



Sun StorageTek™ Availability Suite 4.0 Software Installation and Configuration Guide

Sun Microsystems, Inc.
www.sun.com

Part No. 819-6147-10
April 2006, [Revision A](#)Revision

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, 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, 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	vii
1. Upgrading, Installing, and Uninstalling Availability Suite Software	1
Upgrading Availability Suite Software	2
▼ To Upgrade from AVS 3.2	2
Installing Availability Suite Software	3
▼ To Install AVS 4.0	3
Uninstalling Availability Suite Software	4
▼ To Uninstall AVS 3.2	4
2. Initial Configuration Procedures	5
Overview of Initial Configuration Steps	6
Configuring System Files	7
▼ To Edit the <code>/etc/hosts</code> File	7
Configuring the IP Stack (IPv4 and IPv6)	8
▼ To Set Up an IPv6 Address	8
▼ To Edit the <code>/etc/services</code> File	11
▼ To Edit the <code>/etc/nsswitch.conf</code> File	12
Modifying Settings	13
Setting the Bitmap Operation Mode	13

Increasing the Number of Volume Sets	14
Increasing the Storage Volume Device Limit	14
Initializing the Configuration Database and Starting Services Using <code>dscfgadm</code>	15
The <code>dscfgadm</code> Utility	15
Usage	15
Options	15
Initializing the Configuration Database and Starting Services	16
Enabling or Disabling Services	16
Using Bitmap Volumes	18
Suggested Location of Bitmap Volumes	18
Size Requirements for Bitmap Volumes	18
Using a Volume Set File	20
Backing Up Configuration Information	22
▼ To Back Up Configuration Information	22
3. Configuring the Remote Mirror Software	23
Replication	24
Synchronous Replication	24
Asynchronous Replication	25
Consistency Groups	26
Planning for Remote Replication	27
Business Needs	27
Application Write Load	27
Network Characteristics	28
Configuring the Asynchronous Queue	28
Disk or Memory Queue	28
Queue Size	29
▼ To Resize a Queue	30

- ▼ To Display the Current Queue Size, Length, and hwm 30
- Setting the Correct Size for Disk-Based Asynchronous Queue 33
- Configuring Asynchronous Queue Flusher Threads 34
- Network Tuning 36
 - TCP Buffer Size 36
 - Viewing and Tuning TCP Buffer Sizes 37
 - Network Tuning To View TCP Buffers and Values 37
 - ▼ To View all TCP buffers 37
 - ▼ To View settings by buffer name 37
 - ▼ To View Buffer Sizes for a Socket 38
 - ▼ To Set and Verify the Buffer Size in a Startup Script 38
 - Remote Mirror's Use of TCP/IP ports 39
 - Default TCP Listening Port 39
 - Using Remote Mirror With a Firewall 40
 - Remote Mirror Software With Point-in-Time Copy Software 40
 - Remote Replication Configurations 41
- A. Glossary 43**
- Index 47**

Preface

The *Sun StorageTek Availability Suite 4.0 Software Installation and Configuration Guide* provides information for the efficient installation, set-up, and use of the software.

How This Book Is Organized

This book contains the following chapters:

- [Chapter 1](#) provides information on upgrading, installing, and uninstalling Availability Suite software.
 - [Chapter 2](#) describes describes the required initial configuration procedures *before* you use the Sun StorageTek™ Availability Suite software for the first time.
 - [Chapter 3](#) discusses configuration issues for the Remote Mirror 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. Refer to the following for this information:

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

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	#

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> Password:
<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> .

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

Related Documentation

Application	Title	Part Number
Man pages	sndradm iiadm dsstat kstat svadm dscfgadm	N/A
System administration	<i>Sun StorageTek Availability Suite 4.0 Point-in-Time Copy Software Administration Guide</i>	819-6149
	<i>Sun StorageTek Availability Suite 4.0 Remote Mirror Software Administration Guide</i>	819-6148
Integration	<i>Sun Cluster and Sun StorageTek Availability Suite 4.0 Software Integration Guide</i>	819-6150
Troubleshooting	<i>Sun StorageTek Availability Suite 4.0 Software Troubleshooting Guide</i>	819-6151
Release Notes	<i>Sun StorageTek Availability Suite 4.0 Software Release Notes</i>	819-6152

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 Software Installation and Configuration Guide, part number 819-6147-10

Upgrading, Installing, and Uninstalling Availability Suite Software

This chapter provides information on upgrading, installing, and uninstalling Availability Suite (AVS) software on Sun Solaris 10 Operating Environments, and subsequent update releases.

Note – This chapter applies only to versions of the Availability Suite that are not bundled in the Solaris Operating Environment (OE).

This chapter discusses the following topics:

- [“Upgrading Availability Suite Software” on page 2](#)
- [“Installing Availability Suite Software” on page 3](#)
- [“Uninstalling Availability Suite Software” on page 4](#)

Upgrading Availability Suite Software

AVS 4.0 only supports upgrades from AVS 3.2 running on Solaris 8 or 9. Since AVS 3.2 does not run on Solaris 10, and AVS 4.0 runs only on Solaris 10, upgrading a system from AVS 3.2 to AVS 4.0 requires that you first perform an upgrade of the Solaris operating environment to Solaris 10.

Following a successful Solaris OE upgrade to Solaris 10, any volumes formerly under AVS 3.2 control will be automatically enabled following the installation of the new AVS packages on Solaris 10.

▼ To Upgrade from AVS 3.2

To upgrade from AVS 3.2 perform the following steps:

1. **As a precaution, save the information contained in the `dscfg` database to a file in a remote location:**

```
# dscfg -l > remote-node:/backup/database-file
```

This backup file contains a listing of volumes under the control of AVS 3.2 that can be used to manually reconstruct AVS sets if the Solaris OE upgrade (step 3, below) fails. For example, if the controller numbers change after the Solaris upgrade, AVS sets using raw slices may need to be reconfigured with the updated controller numbers.

2. **Remove AVS 3.2 packages, according to the instructions in [“Uninstalling Availability Suite Software”](#) on page 4.**

3. **Perform an operating environment upgrade to Solaris 10 OE or later.**

Systems running Solaris 8 may be upgraded directly to Solaris 10 without an intermediate upgrade to Solaris 9. Please note that a clean installation of Solaris 10 OE or later on the system is not considered an upgrade.

4. **Install the new AVS packages, according to the instructions in [“Installing Availability Suite Software”](#) on page 3.**

Installing Availability Suite Software

This section provides information on installing AVS 4.0 software.

▼ To Install AVS 4.0

To install AVS 4.0 perform the following steps:

1. **If you plan to run AVS in a Sun Cluster OE, it is recommended that you install the Sun Cluster OE prior to installing AVS. If you choose to install the Sun Cluster OE on a system that already has AVS installed, there is no need to uninstall AVS.**
2. **After installing Sun Cluster OE, run `dscfgadm` to select a Sun Cluster-specific configuration location.**
3. **Use `pkgadd (1M)` to install the new AVS packages in the following order:**
 - SUNWscmr
 - SUNWscmu
 - SUNWspsvr
 - SUNWspsvu
 - SUNWiir
 - SUNWiiu
 - SUNWrdcr
 - SUNWrdcu
4. **Prior to using AVS for the first time, follow the steps in [Chapter 2, “Initial Configuration Procedures”](#) on page 5.**

Note – The installation of the Availability Suite software creates the file reconfiguration in the root directory. However, no reconfiguration reboot of Solaris is require in order to use the Availability Suite software.

Uninstalling Availability Suite Software

This section provides information on uninstalling AVS 3.2 software.

▼ To Uninstall AVS 3.2

To uninstall AVS 3.2 perform the following steps:

1. **Quiesce writes from any applications writing to AVS-enabled volumes.**
2. **Use `pkgrm(1M)` to uninstall AVS packages in the following order:**

- SUNWrdcu
- SUNWrdcr
- SUNWiiu
- SUNWiir
- SUNWspsvu
- SUNWspsvr
- SUNWscmu
- SUNWscmr

Initial Configuration Procedures

After you install the Sun StorageTek Availability Suite software and *before* you use it for the first time, you must configure certain files for the Point-in-Time Copy software and the Remote Mirror software. This chapter describes the required initial configuration procedures:

- [“Overview of Initial Configuration Steps” on page 6](#)
- [“Configuring System Files” on page 7](#)
- [“Modifying Settings” on page 13](#)
- [“Using Bitmap Volumes” on page 18](#)

This chapter also describes the following topics for your information:

- [“Using a Volume Set File” on page 20](#)
- [“Initializing the Configuration Database and Starting Services Using `dscfgadm`” on page 15](#)
- [“Backing Up Configuration Information” on page 22](#)

Overview of Initial Configuration Steps

TABLE 2-1 summarizes the required and optional initial configuration tasks.

TABLE 2-1 Initial Configuration Summary for the Availability Suite Software

Tasks	Instructions
1. Configure the following files: <ul style="list-style-type: none">• /etc/hosts• IP stack (IPv4 and IPv6)• (Optional) /etc/services• /etc/nsswitch.conf• (Optional) /usr/kernel/drv/rdc.conf	“Configuring System Files” on page 7
2. (Optional) Adjust the default number of volumes configured for use by the software.	“Modifying Settings” on page 13
3. (Optional) Tune the asynchronous queue.	<i>Sun StorageTek Availability Suite 4.0 Remote Mirror Software Administration Guide</i>
4. Choose the bitmap volumes.	“Using Bitmap Volumes” on page 18
5. (Optional) Set up an optional Remote Mirror volume configuration file.	“Using a Volume Set File” on page 20

Configuring System Files

This section describes how to edit and check the following system files so that the Sun StorageTek Remote Mirror software runs properly:

- [“To Edit the /etc/hosts File” on page 7](#)
- [“To Set Up an IPv6 Address” on page 8](#)
- [“To Edit the /etc/services File” on page 11](#)
- [“To Edit the /etc/nsswitch.conf File” on page 12](#)
- [“To Back Up Configuration Information” on page 22](#)

▼ To Edit the /etc/hosts File

This ensures that the host names in the `/etc/hosts` file are read and known by machines running the Remote Mirror software.

- **Add the names and IP addresses of all machines you plan to use with the Remote Mirror software to the `/etc/hosts` file.**

Edit this file on each machine where you are installing and running the Remote Mirror software.

Configuring the IP Stack (IPv4 and IPv6)

If you use the Internet Protocol version 6 (IPv6) transport protocol for replication, configure the IPv4 and IPv6 stack concurrently on the host for the interface where the Remote Mirror software is used. The IPv6 protocol provides increased addressability. For the Solaris 10 OS, see the *System Administration Guide: IP Services* for more information about IPv6.

To use the IPv6 protocol, define the IPv4 and IPv6 interfaces with the same name. You must define the primary and secondary hosts such that the same transport protocol is used by both machines.

▼ To Set Up an IPv6 Address

This example procedure shows how to set your network interface to use IPv6 addresses. Use this procedure to test your Remote Mirror hosts connection. The following procedure assumes this configuration information:

Network interface	hme1
Primary host interface name	sndrpri
Secondary host interface name	sndrsec

1. Use a text editor to create the `/etc/hostname6.hme1` file on both the primary host and the secondary host. On the primary host, add the interface name `sndrpri` to the file. On the secondary host, add the interface name `sndrsec` to the file. Save and close the files.

```
primary-host# more /etc/hostname6.hme1
sndrpri
secondary-host# more /etc/hostname6.hme1
sndrsec
```

2. Shut down and restart both machines to activate IPv6.

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

3. After both machines reboot, get the IPv6 `inet` address for the `hme1` interface address.

4. In the following example, the address is fe80::a00:20ff:febd:c33f/128

```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 2
    inet 127.0.0.1 netmask ffffffff
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 192.9.200.125 netmask ffffffff broadcast 192.9.200.255
    ether 8:0:20:ae:85:fa
lo0: flags=2000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6> mtu 8252 index 2
    inet6 ::1/128
hme0: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 3
    ether 8:0:20:ae:85:fa
    inet6 fe80::a00:20ff:feae:85fa/10
hme1: flags=2000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 4
    ether 8:0:20:bd:c3:3f
    inet6 fe80::a00:20ff:febd:c33f/128
```

5. Edit the `/etc/inet/ipnodes` file and add the inet address from [Step 4](#), assigning the primary host address to `sndrpri` and the secondary host address to `sndrsec`. Do not include the `/128` portion of the address.

Note – Ensure that the `/etc/inet/ipnodes` file on each system running the Remote Mirror software contains the IPv6 inet address and names of each system.

6. Save and close the file and then check the file contents.

In the following example, `sndrsec` is the secondary host interface name.

```
primary-host# more /etc/inet/ipnodes
#
# Internet host table
#
::1                localhost
127.0.0.1          localhost
fe80::a00:20ff:febd:c33f      sndrpri
fe80::a00:20ff:fee1:195e      sndrsec
```

7. **Edit the `/etc/nsswitch.conf` file to make sure that `ipnodes:` points to `files`.**
Look for the following text in the file and make sure the `ipnodes:` line is uncommented.

```
# consult /etc "files" only if nis is down.  
hosts: files nis [NOTFOUND=return] files  
ipnodes: files
```

8. **Add the host names and IPv6 inet primary addresses of all machines you plan to use with the Remote Mirror software to the `/etc/hosts` file on each machine.**

Edit this file on each machine where you are installing and running the Remote Mirror software.

Note – If you do not complete this step (described in [“To Edit the `/etc/hosts` File” on page 7](#)), the following error message is displayed when you enable the Remote Mirror software:

```
sndradm: Error: neither sndrpri nor sndrsec is local
```

9. **Ensure that one system can ping another and that these systems are using the IPv6 protocol.**

To ping from the primary host, enter the following:

```
# ping -s sndrsec  
PING sndrsec: 56 data bytes  
64 bytes from sndrsec (fe80::a00:20ff:fe1:195e): icmp_seq=0. time=0. ms  
64 bytes from sndrsec (fe80::a00:20ff:fe1:195e): icmp_seq=1. time=0. ms  
64 bytes from sndrsec (fe80::a00:20ff:fe1:195e): icmp_seq=2. time=0. ms
```

To ping from the secondary host, enter the following:

```
# ping -s sndrpri  
PING sndrpri: 56 data bytes  
64 bytes from sndrpri (fe80::a00:20ff:febd:c33f): icmp_seq=0. time=0. ms  
64 bytes from sndrpri (fe80::a00:20ff:febd:c33f): icmp_seq=1. time=0. ms  
64 bytes from sndrpri (fe80::a00:20ff:febd:c33f): icmp_seq=2. time=0. ms
```

10. **Use the `netstat(1M)` command to verify that the interface has the correct IPv6 address and IPv6 name.**

Use this command on both the `sndrpri` and `sndrsec` hosts. For example:

```
# netstat -in
Name Mtu Net/Dest      Address      Ipkts  Ierrs Opkts  Oerrs Collis Queue
lo0  8232 127.0.0.0    127.0.0.1   3844   0     3844   0     0     0
hme0 1500 192.0.0.0    192.9.200.225 22007  0     1054   0
0     0

Name Mtu Net/Dest      Address      Ipkts  Ierrs Opkts  Oerrs Collis Queue
lo0  8252 :::1          :::1         3844   0
3844 0     0
hme1 1500 fe80::a00:20ff:febd:c33f fe80::a00:20ff:febd:c33f 43 0 65 0 0
```

```
# netstat -i
Name Mtu Net/Dest      Address      Ipkts  Ierrs Opkts  Oerrs Collis Queue
lo0  8232 loopback     localhost    3844   0     3844   0     0     0
hme0 1500 arpanet      rick1        22038  0
1067 0     0     0

Name Mtu Net/Dest      Address      Ipkts  Ierrs
Opkts Oerrs Collis
lo0  8252 localhost localhost    3844   0     3844   0     0
hme1 1500 sndrpri  sndrpri      43     0     65
0     0
```

▼ To Edit the `/etc/services` File

Port 121 is the default port for use by the Remote Mirror `rdc` daemon.

```
# cat /etc/services
...
rdc 121/tcp # SNDR server daemon
...
```

If you change the port number, you must change it on all Remote Mirror hosts within this configuration set (that is, primary and secondary hosts, and all hosts in one-to-many, many-to-one, and multihop configurations).

1. Edit the `/etc/services` file on each machine running the Remote Mirror software.
2. Shutdown and restart all hosts to make that the new port number take effect.

▼ To Edit the /etc/nsswitch.conf File

If the file includes the `hosts:` and `services:` entries, verify that `files` is placed before `nis`, `nisplus`, `ldap`, `dns`, or any other service that the machine is using. For example, for systems using the NIS naming service, the file contains the following lines:

```
hosts: files nis
services: files nis
```

- **If the host and service entries are not correct, edit the file and save it.**

If you are using the IPv6 protocol, see the changes for this file in [“Configuring the IP Stack \(IPv4 and IPv6\)”](#) on page 8.

Modifying Settings

The following sections describe how to modify the Remote Mirror software settings.

- [“Setting the Bitmap Operation Mode” on page 13](#)
- [“Increasing the Number of Volume Sets” on page 14](#)
- [“Increasing the Storage Volume Device Limit” on page 14](#)

Note – After editing the files in this section, shut down and restart your server using the `shutdown` command for changes to take effect. If you edit the `rdc.conf` file to use more than 64 volume sets, ensure that you have enough system resources such as a large swap space.

Setting the Bitmap Operation Mode

A bitmap maintained on disk can persist across a system crash, depending on the setting of `rdc_bitmap_mode` in `/usr/kernel/drv/rdc.conf`. The default setting is 1 (force bitmap writes for every write operation).

- **Edit the `rdc.conf` file and locate the following section. Edit the value for the bitmap mode, save the file, and close it.**

```
#
# rdc_bitmap_mode
# - Sets the mode of the RDC bitmap operation, acceptable values are:
# 0 - autodetect bitmap mode depending on the state of SDBC (default).
# 1 - force bitmap writes for every write operation, so an update resync
#    can be performed after a crash or reboot.
# 2 - only write the bitmap on shutdown, so a full resync is
#    required after a crash, but an update resync is required after
#    a reboot.
#
rdc_bitmap_mode=1;
```

Increasing the Number of Volume Sets

The default number of configured volume sets is 64. To configure more than 64 volume sets, edit the `rdc_max_sets` field in the `/usr/kernel/drv/rdc.conf` file on each machine running the Remote Mirror software.

- **Edit the `rdc.conf` file and locate the following section. Edit the value for the volume sets, save the file, and close it.**

For example, to use 128 sets, change the file as shown in the following section:

```
#  
# rdc_max_sets  
# - Configure the maximum number of RDC sets that can be enabled on  
# this host. The actual maximum number of sets that can be enabled  
# will be the minimum of this value and nsc_max_devices (see  
# nsctl.conf) at the time the rdc kernel module is loaded.  
#  
rdc_max_sets=128;
```

Increasing the Storage Volume Device Limit

The Availability Suite software has a default limit of 4096 storage volumes. The default number of storage volume driver devices (that is, volumes) is set by the `nsc_max_devices` value in the `nsctl.conf` file.

The number of volumes is divided for use between the Remote Mirror and Point-in-Time Copy software. For example, if you use the point-in-time copy software only, you can have 1365 volume sets, each consisting of master, shadow, and bitmap volumes. If you use the Remote Mirror and Point-in-Time Copy software packages together, the number of volume sets is divided between the two packages.

Some installations might benefit from changing this limit. Sites with plenty of available memory can increase the limit if necessary to enable more storage volumes. Sites with limited available memory might benefit from lowering this limit, thus freeing up system resources.

- **Edit the `nsctl.conf` file and locate the `nsc_max_devices` field. Edit the value, save the file, and close it.**

Initializing the Configuration Database and Starting Services Using `dscfgadm`

The startup and shutdown of the Availability Suite software is implemented through Service Management Facility (SMF) services, which may be administered using the `dscfgadm` utility.

```
# svcs | grep nws_
online      Mar_14   svc:/system/nws_scm:default
online      Mar_14   svc:/system/nws_sv:default
online      Mar_14   svc:/system/nws_ii:default
online      Mar_14   svc:/system/nws_rdc:default
online      Mar_14   svc:/system/nws_rdcsyncd:default
```

The `dscfgadm` Utility

`dscfgadm` controls the Availability Suite configuration services by providing facilities to set the configuration location, and to enable and disable the Availability Suite services.

```
# dscfgadm [-x]
```

Usage

```
dscfgadm [-x ]
dscfgadm [-x ] -i
dscfgadm [-x ] -e [-r] [-p]
dscfgadm [-x ] -d [-r]
```

Options

- i displays information on the Availability Suite services
- e enables the Availability Suite SMF services (all, by default)
- d disables the Availability Suite SMF services (all, by default)

- r enables/disables the Remote Mirror software
- p enables the Point-in-Time Copy software
- x displays verbose debugging information

Initializing the Configuration Database and Starting Services

By default, Availability Suite services are not started, and no Availability Suite configuration database is present on the system. When `dscfgadm` is executed with no options (or with only `-x`), it runs in interactive mode. This mode allows you to initialize the local configuration database required by the Availability Suite software, and gives you the choice of starting the Availability Suite services at that time.

If you choose not to start the Availability Suite SMF services during the initialization of the database, you may start the SMF services later using the `dscfgadm -e` command.

```
# dscfgadm -e
```

Note – The Availability Suite software is unusable until the Availability Suite services have been started.

Enabling or Disabling Services

To enable or disable services at a later time, use the `-e` and `-d` options with `dscfgadm`.

```
# dscfgadm -e
```

```
# dscfgadm -d
```

The default behavior acts upon all services, but the `-r` and `-p` options allow selectively disabling or enabling the Remote Mirror or Point-in-Time Copy services, respectively.

Note – Be sure to quiesce all application writes to the volumes in use by a service before disabling that service.

Note – Settings are persistent across system boots.

Using Bitmap Volumes

Both Point-in-Time Copy software and Remote Mirror software use raw volumes to store bitmaps. Bitmap files are not supported.

Suggested Location of Bitmap Volumes

Store bitmap raw volumes on a disk separate from the disk that contains its associated master and shadow volumes (for Point-in-Time Copy software) for replicated volumes (for Remote Mirror software). Configure RAID (such as mirrored partitions) for these bitmap volumes and ensure that the mirrored members are not stored on the same disk as the master and shadow volumes or replicated volumes.

When you use Point-in-Time Copy software in a clustered environment, the bitmap volume must be part of the same disk group or cluster resource group as the corresponding master or shadow volume.

Size Requirements for Bitmap Volumes

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).

- Independent or dependent shadow volume sets require:

8 KBytes per 1 GBytes 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 $(3 \times 8 \text{ KBytes}) + 24 \text{ KBytes}$, or 48 KBytes in size. A 50 GByte master volume requires a 424 KByte bitmap volume.

- Compact dependent shadow volume sets require:

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 KBytes 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.

The Availability Suite software provides the `dsbitmap` utility to calculate the required size of a bitmap for a Point-in-Time Copy shadow volume set or a Remote Mirror volume set.

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

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

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

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

Refer to the `dsbitmap(1SCM)` man page for more information on the `dsbitmap` utility.

Using a Volume Set File

When you enable the Remote Mirror software, you can specify an optional *volume set file* containing information about the volume set: volumes, primary and secondary hosts, bitmaps, operating mode, and so on. Use the `snradm -f volset-file` option when you use a volume set file.

You can also enter information about each volume set from the command line, but it is convenient to put this information in a file when you have multiple volume sets. Another advantage is that you can operate on specific volume sets, excluding other sets from the operation. Unlike adding the volume sets to an I/O group, you can mix replication modes in a volume set file. The fields for the volume set file specified are:

```
phost pdev pbitmap shost sdev sbitmap ip {sync|async} [g io-groupname] [C tag] [q qdev]
```

[TABLE 2-2](#) describes these fields. See the `rdc.cf` man page for more information about the volume set file format.

The following shows an example file entry:

```
atm10 /dev/vx/rdisk/oracle816/oratest /dev/vx/rdisk/oracle816/oratest_bm \  
atm20 /dev/vx/rdisk/oracle816/oratest /dev/vx/rdisk/oracle816/oratest_bm \  
ip sync g oragroup
```

TABLE 2-2 Fields for the Volume Set File

Field	Meaning	Description
<i>phost</i>	Primary host	Server on which the primary volume resides.
<i>pdev</i>	Primary device	Primary volume partition. Specify full path names only (for example, /dev/rdisk/c0t1d0s4).
<i>pbitmap</i>	Primary bitmap	Volume partition in which the bitmap of the primary partition is stored. Specify full path names only.
<i>shost</i>	Secondary host	Server on which the secondary volume resides.
<i>sdev</i>	Secondary device	Secondary volume partition. Specify full path names only.
<i>sbitmap</i>	Secondary bitmap	Volume partition in which the bitmap of the secondary partition is stored. Specify full path names only.
<i>ip</i>	Network transfer protocol	Specify ip.
<i>sync async</i>	Operating mode	<ul style="list-style-type: none"> • <i>sync</i> is the mode in which the I/O operation is confirmed as complete when the remote volume has been updated. • <i>async</i> is the mode in which the primary host I/O operation is confirmed as complete before updating the remote volume.
<i>g io-groupname</i>	I/O group name	I/O group name that can be specified using the <i>g</i> character. In the example, the name is <i>oragroup</i> .
<i>C tag</i>	Cluster tag	Tag that limits operations to only those Remote Mirror sets belonging to the cluster resource group.
<i>q qdev</i>	Disk queue volume	Volume to be used as a disk-based I/O queue for an asynchronous set or group. You must specify the full path name. For example: /dev/rdisk/c1t2d0s6.

Backing Up Configuration Information

You must back up your Sun StorageTek, VERITAS Volume Manager, and Solaris Volume Manager configuration information on a regular basis. To make any volume set-related changes, use the `/usr/sbin/iiaadm` command, described in the *Sun StorageTek Availability Suite 4.0 Point-in-Time Copy Software Administration Guide*, and consider the following:

- Place backup commands in a shell script and run the script as part of a daily `cron(1M)` job
- Store the output of the commands in a location that is backed up to tape routinely.

You may wish to set up a `cron(1M)` job to periodically back up the configuration database automatically. In addition, back up the configuration information whenever you change your configuration, for example, adding and deleting volumes.

▼ To Back Up Configuration Information

- **Copy the configuration database (`/etc/dscfg_local`) to a safe location.**

```
# cp /etc/dscfg_local /var/backups/dscfg_db
```


Configuring the Remote Mirror Software

The Sun StorageTek Availability Suite Remote Mirror software is a volume-level *replication* facility for the Solaris 10 (Update 1 and higher) Operating Systems. The Remote Mirror software replicates disk volume write operations between physically separate *primary* and *secondary* sites in real time. The Remote Mirror software can be used with any Sun network adapter and network link that supports TCP/IP.

Since the software is volume-based, it is storage-independent and supports raw volumes or any volume manager, for both Sun and third-party products. Additionally, the product supports any application or database that has a single host running the Solaris OS that writes data. Databases, applications, or file systems that are configured to allow multiple hosts running the Solaris OS to write data to a shared volume are not supported (for example: Oracle® 9iRAC, Oracle® Parallel Server).

As part of a disaster recovery and business continuance plan, the Remote Mirror software keeps up-to-date copies of critical data at remote sites. The Remote Mirror software enables you to rehearse and test business continuance plans. For a highly available solution, the Sun StorageTek Availability Suite software can be configured to failover within Sun™ Cluster 3.x environments.

The Remote Mirror software is active while your applications are accessing the data volumes, replicating the data continually to the remote sites or scoreboarding changes, which allow for a fast resynchronization at a later time.

The Remote Mirror software enables you to initiate resynchronization manually from either the primary site to the secondary site (typically called *forward synchronization*), or from the secondary site to the primary site (typically called *reverse synchronization*).

Replication and configuration in the Remote Mirror software is done on a set basis. A Remote Mirror set consists of a primary volume, a secondary volume, a bitmap volume on both the primary and secondary sites (used to track and scoreboard changes for fast resynchronization), and an optional *asynchronous queue* volume for

asynchronous replication mode. The primary and secondary volumes are recommended to be the same size. You can use the `dsbitmap` tool to determine the required size of the bitmap volumes. For more information on configuring Remote Mirror sets or the `dsbitmap` tool see the *Sun StorageTek Availability Suite 4.0 Remote Mirror Software Administration Guide*.

Replication

Replication can occur either synchronously or asynchronously. In synchronous mode, an application write operation is not acknowledged until the write operation is committed on both the primary and secondary hosts. In asynchronous mode, the application write operation is acknowledged when it is committed to storage locally and written to an asynchronous queue. This queue drives write operations to the secondary site asynchronously.

Synchronous Replication

The data flow for synchronous operation is as follows:

1. Scoreboard bit is set in the bitmap volume.
2. Local write operation and network write operation are initiated in parallel
3. When both write operations are complete, the scoreboard bit is cleared (*lazy clear*).
4. Write operation is acknowledged to application.

The advantage of *synchronous replication* is that both primary and secondary sites are always in sync. This type of replication is practical only if the latency of the link is low and the bandwidth requirements of the application can be met by the link. These constraints usually confine a synchronous solution to a campus or metropolitan location.

In this case, the average service time for a write operation is:

bitmap write + MAX (local data write, network round trip + remote data write)

In the campus and metropolitan location, the network round trip is negligible and the service time is approximately twice what is observed when the Remote Mirror software is not installed.

Assuming 5 milliseconds for a write, then:

$5\text{ms} + \text{MAX}(5\text{ms}, 1\text{ms} + 5\text{ms}) = 11\text{ms}$

Note – This value of 5 milliseconds is a reasonable assumption on a lightly loaded system. On a more realistically loaded system, queuing backlog increases the value.

However, if the network round trip is approximately 50 milliseconds (typical for long distance replication), the network latency renders the synchronous solution impractical as shown in the following example:

$$5\text{ms} + \text{MAX}(5\text{ms}, 50\text{ms} + 5\text{ms}) = 60\text{ms}$$

Asynchronous Replication

Asynchronous replication separates the remote write operation from the application write operation. In this mode, the acknowledgment occurs when the network write operation is added to the asynchronous queue. This means that the secondary site can get out of synchronization with the primary site until all write operations are delivered to the secondary site. In this mode, data flows in this manner:

1. Scoreboard bit is set.
2. Local write, asynchronous queue write operations are done in parallel.
3. Write is acknowledged to application.
4. Flusher threads read asynchronous queue entry and perform network write.
5. Scoreboard bit is cleared (lazy clear).

The service time is the time required for the following:

$$\text{bitmap write} + \text{MAX}(\text{local write, asynchronous queue entry data})$$

Using the value of 5 milliseconds service time for a write operation, the estimated service time for an asynchronous write operation is:

$$5\text{ms} + \text{MAX}(5\text{ms}, 5\text{ms}) = 10\text{ms}$$

If the network drain rate for the volume or *consistency group* is exceeded by the write rate for an extended period of time, the asynchronous queue fills up. Proper sizing is important so a method for estimating the appropriate volume size is discussed later in this document.

There are two modes that govern how the Remote Mirror software behaves in the event of the asynchronous disk queue filling;

- Blocking mode

In blocking mode, which is the default setting, the Remote Mirror software blocks, and waits for the asynchronous disk queue to drain to a certain point before adding the write to the asynchronous queue. This impacts application write operations, but maintains write ordering across the link.

- Non-blocking mode

In non-blocking mode (not available with memory-based queues), the Remote Mirror software does not block when the disk asynchronous queue fills, but drops into *logging* mode and scoreboards the write. On a subsequent *update synchronization* these are read from bit 0 forward, and there is no preservation of write ordering. If this mode is used and if the asynchronous disk queue fills and write ordering is lost, the associated volume or consistency group is inconsistent.

Note – It is strongly advised that a point-in-time copy be taken on the secondary site prior to starting the update synchronization, for example, using the `autosync` daemon.

Consistency Groups

In synchronous mode, write ordering for an application that spans many volumes is assured because the application waits for completion before issuing another I/O operation when ordering is required, and the Remote Mirror software does not signal completion until the write operation is on both primary and secondary sites.

In asynchronous mode, by default, the queue for each volume is drained by one or more independent threads. Because this operation is separated from the application, write ordering is not preserved across write operations to multiple volumes.

If write ordering is required for an application, the Remote Mirror software provides the consistency group feature. Each consistency group has a single network queue, and although multiple write operations are allowed in parallel, write ordering is preserved through the use of sequence numbers.

Planning for Remote Replication

When you are planning for remote replication, consider your business needs, the application write loads, and your network's characteristics.

Business Needs

When you decide to replicate your business data, consider the maximum delay. How long out of date can you allow the data on the secondary site to become? This determines the replication mode and snapshot scheduling. Additionally, it is very important to know if the applications that you are replicating require the write operations to the secondary volume to be replicated in the correct order.

Application Write Load

Understanding the average and peak write loads is critical to determining the type of network connection required between the primary and secondary sites. To make decisions about the configuration, collect the following information:

- The average rate and size of data write operations
The average rate is the amount of data write operations while the application is under typical load. Application read operations are not important to the provisioning and planning of your remote replication.
- The peak rate and size of data write operations
The peak rate is the largest amount of data written by the application over a measured duration.
- The duration and frequency of the peak write rate
The duration is how long the peak write rate lasts and the frequency is how often this condition occurs.

If these application characteristics are not known, you can measure them using tools such as `iostat` or `sar` to measure write traffic while the application is running.

Network Characteristics

When you know the application write load, determine the requirements of the network link. The most important network properties to consider are the network bandwidth and the network latency between the primary and secondary sites. If the network link already exists prior to installing the Sun StorageTek Availability Suite software, you can use tools such as `ping` to help determine the characteristics of the link between the sites.

To use synchronous replication, the network latency must be low enough that your application response time is not affected dramatically by the time of the network round trip of each write operation. Also, the bandwidth of the network must be sufficient to handle the amount of write traffic generated during the application's peak write period. If the network cannot handle the write traffic at any time, the application response time will be impacted.

To use asynchronous replication, the bandwidth of the network link must be able to handle the write traffic generated during the application's average write period. During the application peak write phase, the excess write operations are written to the local asynchronous queue and then written to the secondary site at a later time when the network traffic allows. The application response time can be minimized during bursts of write traffic above the network limit as long as the asynchronous queue is properly sized.

See the [“Configuring the Asynchronous Queue” on page 28](#) section of this document. The Remote Mirror asynchronous option mode selected (blocking or non-blocking) determines how the software reacts to the queue filling.

Configuring the Asynchronous Queue

If you use asynchronous replication, plan for the configuration settings described in this section. These settings are set on a Remote Mirror set or consistency group basis.

Disk or Memory Queue

In version 3.2 of the software, the Remote Mirror software added support for disk-based asynchronous queues. For ease of upgrading from previous versions, memory-based queues are still supported, but the new disk-based queues provide the ability to create significantly larger, more efficient queues. Larger queues allow for larger bursts of write activity without affecting application response time. Also, disk-based queues have less impact to system resources than the memory-based queues.

The asynchronous queue must be sufficient in size to handle the bursts of write traffic associated with the application peak write periods. A large queue can handle prolonged bursts of write activity, but also allows the possibility that the secondary site gets further out of sync with the primary. Using the peak write rate, peak write duration, write size, and network link characteristics, you can determine how the queue should be sized. See [“Setting the Correct Size for Disk-Based Asynchronous Queue” on page 33](#).

The queue option you select (blocking or non-blocking) determines how the software reacts to a filled disk queue. Use the `dsstat` tool to determine statistics for the asynchronous queue, including the high-water mark (`hwm`), which shows the largest amount of the queue that has been used. To add an asynchronous queue to a Remote Mirror set or consistency group, use the `sndradm` command with the `-q` option: `sndradm -q a`

Queue Size

Monitor the asynchronous queue using the `dsstat(1SCM)` command to check the *high water mark* (`hwm`). If the `hwm`, caused by the application writes more data than the queue can handle, reaches 80 to 85 percent of the total size of the queue frequently, increase the queue size. This principle applies both to disk-based and memory-based queues. However, the procedure to resize each queue type is different.

Memory-Based Queue

- The default maximum number of write operations in the queue (tunable) is 4096. Use the `sndradm -W` command to change this value.
- The default maximum number of 512-byte data blocks (default queue size) (tunable) is 16384, which is about 8 Mbytes of data. Use the `sndradm -F` command to change this value.

Disk-Based Queue

The effective size of the disk queue is the size of the disk queue volume. A disk queue can only be resized by replacing it with a volume of a different size. For example, for a queue size of 16384 blocks, check that the `hwm` does not exceed 13000 to 14000 blocks. If it exceeds this amount, resize the queue using the following procedure.

▼ To Resize a Queue

1. Place the volume into logging mode using the `sndradm -l` command.
2. Resize the queue.
 - Memory-based: use the `sndradm -F` command.
 - Disk-based: replace the existing disk queue volume with a volume of larger size using the `sndradm -q` command.
3. Perform an update synchronization by using the `sndradm -u` command.

▼ To Display the Current Queue Size, Length, and hwm

1. Type the following to display the queue size:
 - Memory-based:

```
# sndradm -P
/dev/vx/rdisk/data_t3_dg/vol0 -> priv-2-
230:/dev/vx/rdisk/data_t3_dg/vol0
autosync: off, max q writes: 4096, max q fbas: 16384, async
threads: 8, mode: async, state: replicating
```

The size of the queue in blocks is given by `max q fbas` (16384 blocks in this example). The maximum number of items allowed in the queue is given by `max q writes` (4096 in this example). In this example, this means that the average size of an item in the queue is 2K.

- Disk-based:

```
# sndradm -P
/dev/vx/rdisk/data_t3_dg/vol0 -> priv-
230:/dev/vx/rdisk/data_t3_dg/vol0
autosync: off, max q writes: 4096, max q fbas: 16384, async
threads: 1, mode: async, blocking diskqueue:
/dev/vx/rdisk/data_t3_dg/dq_single, state: replicating
```

The diskqueue volume is displayed (`/dev/vx/rdisk/data_t3_dg/dq_single`). The size of the queue can be determined by examining the size of the volume.

2. Type the following to show the current queue length and its hwm:

```
# dsstat -m sndr -d q
name          q role   qi    qk   qhwi  qhwk
data_a5k_dg/vol0 D net   4     13    5    118
```

Where:

- `qi` is the current number of items in the queue
- `qk` is the current total data size in the queue (in Kbytes)
- `qhwi` is the maximum number of items that have ever been in the queue at any one time
- `qhwk` is the maximum data in Kbytes that have ever been in the queue at any one time

3. To show streaming summary and disk queue information, type:

```
# dsstat -m sndr -r bn -d sq 2
```

4. To show more information, run `dsstat(1SCM)` with other display options.

Sample dsstat Output For a Correctly-Sized Queue

Note – This example shows only a portion of the command output required for this section; the `dsstat` command actually displays more information.

The following `dsstat(1SCM)` kernel statistics output shows information about the asynchronous queue. In these examples, the queue is sized correctly and is not currently filled. This example shows the following settings and statistics:

```
# dsstat -m sndr -r n -d sq -s priv-2-230:/dev/vx/rdisk/data_t3_dg/vol167
name          q role   qi    qk   qhwi  qhwk   kps   tps   svt
data_t3_dg/vol167 D net   48    384   240   1944   10    1    54
```

Where:

- The `qi` entry means that a total of 48 write transactions have been put into the queue
- The `qk` entry means that 384 Kbytes have been put into the queue
- The `qhwi` entry shows that the hwm for queued items is 240 items; not currently being reached

- The `qhwk` entry shows that the hwm for queued data (Kbytes) is 1944; not currently being reached

Assuming the disk queue volume size is 1 Gbyte, or 2097152 disk blocks, the hwm of 1944 blocks is well below 80% full. The disk queue is sized correctly for the write load.

Sample dsstat Output for an Incorrectly-Sized Disk Queue

The following `dsstat(1SCM)` kernel statistics output shows information about the asynchronous queue, which is incorrectly sized:

```
# sndradm -P
/dev/vx/rdisk/data_a5k_dg/vol0 -> priv-230:/dev/vx/rdisk/data_a5k_dg/vol0
autosync: off, max q writes: 4096, max q fbas: 16384, async threads: 2, mode:
async, state: replicating

# dsstat -m sndr -d sq
name           q role   qi    qk  qhwi  qhwk   kps   tps   svt
data_a5k_dg/vol0 M net  3609  8060  3613  8184   87    34    57
k/bitmap_dg/vol0  bmp     -     -     -     -     0     0     0
```

This example shows the default queue settings but the application is writing more data than the queue can handle. The `qhwk` value of 8184 Kbytes compared to `max q fbas` of 16384 blocks (8192 Kbytes) indicates that the application is approaching the maximum allowed limit of 512-byte blocks. It is possible that the next few I/O operations are not going to be placed into the queue.

Increasing the queue size would be a solution in this case. However, consider improving the network link (such as using larger bandwidth interfaces) to achieve long-term benefits. Alternatively, consider taking point-in-time volume copies and replicating the shadow volumes. See the *Sun StorageTek Availability Suite 4.0 Point-in-Time Copy Software Administration Guide*.

In summary:

- If the fill rate is less than or equal to the drain rate, the default queue size is sufficient.
- If the drain rate is less than the fill rate, increasing the queue size might provide a temporary solution. However, if the write operations continue for a prolonged period, the queue eventually fills.

Setting the Correct Size for Disk-Based Asynchronous Queue

Consider the following example. In this example, `iostat` was run at an hourly interval to profile the I/O load that will be replicated. In this example, we assume a DS3 (45Mbyte/sec) link. Also assume that this application uses a single consistency group, consequently, a single queue is involved.

After collecting stats over a 24 hour period, and assuming that this is a typical day for the application in question, one may determine the average write rate, the proper sizing for the async queues, how far out of date the remote site may become over the course of the day, and whether the network bandwidth chosen is adequate for this application.

TABLE 3-1 Example for Determining Correct Size for Disk-Based Queue

Time	kwr/s	wr/s	Network Throughput	Queue Growth	Queue Size
	A	B	C	A/1000 - C)*3600	
6am	0	0	4MBps*		
7am	1000	400	4MBps		
8am	2000	1000	4MBps		
9am	2000	1000	4MBps		
10am	4000	1800	4MBps		
11am	5000	2400	4MBps	3.6GB	3.6GB
12pm	1000	400	4MBps	-10GB	
1pm	1200	600	4MBps		
2pm	1000	500	4MBps		
3pm	1200	400	4MBps		
4pm	2000	600	4MBps		
5pm	1000		4MBps		
6pm	800		4MBps		
7pm	800		4MBps		
8pm	3200	1000	4MBps		
9pm	8000	2500	4MBps	14GB	14GB
10pm	8000	2500	4MBps	14GB	28GB
11pm	1000	400	4MBps	-10	18

TABLE 3-1 Example for Determining Correct Size for Disk-Based Queue

Time	kwr/s	wr/s	Network Throughput	Queue Growth	Queue Size
12pm	0		4MBps	-14	4
1am	0		4MBps	-14	
2am	0		4MBps		
3am	0		4MBps		
4am	0		4MBps		
5am	0		4MBps		
Average bandwidth	1.8MBps				

* megabytes/second

After filling in the table and calculating queue growth and size, it is evident that a 30Gbyte queue is sufficient. Although the queue grows large and, consequently, the secondary grows out of sync, a batch job run in the evenings ensures that the queue is empty by normal business hours and the two sites are in sync.

This exercise also validates that the network bandwidth is adequate for the write load the application produces.

Configuring Asynchronous Queue Flusher Threads

The Sun StorageTek Availability Suite software provides the ability to set the number of threads flushing the asynchronous queue. Changing this number allows for multiple I/Os per volume or consistency group on the network at one time. The Remote Mirror software on the secondary node handles write ordering the I/Os using sequence numbers.

Many variables must be considered when determining the number of queue flusher threads that is most efficient for your replication configuration. These variables include the number of sets or consistency groups, available system resources, network characteristics, and whether or not there is a file system. If you have a small number of sets or consistency groups, a larger number of flusher threads might be more efficient. It is recommended that you do some basic testing or prototyping with this variable at slightly different values to determine the most efficient setting for your configuration.

Knowledge of the configuration, network characteristics, and operation of the Remote Mirror software can provide guidelines to proper selection of the number of network threads. The Remote Mirror software utilizes Solaris RPCs, which are synchronous, as a transport mechanism. For each network thread, the maximum throughput the individual thread can achieve is I/O size / Round trip time. Consider a workload that is predominately 2Kbyte I/Os, and a round trip time of 60 milliseconds. Each network thread would be capable of:

$$2\text{Kbyte}/0.060\text{sec} = 33\text{Kbyte}/\text{sec}$$

In the case where there is a single volume, or many volumes in a single consistency group, the default of 2 network threads would limit network replication to 66Kbyte/sec. Tuning this number up would be advisable. If the replication network were provisioned for 4Mbyte/sec, then theoretically, the optimal number of network threads for a 2Kbyte workload would be:

$$(4096\text{Kbyte}/\text{sec}) / (2\text{Kbyte}/0.060 \text{ IO}/\text{sec}) = 123 \text{ threads}$$

This assumes linear scalability. In practice it has been observed that adding more than 64 network threads yields no benefit. Consider the case where there is no consistency group, 30 volumes being replicated over a 4Mbyte/sec link, and 8Kbyte I/Os. The default of 2 network threads per volume would yield 60 network threads, and if the workload were spread evenly across these volumes, the theoretical bandwidth would be:

$$60 * (8\text{Kbyte} / 0.060 \text{ IO}/\text{sec}) = 8\text{Mbyte}/\text{sec}$$

This is more than the network bandwidth. No tuning is required.

The default setting for the number of asynchronous queue flusher threads is 2. To change this setting, you would use the `sndradm` command line interface with the `-A` option. The description for the `-A` option is: `sndradm -A` specifies the maximum number of threads that can be created to process the asynchronous queue when a set is replicating in asynchronous mode (default 2).

To determine the number of flusher threads that are currently configured to serve an asynchronous queue, you can use the `sndradm -P` command. For example, you can see that the set below has 2 asynchronous flusher threads configured.

```
# sndradm -P
/dev/md/rdisk/d52 -> lh1:/dev/md/sdsdg/rdsk/d102
autosync: off, max q writes: 4096, max q fbas: 16384, async threads: 2, mode:
async, group: butch, blocking diskqueue: /dev/md/rdisk/d100, state: replicating
```

An example of how to use the `sndradm -A` option to change the number of asynchronous queue flusher threads to 3 is:

```
# sndradm -A 3 lh1:/dev/md/sdsdg/rdsk/d102
```

Network Tuning

The Remote Mirror software injects itself directly into the system's I/O path, monitoring all traffic to determine if it is targeted to Remote Mirror volumes. The I/O commands that are targeted for Remote Mirror volumes are tracked and replication of these write operations is managed. Due to the fact that the Remote Mirror software is directly in the system's I/O path, some performance impact to the system is expected. Additional TCP/IP processing that is required for network replication also consumes host CPU resources. Perform the procedures in this section on the primary and secondary Remote Mirror hosts.

TCP Buffer Size

The *TCP buffer* size is the number of bytes that the transfer control protocol allows to be transferred before it waits for an acknowledgment. To get maximum throughput, it is critical to use optimal TCP send and receive socket buffer sizes for the link you are using. If the buffers are too small, the TCP congestion window will never fully open. If the receiver buffers are too large, TCP flow control breaks and the sender can overrun the receiver, causing the TCP window to shut down. This event is likely to happen if the sending host is faster than the receiving host. Overly large windows on the sending side are not a problem as long as you have excess memory.

Note – Increasing the buffer size to a much higher value over a shared network might impact the network performance. See the *Solaris System Administrator Collection* for information about tuning the size.

TABLE 3-2 shows the maximum possible throughput for a 100BASE-T network.

TABLE 3-2 Network Throughput and Buffer Size

Latency	Buffer Size = 24 Kbytes	Buffer Size = 256 Kbytes
10 milliseconds	18.75 MBps*	100 MBps
20 milliseconds	9.38 MBps	100 MBps
50 milliseconds	3.75 MBps	40 MBps
100 milliseconds	1.88 MBps	20 MBps
200 milliseconds	0.94 MBps	10 MBps

* megabytes/second

Viewing and Tuning TCP Buffer Sizes

You can view and tune your TCP buffer size by using `/usr/bin/netstat(1M)` and `/usr/sbin/ndd(1M)` commands. TCP parameters to consider tuning include:

- `tcp_max_buf`
- `tcp_cwnd_max`
- `tcp_xmit_hiwat`
- `tcp_recv_hiwat`

When you change one of these parameters, restart the Remote Mirror software with the `shutdown` command, allowing the software to use the new buffer size. However, after you shut down and restart your server, the TCP buffers return to a default size. To keep your change, set the values in a startup script as described later in this section.

Network Tuning To View TCP Buffers and Values

Following are procedures to view TCP buffers and values.

▼ To View all TCP buffers

- Type the following to view all TCP buffers:

```
# /usr/sbin/ndd /dev/tcp ? | more
```

▼ To View settings by buffer name

- Type the following to view settings by buffer name:

```
# /usr/sbin/ndd /dev/tcp tcp_max_buf
1073741824
```

This command shows a value of 1073741824.

▼ To View Buffer Sizes for a Socket

- Use the `/usr/bin/netstat(1M)` command to view the buffer size for a particular network socket.

For example, view the size for port 121, the default Remote Mirror port:

```
# netstat -na |grep "121 "  
*.121 *.* 0 0 262144 0 LISTEN  
192.168.112.2.1009 192.168.111.2.121 263536 0 263536 0 ESTABLISHED  
192.168.112.2.121 192.168.111.2.1008 263536 0 263536 0 ESTABLISHED  
  
# netstat -na |grep rdc  
*.rdc *.* 0 0 262144 0 LISTEN  
ip229.1009 ip230.rdc 263536 0 263536 0 ESTABLISHED  
ip229.rdc ip230.ufsd 263536 0 263536 0 ESTABLISHED
```

The value 263536 shown in this example is the 256 Kbyte buffer size. It must be set identically in the primary and secondary hosts.

▼ To Set and Verify the Buffer Size in a Startup Script

Note – Create this script on the primary and secondary hosts.

1. Create the script file in a text editor using the following values:

```
#!/bin/sh  
ndd -set /dev/tcp tcp_max_buf 16777216  
ndd -set /dev/tcp tcp_cwnd_max 16777216  
  
# increase DEFAULT tcp window size  
ndd -set /dev/tcp tcp_xmit_hiwat 262144  
ndd -set /dev/tcp tcp_recv_hiwat 262144
```

2. Save the file as `/etc/rc2.d/S68ndd` and exit the file.
3. Set the permissions and ownership to the `/etc/rc2.d/S68ndd` file.

```
# /usr/bin/chmod 744 /etc/rc2.d/S68ndd  
# /usr/bin/chown root /etc/rc2.d/S68ndd
```


4. Shut down and restart your server.

```
# /usr/sbin/shutdown -y g0 -i6
```

5. Verify the size as shown in [“To View Buffer Sizes for a Socket”](#) on page 38.

Remote Mirror’s Use of TCP/IP ports

The Remote Mirror software on both the primary and secondary nodes listens on a well-known port specified in `/etc/services`, port 121. Remote Mirror write traffic flows from primary to secondary site over a socket with an arbitrarily assigned address on the primary site and the well-known address on the secondary site. The health monitoring heartbeat travels over a different connection, with an arbitrarily assigned address on the secondary and the well-known address on the primary. The Remote Mirror protocol utilizes SUN RPCs over these connections.

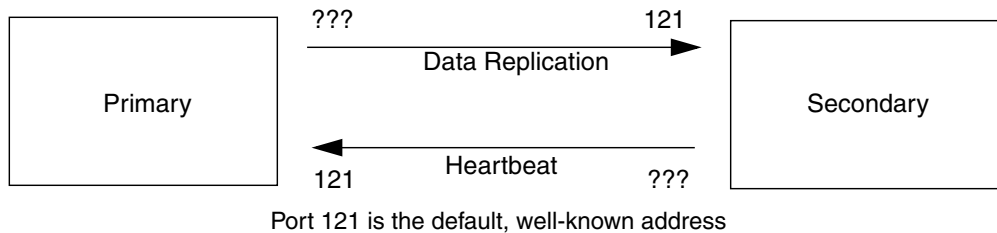


FIGURE 3-1 Remote Mirror’s Use of TCP Port Addresses

Default TCP Listening Port

Port 121 is the default TCP port for use by the Remote Mirror `sndrd` daemon. To change the port number, edit the `/etc/services` file using a text editor.

If you change the port number, you must change it on all Remote Mirror hosts within this configuration set (that is, primary and secondary hosts, all hosts in one-to-many, many-to-one, and multihop configurations). In addition, you must shutdown and restart all hosts affected, so that the port number change can take effect.

Using Remote Mirror With a Firewall

Because RPCs require an acknowledgment, the *firewall* must be opened to allow the well-known port address to be in either the source or destination fields of the packet. If the option is available, be sure to configure the firewall to allow RPC traffic as well.

In the case of write replication traffic, packets destined for the secondary will have the well-known port number in the destination field, acknowledgments of these RPCs will contain the well-known address in source field.

For health monitoring, the heartbeat will originate from the secondary with the well-known address in the destination field, and the acknowledgment will contain this address in the source field.

Remote Mirror Software With Point-in-Time Copy Software

To help ensure the highest level of data integrity and system performance on both sites during normal operations, the Sun StorageTek Availability Suite 4.0 Point-in-Time Copy software is recommended for use in conjunction with Remote Mirror software.

A point-in-time copy can be replicated to a physically remote location, providing a consistent copy of the volume as part of an overall disaster recovery plan. This is commonly referred to as batch replication, and the process and advantages of this practice are described in the best practice guide: *Sun StorageTek Availability Suite Software—Improving Data Replication over a Highly Latent Link*.

The point-in-time copy of a Remote Mirror secondary volume can be established prior to starting synchronization of a secondary volume from the primary site (the site the primary volume is hosted from). Protection against double failure is provided by enabling the Point-in-Time Copy software to create a point-in-time copy of the replicated data at the secondary site before beginning resynchronization. If a subsequent failure occurs during resynchronization, the point-in-time copy can be used as a fallback position, and resynchronization can be resumed when the subsequent failure issues have been resolved. Once the secondary site is fully synchronized with the primary site, the Point-in-Time Copy software volume set can be disabled, or put to other uses, such as remote backup, remote data analysis, or other functions required at the secondary site.

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

To allow this replication to occur, the Point-in-Time Copy software can be configured to offer the Remote Mirror software the changed bitmap. If the Remote Mirror software is in logging mode, it accepts the bitmap and performs an OR comparison of the Point-in-Time Copy software bitmap with its own bitmap for that volume, adding 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, will fail the enable, copy, or update operation. Once Remote Mirror logging has been re-enabled, the Point-in-Time Copy software operation can be reissued.

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

Remote Replication Configurations

The Remote Mirror software enables you to create one-to-many, many-to-one, and multihop volume sets.

- One-to-many replication enables you to replicate data from one primary volume to many secondary volumes residing on one or more hosts. One primary and each secondary site volume is a single volume set. For example, with one primary and three secondary host volumes, you need to configure three volume sets: primary A and secondary B1, primary A and secondary B2, and primary A and secondary B3.
- Many-to-one replication enables you to replicate volumes across more than two hosts through more than one network connection. The software supports the replication of volumes located on many different hosts to volumes on a single host. The terminology differs from the one-to-many configuration terminology, where the one and the many referred to are volumes.
- Multihop replication indicates that the secondary host volume of one volume set acts as the primary host volume of another volume set. In the case of one primary host volume A and one secondary host volume B, the secondary host volume B appears as primary host volume A1 to the secondary host volume B1.

Any combination of the above configurations is also supported by the Remote Mirror software.

Glossary

- asynchronous queue** A local area of disk or memory used to store writes that are to be replicated to a remote site. After the writes have been put into the queue, the write is acknowledged to the application, and the writes are forwarded to the remote site at a later time, as the network capabilities permit.
- asynchronous replication** A synchronous replication confirms to the originating host that the primary I/O transaction is complete before updating the remote image. That is, completion of the I/O transaction is acknowledged to the host when the local write operation is finished and the remote write operation has been queued. Deferring the secondary copy removes the long distance propagation delays from the I/O response time.
- auto synchronization** With the auto synchronization option enabled on the primary host, the synchronization daemon (`autosyncd`) attempts to resynchronize volume sets if the system reboots or a link failure occurs.
- blocking** (asynchronous queue) In blocking mode, if the asynchronous queue fills, all future writes are delayed until the queue drains enough to allow for a write to occur. Blocking mode, which is the default Asynchronous running option, ensures write ordering of the packets to the secondary site. If the asynchronous queue fills with the blocking option set, response time to the application may be impacted.
- configuration location** Location where the Sun StorageTek Availability Suite software stores configuration information about all enabled volumes used by the software.
- consistency group** A consistency group is a group of remote volumes that share a single asynchronous queue to maintain write ordering.
- `dsstat` A tool from the Sun StorageTek Availability Suite tool set that can be used to display kernel statistics from the Remote Mirror and Point-in-Time snapshot products.

firewall	A computer that acts as an interface between two networks and regulates traffic between those networks for the purpose of protecting the internal network from electronic attacks originating from the external network.
forward resynchronization	See Update synchronization.
full synchronization	Full synchronization performs a complete volume-to-volume copy, which is the most time-consuming of the synchronization operations. In most cases, a secondary volume is synchronized from its source primary volume. However, restoration of a failed primary disk might require reverse synchronization, using the surviving Remote Mirror as the source.
hwm	See High water mark
high water mark	The high water mark is the largest amount of the asynchronous queue that has been used.
lazy clear	The practice of clearing the bit in core, but not writing the bitmap block back to disk until either another bit is set, or the in-core copy is reclaimed. This is safe because the change will simply be retransmitted after a system failure.
logging	Mode where a bitmap tracks writes to a disk, rather than a running log of each I/O event. This method tracks disk updates that have not been remotely copied while the remote service is interrupted or impaired. The blocks that no longer match their remote sets are identified for each source volume. The software uses this log to re-establish a Remote Mirror through an optimized update synchronization rather than a complete volume-to-volume copy.
non-blocking	(asynchronous queue) In non-blocking mode, if the asynchronous queue fills, the Remote Mirror software goes into scoreboarding mode and the contents of the queue are discarded. Non-blocking mode, does not ensure write ordering of the packets to the secondary site, but it assures that response time to the application will not be impacted if the asynchronous queue fills.
primary or local: host or volume	The system or volume on which the host application is principally dependent. For example, this is where the production database is being accessed. This data is to be replicated to the secondary by the software.
replication	Once a volume set has been initially synchronized, the software ensures that the primary and secondary volumes contain the same data on an ongoing basis. Replication is driven by user-layer application write operations; replication is an ongoing process.
reverse synchronization	An operation used during recovery rehearsals. Logging keeps track of test updates applied to the secondary system during the rehearsal. When the primary is restored, the test updates are overwritten with the blocks from the primary image, restoring matching remote sets.

secondary or remote: host or volume	The remote counterpart of the primary, where data copies are written to and read from. Remote copies are transmitted without host intervention between peer servers. A server might act as primary storage for some volumes and secondary (remote) storage for others.
synchronization	The process of establishing an identical copy of a source disk onto a target disk as a precondition to the software mirroring.
synchronous replication	Synchronous replication is limited to short distances (tens of kilometers) because of the detrimental effect of propagation delay on I/O response times.
TCP buffer	The TCP buffer size is the number of bytes that the transfer control protocol allows to be transferred before it waits for an acknowledgment.
update synchronization	Update synchronization copies only those disk blocks identified by logging, reducing the time to restore remotely mirrored sets.
volume set file	A text file containing information about specific volume sets. This text file is not the same as the configuration location, which contains information about all configured volume sets used by the Remote Mirror and Point-in-Time Copy software.

Index

Symbols

`/etc/hosts`, 7
`/etc/nsswitch.conf` file
 editing, 12
`/etc/services` file
 editing, 11
`/usr/kernel/drv/rdc.conf`, 13

A

asynchronous queue
 configuring, 28
 configuring flusher threads, 34
 setting size of, 33
asynchronous replication, 25
Availability Suite software
 installing, 3
 uninstalling, 4
 upgrading, 2

B

bitmap
 requirements, 18
bitmap volumes
 size requirements, 18
 suggested location, 18

C

configuration
 file (optional), 20
 files, 7
 IPv6 addresses, 8
 post-installation, 5

configuration information
 backing up, 22
configuration steps, 6
consistency groups, 26

D

`dscfgadm` utility, 15

F

files
 `/etc/hosts`, 7
 `/usr/kernel/drv/rdc.conf`, 13

I

installing Availability Suite software, 3
Internet Protocol version 6 (IPv6), 8

N

network tuning, 36
 TCP buffer size, 36

P

post-installation
 configuration, 5

R

replication
 asynchronous, 25
 remote, 27
 remote configurations, 41
 synchronous, 24

S

- software settings
 - modifying, 13
- Sun StorEdge
 - configuration, 5
 - post-installation, 5
- synchronous replication, 24

T

- TCP / IP ports, 39

U

- uninstalling Availability Suite software, 4
- upgrading Availability Suite software, 2

V

- volume set file
 - using, 20