions
 - syntax, 109
- resetting I/O groups
 - syntax, 106
- resetting volume sets
 - syntax, 106
- setting copy parameters
 - syntax, 103
- updating volume sets
 - syntax, 108
- waiting for copies or updates
 - syntax, 110
- commands
 - dsstat, 111
- compact dependent shadow volumes, 12

- and overflow volumes, 47
 - overflow volumes for, 13
- configuration parameters, 86
 - ii_Bitmap, 86
 - ii_debug, 86
 - ii_throttle_delay, 86
 - ii_throttle_units, 86
- copy
 - wait for completion, 55
- copy and update operations
 - Using the CLI for, 33
- copy data
 - master volume to shadow volume, 54
 - shadow volume to master volume, 54
- copy operation
 - aborting, 61, 87
- copy operations
 - dependent, 10
 - independent, 7
- copy parameters, 51
 - getting and setting, 51
- copying volumes, 89

D

- data service I/O stack
 - interaction in, 37
- data services logging file, 32
- delay units function, 28
- dependent copy operations, 10
- dependent shadow volume
 - accessing, 11
 - creating, 10
- dependent volume set
 - enabling, 46
- dependent volume sets, 46
- detaching overflow volumes, 91
- disabling volume sets, 90
- displaying command usage, 95
- displaying help, 77
- displaying overflow volume status, 60, 105
- displaying software versions, 78
- displaying volume set status, 60, 96
- dsbitmap utility, 79
- dsstat command, 111

E

- Enable, Copy, and Update Operations, 33
- enabling volume sets, 92
- encapsulated volumes, 34
 - shadowing, 34
- export shadow, 21
- Export/Import/Join
 - SAN-accessible shadow volumes, 29
- exporting shadow volumes, 94

F

- failovers, 39

H

- Help
 - displaying, 77
- help
 - information, 85

I

- I/O groups
 - listing, 59
- I/O stack
 - interaction in, 37
- ii, 86
- iiadm -A command, 88
- iiadm -a command, 87
- iiadm -c command, 89
- iiadm CLI, 82
- iiadm -D command, 91
- iiadm -d command, 90
- iiadm -E command, 94
- iiadm -e command, 92
- iiadm -h command, 95
- iiadm -I command, 97
- iiadm -i command, 96
- iiadm -J command, 98
- iiadm -L command, 100
- iiadm -l command, 99
- iiadm -m command, 101
- iiadm -O command, 102
- iiadm -P command, 103
- iiadm -Q command, 105
- iiadm -R command, 106

- iiadm -u command, 108
- iiadm -v command, 109
- iiadm -w command, 110
- importing shadow volumes, 97
- independent copy operations, 7
- independent shadow volume
 - accessing, 8
 - creating, 7
- independent volume set
 - enabling, 45
- independent volume sets, 45
- initializing overflow volumes, 102

J

- joining shadow volumes, 98

L

- listing
 - volume sets in I/O group, 99
- listing I/O groups, 100
- listing overflow volumes, 100
- listing volume sets, 99

M

- master volume
 - copy data to shadow volume, 54
 - quiescing, 26
 - updating from shadow volume, 53
- mounted volumes
 - copies of, 27
- moving shadow volume sets, 50
- moving volume sets, 101

N

- nsctl framework, 3

O

- operating the Point-in-Time Copy software, 24
- operational notes, 33
- overflow volume, 47
 - attaching, 88
 - attaching to shadow volume sets, 48
 - detaching from shadow volume sets, 49
 - display status, 60
 - initializing, 47

overflow volumes, 82

P

parameters, 103

performance considerations, 40

performing full copy operations, 54

PID locking a shadow volume set, 56

PID unlocking a shadow volume set, 57

point-in-time copy, xi, 3

Point-in-Time Copy Operations, 24

Point-in-Time Copy software

 using with Remote Mirror software, 36

point-in-time snapshot, xi, 3

printing software versions, 109

Q

quiescing, 33

quiescing a master volume, 26

R

RAID-1, 2

RAID-5, 2

raw volumes, 44

resetting volume groups, 106

resetting volume sets, 106

resynchronizing

 master volume to shadow volume, 11

 shadow and master volumes, 9

 shadow volume to master volume, 11

root file system

 shadowing, 33

S

scoreboarding, 14

setting copy parameters, 103

shadow and master volumes

 resynchronizing, 9, 11

shadow volume

 copy data to master volume, 54

 Exporting, Importing, Joining in standalone environment, 67

 Exporting, Importing, Joining in Sun Cluster environment, 70

 updating from master volume, 53

shadow volume set

 disabling, 65

 display status, 60

 PID locking, 56

 PID unlocking, 57

 resetting, 62

 resetting master volume, 63

 to back up Remote Mirror resync, 38

shadow volume sets, 6, 50

 listing, 58

 moving into I/O groups, 50

 multiple, 21

shadow volumes

 compact dependent, 12

 SAN accessible, 29

shadowing encapsulated volumes, 34

shadowing the root file system, 33

shutdown, 25

Software version numbers

 displaying, 78

Solstice DiskSuite, 44

startup, 25

storage volume limit

 increasing, 43

Sun Cluster

 Point-in-Time copy set considerations, 76

 using Point-in-Time Copy software with, 39

svadm, 34

System, 25

system startup and shutdown, 25

T

tags, 82

throttled copies, 103

U

update

 wait for completion, 55

update operation

 aborting, 61, 87

updating volume sets, 108

V

VERITAS Volume Manager, 44

volume names, 33

 length of, 33

volume sets

- creating and configuring, 34
 - enabling, 45
 - grouping, 30
- volume sizing
 - using with Remote Mirror software, 38
- volume sizing requirements, 44
- volumes listing, 58
- VTOC information, 22

W

- waiting for copies or updates, 110

