



Usage Guide for Sun StorEdge™ Availability Suite 3.2 Remote Mirror Software with Oracle® Databases

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

Part No. 817-7380-10
December 2003

Send comments about this document to: docfeedback@sun.com

Copyright 2003 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 embodied in the product 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, AnswerBook2, docs.sun.com, Sun StorEdge, Solstice Disk Suite, Sun ATM, Sun Enterprise, Sun Fire, 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.

Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in the Sun Microsystems, Inc. license agreements and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct. 1998), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable.

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 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et sans la limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains énumérés à <http://www.sun.com/patents> et un ou les brevets plus supplémentaires ou les applications de brevet en attente dans les Etats-Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui 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.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

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

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com, Sun StorEdge, Solstice Disk Suite, Sun ATM, Sun Enterprise, Sun Fire, et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-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 Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation 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 pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISÉE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.



Contents

1. Introduction to the Remote Mirror Software 1

Why Should You Use Data Replication? 2

Oracle and the Sun StorEdge Availability Suite Remote Mirror Software:
A Solution 2

Overview 3

Features 4

Disk-Based Asynchronous Queues 5

6

One-to-Many, Many-to-One, and Multihop Volume Sets 7

Order-Dependent Writes and Volume Set Grouping 8

Mutual Backup 9

Architecture 10

Replication Modes 12

Synchronous Replication 12

Asynchronous Replication 13

Synchronization Modes 14

Full Synchronization 15

Update Resynchronization 16

Reverse Full Synchronization 18

Reverse Update Synchronization 19

Logging	20
Using the Remote Mirror Replication with Oracle	22
Replicating Online Logs	22
Full Database Replication	23
2. Configuring The Remote Mirror and Oracle Software	25
Installation Requirements	26
Configuring the Remote Mirror Software	27
Determining Application Replication Requirements	27
Setting Up the TCP/IP Network Link	29
Allocating Volumes	30
▼ To Create and Set Up Volumes	31
About Replicating File Systems	32
Creating a Remote Mirror Volume Set File	32
Bitmap Volume Configuration	34
Enabling and Replicating the Remote Mirror Volumes	35
▼ To Enable and Replicate Volumes	35
Installing and Configuring Oracle	36
▼ To Install Oracle	36
Configuring the Oracle Database	37
Distributing Data for Optimal Performance	38
Online Log File Replication	38
Full Database Replication	38
Sample Oracle Configuration Files	38
Site-A	39
Site-B	41
Implementing the Remote Mirror Software for an Existing Database	42
3. Failover Scenarios	43

Determine Procedures Before You Begin	43
Identifying and Resolving Failures	44
Link Failures	45
▼ To Manually Place the Remote Mirror Software into Logging Mode	45
▼ To Resynchronize Site-B from Site-A	46
Failing Over to Site-B	46
▼ To Shut Down the Databases	47
▼ To Fail Over to Site-B	47
Switching Back to Site-A as the Production Database By Copying Files	49
▼ To Shut Down the Site-A Database	49
▼ Resynchronize Files from Site-B	50
▼ To Reverse Synchronize Site-A (Full Database Replication)	51
Switching Back to Site-A as the Production Database By Restoring a Backup Copy	53
▼ To Prepare Site-A and Site-B	53
▼ To Return the Sites to Their Original Roles	55
Swapping Host Roles (Database Copy)	57
Volume Set File Used in This Example	57
▼ To Shut Down the Site-A Database	58
▼ To Shut Down Site-B and Enable the remote mirror Software	58
▼ To Perform a Full Database Replication and Start the Database	60
▼ To Replicate the Online Log Files Only	60
▼ To Start the Database at Site-A	61
Swapping Host Roles (Restoring Using a Backup Copy and Archive Logs)	62
Configuration File Used in This Example	62
▼ To Shut Down the Site-A Database	63
▼ To Shut Down Site-B and Enable the Remote Mirror Software	64
▼ To Perform a Full Database Replication and Start the Database	65
▼ To Start the Database at Site-A	66

Implementing One-To-Many Replication 67

Implementing Multihop Replication 68

4. Conclusions 69

References 69

Introduction to the Remote Mirror Software

Note – This software was previously known as the Sun StorEdge™ Network Data Replicator (Sun SNDR) software.

This chapter describes the following:

- [“Why Should You Use Data Replication?” on page 2](#)
- [“Overview” on page 3](#)
- [“Architecture” on page 10](#)
- [“Replication Modes” on page 12](#)
- [“Synchronization Modes” on page 14](#)
- [“Using the Remote Mirror Replication with Oracle” on page 22](#)

Why Should You Use Data Replication?

In today's world of Internet-accelerated business practices, companies must meet the demand for 24x7 online communications and operations to remain successful. The business costs of application outages are increasing. Outages can result from major power, server, and communications disruptions, and natural disasters. The potential consequences of application outages are revenue losses, missed opportunities, poor customer relations, fines, and even business failure.

Businesses are searching for ways to reduce the downtime during outages; long outages could be devastating. Business also need to reduce the amount of manual intervention required, reduce downtime, and provide more up-to-date data in case of an outage. While many hardware and software tools provide high reliability on an individual basis, a degree of unreliability is introduced when they are integrated into a system.

Remote mirroring has traditionally been employed in disaster recovery and business continuance strategies to provide redundant storage of critical information across physically separate sites. An investment in multiple data copies can be leveraged into new information-sharing opportunities across distributed enterprises. The result can be operational efficiency for physically disparate computing environments and strategic advantage due to increased information access.

Oracle and the Sun StorEdge Availability Suite Remote Mirror Software: A Solution

The Oracle® database software is a proven database application for enterprises with large volumes of data. The increasing percentage of Oracle installations on UNIX® platforms has indirectly triggered a dependency on the UNIX operating environment and data availability. Understanding this critical issue, Sun Microsystems Inc. has developed the Sun StorEdge Availability Suite 3.2 Remote Mirror software for remote replication and mirroring of data.

The remote mirror software can help businesses to:

- Replicate data to another site effortlessly
- Reduce maintenance costs
- Minimize downtime and thus increase productivity
- Take point-in-time snapshot copies of data from the secondary site, using the Sun StorEdge Availability Suite Point-in-time Copy software
- Provide easier failover, fallback, and reverse-role scenarios

Overview

The remote mirror software is a remote replication facility for the Sun Solaris™ operating environment. It is intended for use as part of a disaster recovery and business continuance plan to provide redundant storage of critical information across physically separate sites.

The remote mirror software enables you to replicate disk volumes between physically separate primary and secondary sites in real time. As part of a disaster recovery and business continuance plan, the remote mirror software enables you to keep up-to-date copies of critical data at remote sites. You can also rehearse your data recovery strategy to failover data to remote sites. Later, you can write any data changes that occurred to the original disk that “failed”. To transport data, the remote mirror software uses any Sun network adapter that supports TCP/IP.

The remote mirror software volume set consists of a primary volume residing on a local site and a secondary volume residing on a remote site. The volume set also includes a bitmap volume on each site to track write operations and differences between the volumes.

The remote mirror software is active while your applications are accessing the data volumes and continually replicates the data to remote sites. The software operates at the volume level on storage devices attached to one or more hosts.

You can update the data on the secondary volume by issuing a command to *resynchronize* the volumes. You can also restore data from the secondary volume to the primary volume by issuing a command to *reverse resynchronize* the volumes.

Features

TABLE 1-1 describes the remote mirror software features.

TABLE 1-1 Remote Mirror Software Features

This Feature . . .	Enables You To . . .
Active logging	Continue logging operations whenever the remote mirror software is disabled or interrupted.
Disaster rehearsal rollback	Simulate a disaster without committing data changes to volumes.
Groups of remote mirror volume sets	<p>Control the remote mirror software volume sets simultaneously by grouping them. This feature is essential in installations requiring you to maintain consistent contents of a group of volumes.</p> <p>A group is a collection of remote mirror software sets that have the same group name, primary and secondary interfaces, and mirroring mode. Mixed groups (those where mirroring modes are asynchronous for one set and synchronous for another set) are not allowed.</p> <p>The remote mirror software maintains write ordering for volumes in a group to ensure that the data on the secondary volumes is a consistent copy of the corresponding primary volumes.</p>
Disk-based asynchronous queues	Data can be queued on disk as well as in memory. Memory-based queues are the default. See “Disk-Based Asynchronous Queues” on page 5 .
Multihop sets	Replicate data from one primary volume to a secondary volume; the secondary volume then replicates the data again to another secondary volume, and so on, in a daisy-chain fashion. See “One-to-Many, Many-to-One, and Multihop Volume Sets” on page 7 .
Mutual backup	Concurrently transmit to and receive remote dual copies from its remote counterpart. Also known as a <i>bilateral relationship</i> .
One-to-many sets	Replicate data from one primary volume to many secondary volumes residing on one or more sites. When you perform a forward resynchronization, you can synchronize one volume set or all volume sets. Issue a separate command for each set. You can also update the primary volume using a specific secondary volume. See “One-to-Many, Many-to-One, and Multihop Volume Sets” on page 7 .
Optimized resynchronization	Resynchronize volumes following disk, link, system, and storage platform outages; you only replicate those blocks that were modified
RAID support	Use RAID volumes as part of your remote mirror software strategy. Volumes can be any RAID level.

TABLE 1-1 Remote Mirror Software Features (*Continued*)

This Feature. . .	Enables You To. . .
Security and Internet Protocols	Use Version 3.2 securely and efficiently in the Sun Solaris 8 and 9 operating environments, which support Internet Security Protocol (IPsec). The Solaris 8 and 9 operating environments also support Internet Protocol Version 6 (IPv6). (The Solaris 7 operating environment does not support IPv6). Version 3.2 does not require the use of an <code>.rhosts</code> file. You place the host machines to be used in the <code>/etc/hosts</code> file of each host running the remote mirror software in your configuration.
Stop and start control of remote replication on a logical volume basis	Have fine control over replication at the logical volume level. You can include volumes containing critical data by specifying these volumes as part of a remote mirror software volume set. You can also exclude volume with noncritical data from remote mirror software options.
Point-in-time copy software integration	Make point-in-time volume copies of your data for additional point-in-time coordination and recovery.
Synchronous and asynchronous volume replication between physically separate sites	Plan for disaster recovery and business continuance using physically separate primary and secondary sites. The remote mirror software design is link-neutral, meaning that it can use any Sun network adapter that supports TCP/IP. Continue remote data access despite local disk failure (depending on how the remote mirror software is configured).

Disk-Based Asynchronous Queues

In the 3.2 version of the remote mirror software, data can be queued on disk as well as in memory. Memory-based queues are the default.

Disk-based queues enable:

- Larger local queues in asynchronous replication mode
- Larger bursts of I/O activity without affecting application response time

If a disk-based queue fills up, the remote mirror software goes to nonblocking, or scoreboarding, mode.

One-to-Many, Many-to-One, and Multihop Volume Sets

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 (it is still the secondary volume of the first 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.

Order-Dependent Writes and Volume Set Grouping

Write ordering is maintained for *groups* of asynchronously replicating volume sets. (The general definition of write ordering is that write operations directed to the target volume occur in the same order as write operations to the source.) The group of target volumes is a copy of the group of source volumes.

This feature can be valuable in those cases where you can avoid application requirements that limit operations. For example, a database application might limit partition sizes to no greater than 2 Gbytes. In this case, you might group volume sets to create a virtual large “volume” that preserves write operations. Otherwise, you might risk having inconsistent data by trying to update volume sets individually instead of as a group.

When an application has multiple logical volumes assigned, application data integrity can be maintained by one of the following:

- Specifying that all remote mirror software volumes associated with that application are in `sync` mode
- Using the Sun StorEdge Availability Suite 3.2 point-in-time copy software to make periodic recoverable point-in-time copies

If you use point-in-time copy software, the remote point-in-time snapshot is taken while the application is in the recoverable state. For example, most database applications allow for a hot backup. If a remote point-in-time copy were made of the entire replicated database while the primary was in hot backup mode, then a consistent remote database is available by using the point-in-time copy and the log files taken while the database was in hot backup mode.

Note – You must place the remote mirror volume set 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 volume set is not in logging mode, the point-in-time copy operation fails and the remote mirror software reports that the operation is denied.

Mutual Backup

Typically, volumes are replicated from local Site-A to remote Site-B. However, as applications are geographically distributed, a storage system at remote Site-B can be a remote volume backup to local Site-A and also be a direct storage resource for applications on Host-B. Under these circumstances, you can replicate Host-B volumes to Site-A.

This reciprocal backup arrangement supported by the remote mirror software is known as *mutual backup* or *dual backup*.

With mutual backup, the remote mirror software volumes considered primary by Site-B are administered from the Site-B session. Site-B replicated-volume devices are considered secondary volumes at Site-A.

In this case, you need to configure two *unique* volume sets for each site. For example,

- Volume set 1 contains primary Site-A volumes replicated to secondary Site-B
- Volume set 2 contains primary Site-B volumes replicated to secondary Site-A

See [FIGURE 1-1](#).

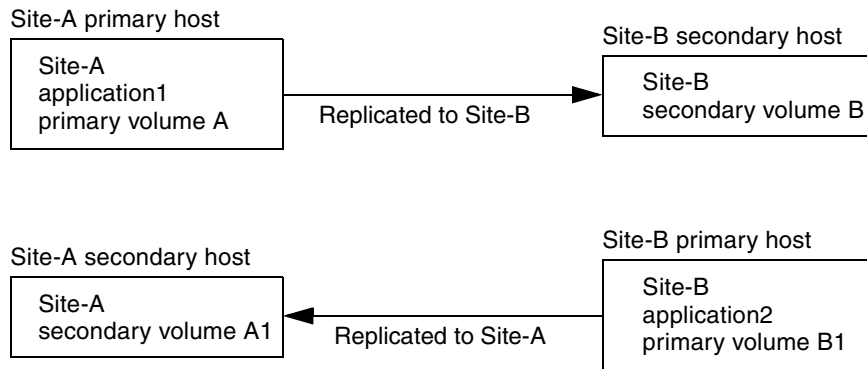


FIGURE 1-1 Mutual Backup

Architecture

The remote mirror software operates on logical volumes that can be linear, striped, or RAID devices. You can create the logical volumes by using the Sun Solstice DiskSuite™ or VERITAS Volume Manager software.

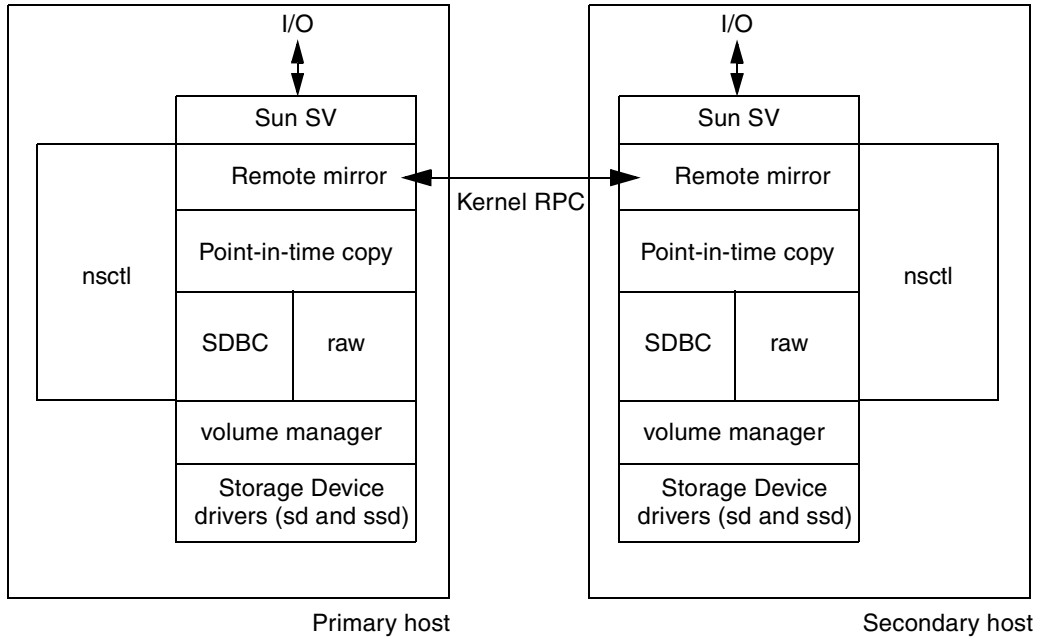
The core remote mirror software code is a kernel module that interfaces to the network storage control module (NSCTL) data service framework. The remote mirror software is configurable on any device that is accessible through the NSCTL data service framework. The `sndradm` CLI is the external user interface used to manage the remote mirror software.

FIGURE 1-2 shows the relationship between the Storage Volume (SV) driver, the remote mirror software, and the rest of the NSCTL data service framework.

The I/O commands targeted to remote mirror volumes are intercepted by the SV driver and routed through the Sun StorEdge I/O stack before being passed on to the storage device driver or the volume manager. The I/O commands originating in user space are intercepted at the top of the Sun StorEdge I/O stack. The SV driver routes them through the stack and feeds them back to the storage device driver or volume manager at the bottom of the stack. Data also flows in the opposite direction, from the storage back to user space.

Because the remote mirror software is located at the top of the stack before the point-in-time copy software, the remote mirror volume sets must be placed in logging mode (`sndradm -l`) before performing point-in-time enable, copy, update or reset operations on any volume that is also part of a remote mirror volume set.

If the volume set is not in logging mode, the point-in-time copy operation fails and the remote mirror software reports that the operation is denied.



1. I/O commands and data enter and exit the remote mirror software through the Sun StorEdge Storage Volume (SV) driver software.
2. Mediated by `nsctl`, the data flows through remote mirror software (and optionally the point-in-time copy software) and the Storage Device Block Cache (SDBC) drivers to its destination on the storage array or in user space.

FIGURE 1-2 The Remote Mirror Software Architecture

Replication Modes

The remote mirror software supports two modes of data replication:

- Synchronous replication
- Asynchronous replication

The replication mode is a user-selectable parameter for each remote mirror volume set. (Use the `sndradm enable` command and select the volume set's `sync` or `async` parameter. Use the `sndradm -R m` command to change the replication mode thereafter.) The volumes can be updated synchronously in real time or asynchronously using a store-and-forward technique.

Typically, a primary volume is first explicitly copied to a designated secondary volume to establish matching contents. As applications write to the primary volume, the software replicates changes to the secondary volume, keeping the two volumes consistent.

In the event of planned or unplanned outages, the software maintains per-device bitmap volumes that are marked to indicate changed blocks with a granularity of 32 Kbytes per segment. This technique optimizes resynchronization by allowing the software to resynchronize only the blocks that have changed since the last synchronization.

Synchronous Replication

In synchronous mode, a write operation is not confirmed as complete until the remote volume has been updated. Synchronous mirroring forces the remote mirror software to wait until an acknowledgement of the receipt of the data is received from the secondary by the primary before returning to the application. The application is not acknowledged until the write at the secondary site is complete.

The advantages of synchronous replication are that it is more reliable and can help reduce the risk of data loss. One disadvantage might be an increase in write response time, especially for large data sets or long-distance replication (where write operations can incur additional latency because of the time required to transfer data and return acknowledgements).

Asynchronous Replication

In asynchronous mode, a write operation is confirmed as complete before the remote volume has been updated. Asynchronous replication allows the remote mirror software to return to the host as soon as the write has been completed on the primary volume and been placed on a per-volume queue for the secondary site. Subsequently, the secondary site receives the queued requests in the order that they were queued. Once the I/O has been completed at the secondary site and the bitmap has been updated to reflect the volume state, notification is sent to the primary.

The advantage of asynchronous replication is that it provides faster response time to the application as long as the asynchronous queue is not full. Once the queue is full, the subsequent data update of that volume waits to enter the queue and this asynchronous replication might essentially become synchronous operation.

A disadvantage is that data loss might occur at the secondary site after primary site or network failures.

The user should size and tune the asynchronous queue size to get optimal operation. The default queue size is 16384 data blocks (approximately 8 Mbytes). To size the queue, you can

- Compute the queue drain rate using `dsstat -m sndr` command
- Monitor the queue using the `kstat sndr` command.
- Discover the I/O pattern

Also, put the remote mirror volumes into logging mode during batch writes like `import`, `SQL*Load`, and so on, as these operations might fill the queue faster. The *Sun StorEdge Availability Suite 3.2 Software Administration and Operations Guide* describes how to adjust the queue, including how to use the `dsstat` and `kstat` commands.

Synchronization Modes

The remote mirror software synchronizes data in five modes:

- Full synchronization
- Update resynchronization
- Reverse full synchronization
- Reverse update synchronization
- Logging

Full Synchronization

Full synchronization starts a full copy operation from the primary volume to the secondary volume. It also enables replication concurrently from the primary volume to the secondary volume; any new writes to the primary volume are also replicated to the secondary volume. After the operation is complete, the remote mirror software maintains the normal replicating mode for the volume: either synchronous or asynchronous replication.

Note – The volumes can be made identical using other methods, not just a full synchronization. When network latencies justify it, you can perform the initial synchronization of a volume set by backing up a source or primary volume on magnetic tape on one site, then restoring the volume from the tape on the other site. During the intervening period (that is, the period between when the backup is completed and the restore is started), place the source or primary volume in logging mode.

Make sure that the backup copy is a physical copy (for example, by using the `dd(1M)` command) and not a logical copy (for example, one made using the `tar(1M)` or `cpio(1M)` commands). The copies must have identical blocks, not just identical files.

[FIGURE 1-3](#) shows the full forward synchronization process.

1. The remote mirror software on the primary system (`host1`) requests disk blocks from the active primary volume. The data might already be resident in the primary system data cache or might require a local disk access.
2. The software transmits the disk blocks, with destaging instructions, over the connection to a cache region on the secondary system.
3. The software on the secondary system updates its remote volume and acknowledges the update to the primary system.

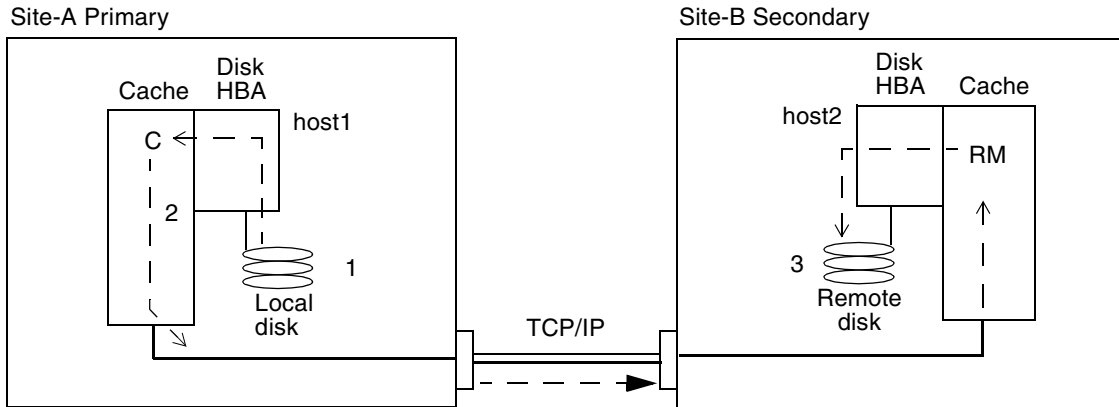


FIGURE 1-3 Full Synchronization (Volume-to-Volume Copy)

Update Resynchronization

During update resynchronization mirroring, the remote mirror software initiates replication of only the changed primary site volume data to the secondary site, based on the bitmap. Only the blocks marked dirty in the bitmap are copied to the target volume.

After the update is complete, the software maintains the normal replicating mode. The software can also be placed in logging mode. See [“Logging” on page 20](#).

The software resynchronizes the secondary volume from the primary volume. It updates the secondary volume according to the changes based on logs maintained while replication was stopped. It also enables concurrent replication between the primary and secondary volumes; any new write operations to the primary volumes are also replicated to the secondary volumes.

If a remote copy interruption lasts numerous hours and the updates are widespread, logging and update resynchronization provide diminishing returns. As time passes, the proportion of bits set to true in the bitmap volumes of a volume set might reach 100 percent. The overhead of logging, coupled with update resynchronization, must then be balanced against that of a full synchronization without intervening periods of logging.

Logging and update resynchronization serve as a built-in safety net should one of your replication processes be interrupted. The software monitors the network connections between the primary and secondary hosts. Link failures and remote system failures are detected by the transport interface and are passed to the remote mirror software.

FIGURE 1-4 shows an update resynchronization from the primary system to its secondary system, when the secondary volumes are stale from the interruption.

1. The remote mirror software on `host1` examines a bitmap from the primary and secondary hosts for the remote mirror software-managed volumes affected by the interruption.
2. The software on `host1` requests the blocks that were updated during the interruption from the up-to-date volume. The data might already reside in the `host1` data cache or on the local disk.
3. The software on `host1` transmits the update blocks `3R` to `host2` remote mirror software.
4. The software on `host2` refreshes its stale replicated image with the updated blocks and acknowledges the action to `host1`.
5. The software revises the bitmap to track the remote update.

All steps repeat until the remote replicated image is updated.

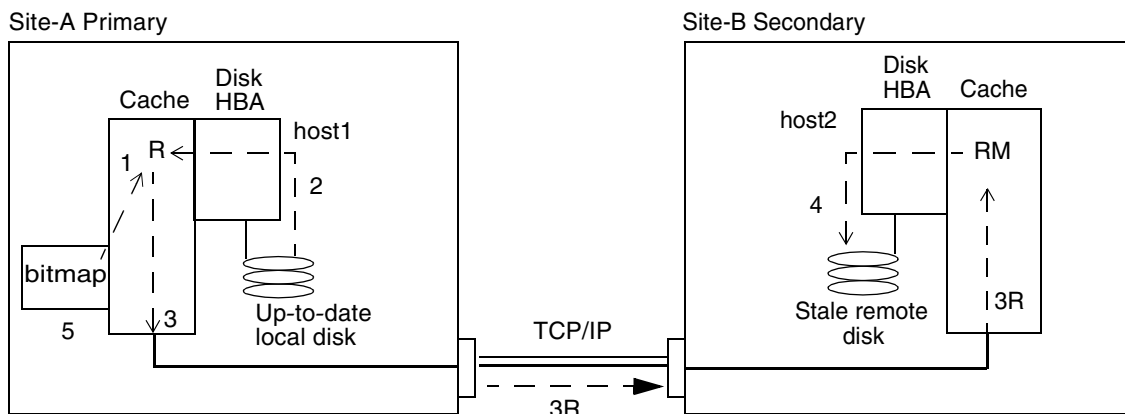


FIGURE 1-4 Update Resynchronization of a Secondary Volume Set

Reverse Full Synchronization

During reverse full synchronization, the remote mirror software replicates the volume data from the secondary site to the primary site. After you issue the `sndradm -m -r` command, the software starts a full reverse copy operation from the secondary volume to the primary volume. It also enables concurrent replication from the primary volume to the secondary volume; any new writes to the primary volume are also replicated to the secondary volume. This remote mirror feature can help reduce application downtime.

You can use the primary volume during reverse synchronization. The primary volume shows a consistent volume image of the latest data as soon as the reverse synchronization starts. If your application had been writing to the secondary volume as part of a failure or disaster rehearsal, you can move the application back to the primary volume when the reverse synchronization starts.

FIGURE 1-5 shows the full reverse synchronization process.

1. The data might already be resident in `host2` data cache, or it might require a secondary disk access. If so, the remote mirror software on `host1` requests blocks from the up-to-date secondary volume on `host2`.
2. The software on `host2` transmits the cache blocks 2R over the link to a software region on `host1` with destaging instructions.
3. The software on `host1` updates its disk.

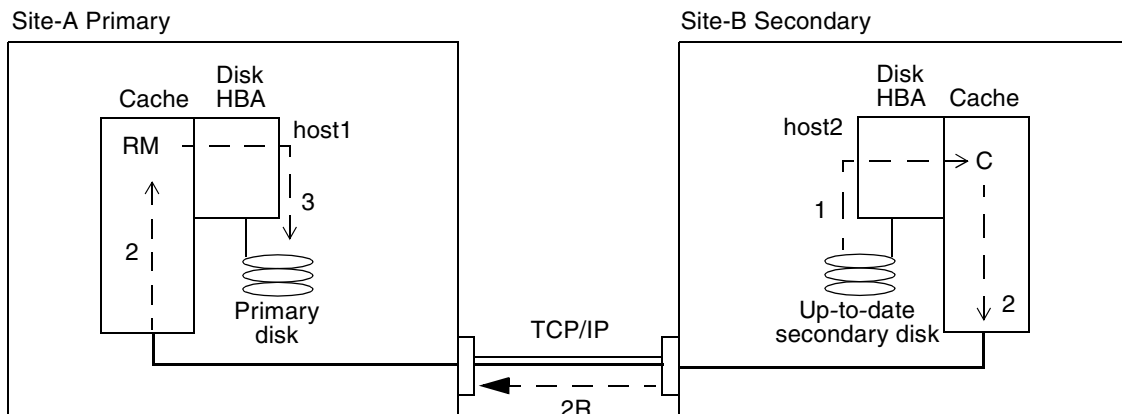


FIGURE 1-5 Reverse Full Synchronization

Reverse Update Synchronization

During reverse update synchronization mirroring, the remote mirror software compares the bitmaps between the primary and secondary sites and replicates only the changed blocks from the secondary site to the primary site.

It resynchronizes the primary volume from the secondary volume. It updates the primary volume according to the changes based on logs maintained while replication was stopped. It also enables concurrent replication between the primary volume and secondary volumes; any new write operations to the primary are also replicated to the secondary volumes. This remote mirror feature can help reduce application downtime.

You can use the primary volume during reverse update synchronization. The primary volume shows a consistent volume image of the latest data as soon as the reverse update synchronization starts. If your application had been writing to the secondary volume as part of a failure or disaster rehearsal, you can move the application back to the primary volume when the reverse update synchronization starts.

FIGURE 1-6 shows a reverse update resynchronization from the secondary system to its primary system.

1. The remote mirror software on `host1` retrieves the secondary bitmap 1R from `host2` for one of the remote mirror software-managed volumes affected by the interruption.
2. The software on `host1` requests the blocks updated during the interruption from the up-to-date secondary volume of `host2`. The data might already be resident in `host2`'s data cache, or it might require secondary disk access.
3. The software on `host2` transmits the updated blocks 3R to `host1` remote mirror software region of cache using the intersite link.
4. The software on `host1` refreshes its stale image with the updated blocks.
5. The software on `host1` revises the bitmap to track the remote update.

All steps repeat until the primary volume is up-to-date.

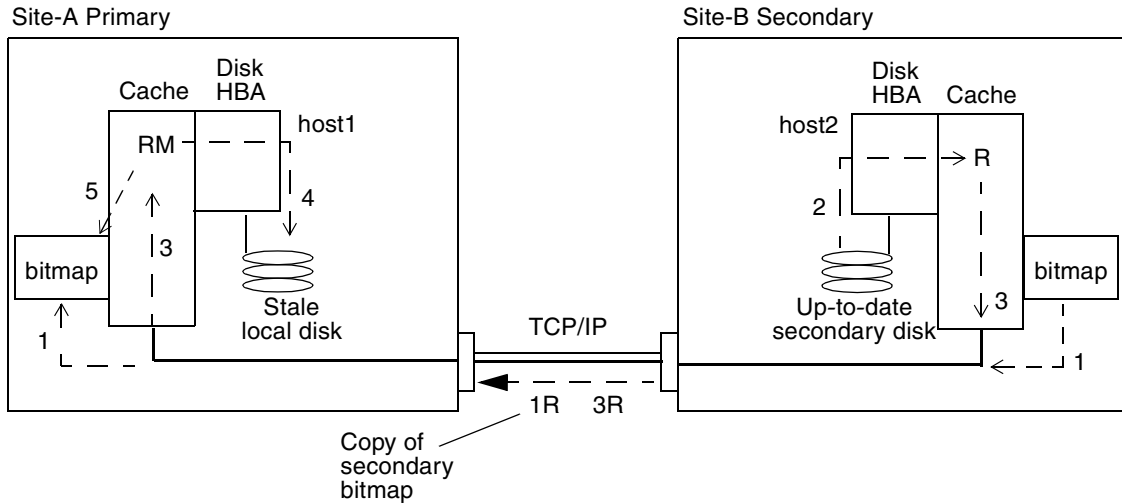


FIGURE 1-6 Reverse Update Synchronization

Logging

Note – With synchronous and asynchronous replication, the remote mirror software automatically switches to logging mode if a break in the network occurs or if the primary site is down.

During logging, the remote mirror software only updates the bitmaps at the primary site; no replication occurs. At a later time, the bitmaps at the primary and secondary sites are compared and the changed blocks in the primary site volume are mirrored by update resynchronization to the secondary site.

If all volume sets in an I/O group are replicating (that is, the secondary volumes contain a valid point-in-time copy of the corresponding primary volumes), when one volume set enters logging mode, all other sets in the I/O group will enter logging mode automatically. This scheme ensures that the secondary volumes will contain a valid copy of the data.

You can use logging to save on telecommunications or connection costs. The risk, however, is the costs incurred by increased data loss if the primary site is lost. If you lose the primary site, you do not have the data at the secondary site that was written to the primary site during logging.

You can also perform logging on the secondary site before a failover. You can then update the primary site using a reverse synchronization or reverse update synchronization command.

Note – To resume remote mirror software operations after using the `sndradm -l` logging command, use the `sndradm -m` command to perform a full resynchronization or the `sndradm -u` command to perform an update resynchronization. Note also that, when issued from the secondary host, the `sndradm -l` command does not work on the secondary volume for any volume that is currently synchronizing.

Using the Remote Mirror Replication with Oracle

You can use the Oracle standby database for disaster recovery purposes. The standby database is maintained by applying the archive redo logs generated at the production database to the standby database. With Oracle, the database can automatically ship archive logs from primary to standby and can apply the archive logs. However, the standby database cannot guarantee no loss of committed transactions; the current online log file information is lost in the standby database.

Replicating Online Logs

Using online logs replication, the remote mirror software mirrors the log files from the primary site to the secondary site. If a disaster occurs at the primary site or a network failure occurs, the mirrored online redo logs can be used to recover the standby database to the last committed transaction. In *sync* mode, potential data loss can be reduced in the event of a disaster. In *async* mode, data loss might occur in the event of a disaster. The advantages of online logs replication are reduced cost and improved performance.

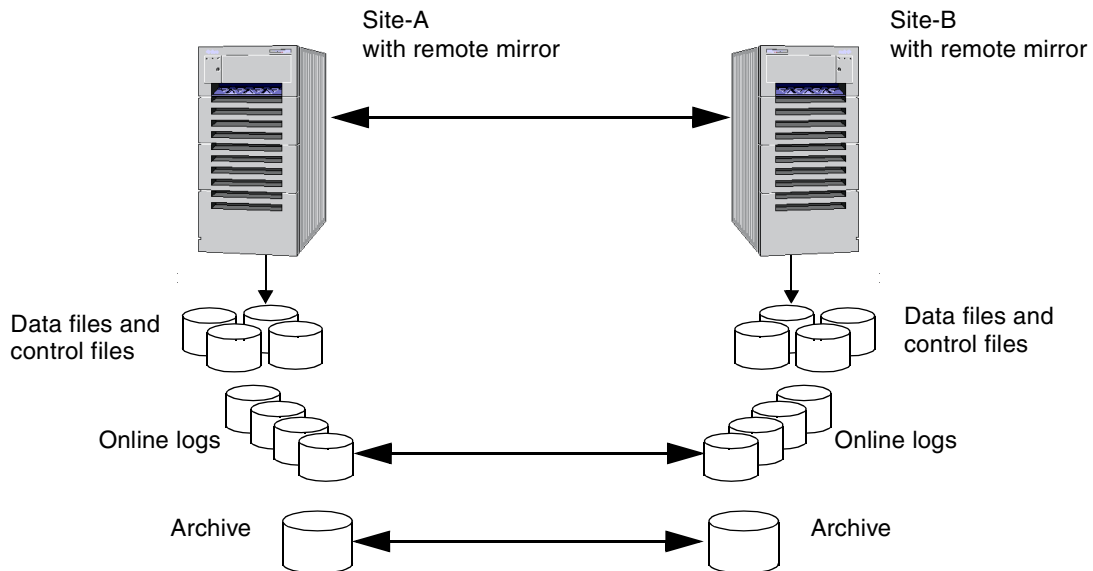


FIGURE 1-7 Replicating Online logs for the Oracle Standby Database

Full Database Replication

Using the remote mirror software, you can define multiple volume sets, each containing information such as index, data, rollback and log files. If a disaster occurs at the primary site, you can accomplish the failover operation to the secondary site quickly to minimize the crash recovery time.

Consider performance issues carefully when replicating the entire database in `sync` mode. You might use a combination of `sync` and `async` modes to replicate, depending on the activity and the importance of the files. *Make sure that if you place volume sets in an I/O group, all volume sets have the same replication mode; do not mix modes in an I/O group.*

In `sync` mode, potential data loss can be reduced in the event of a disaster. In `async` mode, data loss might occur in the event of a disaster.

The advantage of full database replication is that the failover operation is relatively straightforward. The disadvantages are performance degradation, costlier operation, and that any primary data corruption is propagated to the secondary volumes.

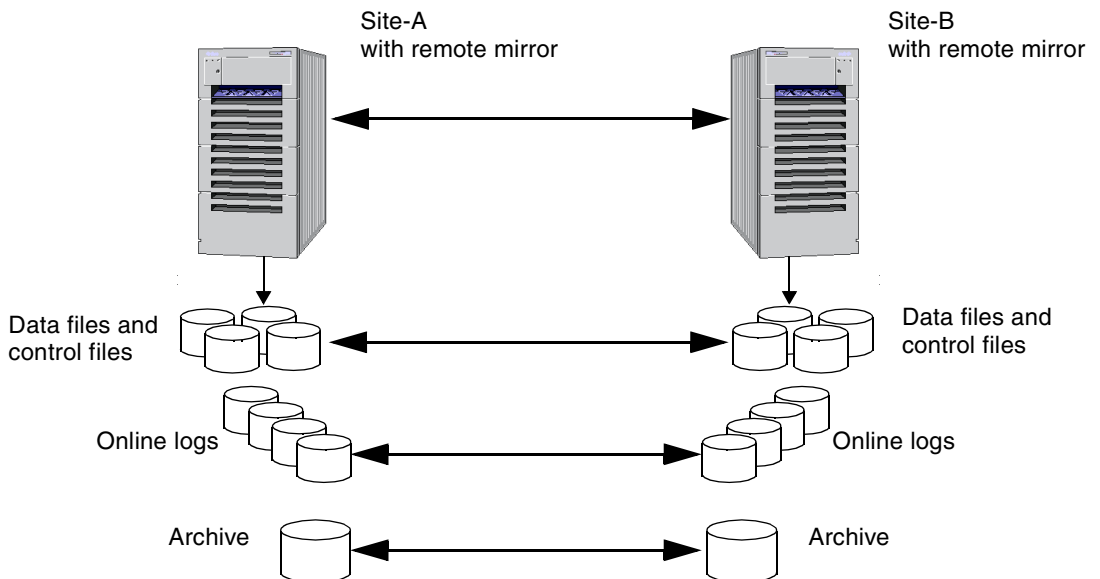


FIGURE 1-8 Replicating the Full Oracle Database

Configuring The Remote Mirror and Oracle Software

This chapter describes installation requirements and configuration procedures for the remote mirror and Oracle software.

Topics described include:

- [“Installation Requirements” on page 26](#)
- [“Configuring the Remote Mirror Software” on page 27](#)
- [“Installing and Configuring Oracle” on page 36](#)

Installation Requirements

Operating Environment	Operating environment: <ul style="list-style-type: none">• Solaris 8• Solaris 9u3 and later
Oracle and Sun StorEdge Software	Oracle database software, versions 8 and 8i Sun StorEdge Availability Suite 3.2 remote mirror software
Supporting Software	TCP/IP network transport software such as SunATM™ or Gigabit Ethernet transports
Optional Recommended Software	Sun StorEdge Availability Suite 3.2 point-in-time copy software; install this package for additional point-in-time copy capability
Host Types	The remote mirror software is supported on server hosts using the Solaris operating environment and any network interface card supported by Sun. Hosts include but are not limited to: <ul style="list-style-type: none">• Sun Enterprise™ 220R, 250, 420R, and 450 servers• Sun Enterprise 3500, 4500, 5500, 6500, and 10000 servers• Sun Fire™ 3800, 4800, 4810, and 6800 servers• Sun Fire 15K server• Sun Ultra™ 60 and 80 workstations• Sun Blade™ 100 and 1000 workstations• Sun Netra™ t 1400/1405 and 1120/1125 servers
Hardware	A CD-ROM drive connected to the host server where the Sun StorEdge software is to be installed.
Supported Attached Storage	The remote mirror software is storage-hardware independent. Allocate approximately 15 Mbytes for the Sun StorEdge Availability Suite software installation. <ul style="list-style-type: none">• The remote mirror software requires approximately 1.7 Mbytes.• The Sun StorEdge configuration location requires 5.5 Mbytes• Supporting packages require approximately 5.4 Mbytes. Oracle disk space and memory requirements: <ul style="list-style-type: none">• Oracle binaries require approximately 1 Gbytes• Also required is a minimum 128 Mbytes of memory (512 Mbytes or greater recommended)• Volume manager required

Configuring the Remote Mirror Software

The remote mirror configuration steps are as follows:

1. Determine the application replication requirements.
2. Set up the replicating TCP/IP network link.
3. Allocate storage for the remote mirror volumes for the primary and secondary hosts.
4. Allocate the remote mirror bitmap volumes.
5. Install the remote mirror software.
6. Create an optional remote mirror configuration file.
7. Perform the initial volume replication using the remote mirror software.

Determining Application Replication Requirements

Before you start planning the actual remote mirror configuration, you must fully understand your application requirements for consistency and the application recovery methods. First, you must decide which data requires replicating and the mode of replication to be used. A number of methods are available:

- Replicate volumes using a mix of synchronous and asynchronous modes according to the volumes' level of importance
This mode choice provides the most practical solution and is the method detailed in this document.
- Replicate all volumes in synchronous mode
This mode choice simplifies procedures because data at the primary and secondary sites is identical, apart from any data lost in transit during a failure. It is reliable and can help reduce the risk of data loss; one disadvantage might be an increase in response time, especially for large data sets or long distance replication.

- Replicate all volumes in asynchronous mode

This mode choice might allow the possibility of lost transactions during system failures and might result in the database being inconsistent. The advantages of asynchronous replication are that it provides fast response and has the least impact on the response time of the primary application. The disadvantage is that there is a possibility of data loss at the secondary site after primary site or network failures.

The database files can be grouped based on the requirements for their availability and the importance for the application recoverability process as shown in [TABLE 2-1](#).

TABLE 2-1 Oracle File Entities and Suggested Replication Mode

Oracle File Entities	Access Method	Replication mode
System tablespace and RollBack tablespace	Raw volumes	Synchronous
Control files and archive logs	File system	Synchronous
Log files	Raw volumes	Synchronous
Data files	Raw volumes	Asynchronous
Index files	Raw volumes	Asynchronous
Temp files	Raw volumes	Asynchronous
Application binaries	File system	Not replicated

If the primary host server becomes unavailable, the most important files required to bring the secondary site online in a consistent and up-to-date state are the System and Rollback tablespace, the control files, and the archive Logs. If consistent data, index and temp volumes are available, then it is possible to place the secondary database into a recent and consistent state. With the addition of the log files, it is possible to completely update the database.

The suggested order and priority when resynchronizing the primary host and secondary host sites are:

1. System and RollBack tablespace
2. Control files and archive logs
3. Data files
4. Index files
5. Temp files

Keep a copy of the control files on the file system used to store the archive logs, so that a single volume resynchronization can update these files, instead of resynchronizing an entire volume solely for the purpose of obtaining access to the control files. Alternatively, you can create an SQL script to generate the control file and copy it to the secondary site.

Note that any future changes made to the database data files usage also require changes to the remote mirror implementation and the configuration of the secondary host. The application binaries are not being replicated in this scenario but could be replicated, if required.

Setting Up the TCP/IP Network Link

You can use any TCP/IP network transport software such as SunATM™ or Gigabit Ethernet transports. This section assumes that one of two types of ATM are installed: SunATM-155 or SunATM-622. These interfaces run at theoretical speeds of 155 Mbytes per second and 622 Mbytes per second, respectively, with realistic throughput of 93 Mbytes per second and 400 Mbytes per second. If the two servers in the remote mirror configuration are not on a local network, then the limiting factor for throughput is the bandwidth of the ATM link being used. See [TABLE 2-2](#).

When a volume set is in enabled and replicating mode, the remote mirror software uses only a small fraction of the available bandwidth, depending on the amount of data being modified on the volume set and the number of sets enabled. Running in sync mode potentially requires a wide network pipe (so that the network will not be a bottleneck), as the I/O operation does not complete until it has been written to both the local and the remote servers. In async mode, a smaller pipe is acceptable, as the data is not transmitted in real time.

During full synchronization operations, however, the maximum available bandwidth is used, since the entire contents of the volumes are transmitted over the link. A trade-off must be made between the cost of installing and operating multiple lines and the time taken to replicate a set of volumes.

TABLE 2-2 Speeds for Various Communications Protocols

Link	Speed
OC-1	51.85 Mbytes per second
OC-3	155 Mbytes per second
OC-12	622Mbytes per second
OC-48	2.4 Gbytes per second
OC-192	9.6 Gbytes per second

TABLE 2-2 Speeds for Various Communications Protocols (*Continued*)

Link	Speed
T-3	45 Mbytes per second theoretical; 27.2 Mbytes per second realistic
E-3	34.5 Mbytes per second theoretical; 21 Mbytes per second realistic
T-1	1.54 Mbytes per second theoretical; 0.92 Mbytes per second realistic

Allocating Volumes

Note – When selecting a volume to be used in the remote mirror software volume set, ensure that volume does not contain disk label private areas (for example, slice 2 on a Solaris operating environment-formatted volume). The disk label region is contained in the first sectors of cylinder 0 of a disk. The safest method is to ensure that cylinder 0 is not part of any logical volume that is replicated (except for volumes under VERITAS Volume Manager control, where cylinder 0 can be part of a logical volume that is replicated).

- Volume size

The size of the secondary site volume must be equal to or greater than the corresponding primary site volume. If you initiate a remote mirror volume set where the secondary volume is larger than the primary, a warning message appears but the transaction is allowed to continue. If you initiate a volume set where the secondary volume is smaller than the primary volume, the remote mirror command `sndradm` fails with an error.

- Bitmap volume

Each volume to be replicated requires a bitmap volume in the volume set. The bitmap size can be calculated using the following formula:

- 1 Kbytes + 4 Kbytes per Gbyte of device storage space

For example, a 2-Gbyte data device requires a bitmap size of 9 Kbytes. You can create bitmaps that are larger than the calculated size. Also, round up any partial Gbyte counts to the next Gbyte. See also [“Bitmap Volume Configuration” on page 34](#).

To determine the volumes to be replicated using the remote mirror software, you must balance remote accessibility and recoverability against capacity usage and I/O response time. Typically, for the Oracle database, you can include the following volumes in the remote replication configuration:

- **For data that is constantly changing:**
 - Online logs
 - Control files
 - Data files (typically multi-volume)
 - Index files (typically multi-volume)
 - Rollback segments (typically multi-volume)
 - Archived logs
- **For data that seldom changes or can be reconstructed:**
 - Static tablespace data files
 - Temporary tablespace data files

Do not include the following as remote mirror volumes:

- Spool files
- Paging volumes
- Oracle binary
- Other UNIX binary volumes

▼ To Create and Set Up Volumes

1. **Create disk groups using volume manager software such as Sun Solstice DiskSuite.**

2. **Create the following volumes.**

- A volume for the Oracle binary (not to be replicated)
- One or more volumes for the database data files
- A volume for online logs
- A volume for the archive log files, if utilizing full database replication

3. **Install the remote mirror software.**

See the *Sun StorEdge Availability Suite 3.2 Software Installation Guide* for details.

4. **Create directories under the root (/) directory as mount points for volumes.**

5. If a file system is used, then create the entries in the `/etc/vfstab` file.

At Site-B, do not mount the remote mirror volumes automatically. See [About Replicating File Systems](#).

About Replicating File Systems

The remote mirror software is not a file or file system replicator. It is a volume replicator. When you replicate a volume on the primary Site-A to a volume on the secondary Site-B, Site-B receives an exact bit-for-bit copy. Make sure that any Site-A file systems you want to replicate are mounted.

When replicating, the primary host file system is mounted. Do not mount the file system on Site-B until you are ready to fail over or write to that site's volume. Changes appear on a replicated file system volume only after mounting the volume.

Creating a Remote Mirror 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, Site-A and Site-B 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 might be more convenient to put this information in a file when you have multiple volume sets.

One advantage when using one or more volume set files is that you can operate on specific volume sets and exclude 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 configuration file specified using the `-f config-file` option are:

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

An example configuration file entry is as follows:

```
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
```

See [TABLE 2-3](#). See the `rdc.cf` man page for more information about this configuration file format.

TABLE 2-3 Volume Set File Format Fields

Field	Meaning	Description
<i>phost</i>	Primary host	Server on which the primary volume resides. In this guide, Site-A.
<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. In this guide, Site-B
<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	<i>sync</i> is the mode where the I/O operation is confirmed as complete only when the remote volume has been updated. <i>async</i> is the mode where the primary host I/O operation is confirmed as complete before updating the remote volume.
<i>g io-groupname</i>	I/O group name	An I/O group name can be specified using the <i>g</i> character. In this example, it is <i>oragroup</i> .

Bitmap Volume Configuration

Note – After editing the `/usr/kernel/drv/rdc.conf` file, reboot your server.

The remote mirror version 3.2 software does not support bitmap files. The remote mirror software uses regular raw devices to store bitmaps, used to maintain the remote mirror bitmap. For example, you can specify a bitmap volume as `/dev/rdisk/c6t0d0s4`.

These raw devices should be stored on a disk separate from the disk that contains the data. Configure RAID (such as mirrored partitions) for these bitmap devices and ensure that the mirrored members are not stored on the same disk as the data.

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`. For example:

```
# 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;
```

If your server is not configured in a clustered environment, you can also choose the bitmap mode setting of 1 to improve error or disaster recovery.

Enabling and Replicating the Remote Mirror Volumes

For `/usr/opt/SUNWesm/sbin/sndradm` command details and information, see the *Sun StorEdge Availability Suite 3.2 Remote Mirror Administration and Operations Guide*.

▼ To Enable and Replicate Volumes

1. Use one of the following commands to enable the volumes defined in your configuration file.

- `sndradm -e -n -f config-file`
- `sndradm -E -n -f config-file`

where:

`-e` indicates that a full resynchronization between volumes is required.

`-E` clears the bitmap logging to indicate that the specified volumes are fully synchronized and enables logging. Typically, you use this command if the data at Site-B already matches Site-A and no replication is required.

`-n` indicates interactive mode is off (that is, you do not have to respond to program prompts to enable the volume sets).

`-f config-file` specifies the configuration file.

2. Perform the initial replication from Site-A to Site-B using the `sndradm -m` command.

The replication command options are:

Replication type	Command
Copy the entire contents of the primary volume to the secondary volume; also known as <i>full synchronization</i> .	<code>sndradm -m</code>
Copy the entire contents of the secondary volume to the primary volume; also known as <i>full reverse synchronization</i> .	<code>sndradm -m -r</code>
Update only the changed data of the primary volume to the secondary volume; also known as <i>update synchronization</i> or <i>resynchronization</i> .	<code>sndradm -u</code>
Update only the changed data of the secondary volume to the primary volume; also known as <i>reverse synchronization</i> or <i>reverse update</i> .	<code>sndradm -u -r</code>

3. **For monitoring remote mirror operations, use the**
`/usr/opt/SUNWscm/sbin/scmadm -SM` **or** `/usr/opt/SUNWscm/sbin/dsstat`
commands.

Installing and Configuring Oracle

The following sections describe general guidelines for installing Oracle. For detailed instructions, see the Oracle installation documentation such as the *Installation Guide for Oracle 8.1.7 for the Sun Solaris Operating Environment*. See [“Installation Requirements” on page 26](#) for volume manager, disk space, and memory requirements.

Note – The remote mirror software operates with any version of Oracle. In this guide, Oracle 8.1.7 is assumed. If you are using any version prior to 8.1, ignore the “recover managed” syntax for managed recovery of the standby database.

▼ To Install Oracle

1. **Login as the root user.**
2. **Create the `orainstall` group and attach `groupid` to it.**
3. **Create user `oracle` with the main group `orainstall` and attach `userid` to it.**
4. **Perform the following steps using `admintool(1M)`.**
 - a. **Create the directory `/opt/lbin` and include it in your path.**
 - b. **Edit the Oracle user's environment variables for the appropriate login shell scripts for `ORACLE_HOME`, `ORACLE_SID`, `ORACLE_TERM`, `LD_LIBRARY_PATH`, and `PATH` (to include `ORACLE_HOME`).**

c. Edit the `/etc/system` file and add the following:

```
set shmsys:shminfo_shmmax=4294967295
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=100
set shmsys:shminfo_shmseg=10
set semsys:seminfo_semmns=200
set semsys:seminfo_semmni=100
set semsys:seminfo_semmsl=100
set semsys:seminfo_semopm=100
set semsys:seminfo_semvmx=32767
```

5. Reboot the system.

6. Login as the `oracle` user.

7. Insert the Oracle Universal Installer CD in the CD-ROM drive and follow the instructions.

Note – Do not install the Oracle binaries on the replicating volume. The software must be installed separately on each system.

8. Creating the starter database is optional. Do so, if required.

9. Create the database.

See the Oracle documentation for creating the database.

Configuring the Oracle Database

Refer to the Oracle documentation for information about installing the Oracle server and the steps to configure the Oracle database. See the white paper *Guidelines for Using Remote Mirroring Storage Systems for Oracle System* for information on planning for the remote mirroring of an Oracle database.

Use similar setups for both Site-A and Site-B hosts.

Distributing Data for Optimal Performance

You must decide whether you want to replicate the full database or only the online redo logs. From a performance perspective, it is recommended that you perform online redo logs replication and use the standby database at the secondary site. However, you can implement full database replication.

Online Log File Replication

This choice assumes that Site-B hosts the standby database. If the online redo files are multiplexed, use `sync` mode on all the members and groups. If `async` mode is used, data loss might occur during disaster failover operations.

Full Database Replication

Use separate volumes for redo logs, system data files, data, index, RBS, and archive logs. Even though using `sync` mode for all the volumes helps make failover operations easier, it has an impact on performance at the Site-A host.

Sample Oracle Configuration Files

You can use these files for both online log replication and full database replication. All files are located in the `$ORACLE_HOME/network/admin` directory.

Any text in *italics* is a variable that you must supply:

- *Site-X-hostname* – name of the host where *X* is Site-A or Site-B
- *sid* – system identifier
- *domain-name* – domain for the Site-A host

Note – For ease of use, Oracle provides a graphical program to help add information to these files instead of editing them in a text editor. The program is named Net8 Assistant and can be started from `$ORACLE_HOME/bin/netasst`.

Site-A

File listener.ora

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC) (KEY = sid))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP) (HOST = Site-A-hostname) (PORT = 1521))
      )
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = sid.domain name)
      (ORACLE_HOME = /ora1/oracle/product/817)
      (SID_NAME = sid)
    )
  )
```

File tnsnames.ora

```
sid.domain name =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = Site-A-hostname) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = sid.domain name)
    )
  )

sid.domain name =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = Site-B-hostname) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = sid)
    )
  )
```

File sqlnet.ora

```
NAMES.DEFAULT_DOMAIN = domain name
automatic_ipc=off
```

Site-B

File listener.ora

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP) (HOST = Site-B-hostname) (PORT = 1521))
      )
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = sid.domain-name)
      (ORACLE_HOME = /ora1/oracle/product/817)
      (SID_NAME = sid)
    )
  )
)
```

File tnsnames.ora

```
sid.domain-name =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = Site-B-hostname)
      (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = sid.domain-name)
    )
  )

sid.domain-name =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = Site-A-hostname) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SID_NAME = sid.domain-name)
    )
  )
)
```

File `sqlnet.ora`

```
NAMES.DEFAULT_DOMAIN = domain-name  
automatic_ipc=off
```

Implementing the Remote Mirror Software for an Existing Database

Use the following steps to implement the remote mirror software for an existing database.

1. Make sure no data changes occur to the database at Site-A; quiesce the application if needed.
2. Determine the requirements for the remote mirror software.
3. Allocate storage on Site-B for the remote mirror volumes.
4. Create the remote mirror configuration files.
5. Perform the initial remote mirror replication of volumes.
6. Define procedures for normal and failure conditions.
7. Rehearse the procedures.

Failover Scenarios

Determine Procedures Before You Begin

Determine when to fail over to Site-B. If you switch to Site-B, you have to perform steps both in switching to it and later switching back to Site-A. Document the time for each step and its related factors so that when a failure occurs, you can determine the best approach for the current scenario.

In summary, only switch to Site-B if it provides greater availability.

1. Calculate the normal time to to:
 - reboot the server
 - fail over to Site-B
 - switch back to Site-A
2. Failing over to Site-B requires you to:
 - mount the necessary file systems
 - start, recover, and roll forward the application
 - switch users from Site-A to Site-B
3. Switching back to Site-A requires you to:
 - perform a reverse synchronization of the remote mirror software volumes multiple times
 - shut down the application on Site-B
 - unmount any file systems on Site-B
4. Reverse synchronization of the remote mirror volumes to Site-A and waiting for completion requires you to:

- mount the application volumes on Site-A
- synchronize the remote mirror volume set
- Start the application on Site-A

Identifying and Resolving Failures

The remote mirror software can automatically detect network link failures and remote system failures; the software can log incomplete and new write operations. After restoring the network or remote site, you can re-enable and resynchronize the volumes.

If the remote mirror volume sets are running in `async` mode, you should not see any performance degradation occurring on Site-A when the link fails. However, in `sync` mode, any current I/O operations must first time-out. During this time, any application on Site-A that has pending I/O operations might appear to freeze. After a time-out has occurred, the remote mirror software enters logging mode and Site-A continues to operate normally.

While the remote mirror software is designed to reduce the negative impact of data unavailability at one site by keeping secondary copies, it cannot prevent data unavailability. For example, a Site-A host failure while the remote mirror software is suspended and logging can result in Site-B data that is current only to the time when logging began on Site-A.

The remote mirror software does not prevent data corruption from software bugs, operator errors, or hardware failures. In fact, the remote mirror software might faithfully replicate some of these anomalies to Site-B volumes as part of the data replication process.

Link Failures

Note – For information about the `/usr/opt/SUNWesm/sbin/sndradm` command, see the *Sun StorEdge Availability Suite 3.2 Remote Mirror Software Administration and Operations Guide*.

If the network link fails and replication is occurring in synchronous mode or asynchronous mode, the remote mirror software at Site-A enters logging mode after a time-out. The bitmap bits are set according to any changes in the data area. The remote mirror software then acknowledges to the database or application that the write operation is complete.

■ **Synchronous mode replication**

The database might freeze momentarily during the time-out (that is, the time required for the remote mirror software to switch from replicating mode to logging mode).

■ **Asynchronous mode replication**

The database or application gets the write operation acknowledgment immediately, as soon as the bitmap is set.

The following steps show how to resynchronize the sites after a link failure.

▼ To Manually Place the Remote Mirror Software into Logging Mode

Perform this step from Site-A if you want simulate a link failure, which will place both sites into logging mode.

● **To start logging:**

```
# /usr/opt/SUNWesm/sbin/sndradm -ln -f config-file
```

See [“Creating a Remote Mirror Volume Set File”](#) on page 32.

▼ To Resynchronize Site-B from Site-A

Note – Perform these steps from Site-A.

1. **Make sure that Site-A and Site-B show they are in logging mode using the `dsstat` command.**

After the network link is restored:

2. **Perform fast synchronization to update the volumes at Site-B.**

```
# sndradm -un -f config-file
```

After this step, Site-A replicates the changes to Site-B. You can monitor this process using the `scmadm -SM` command. Once the replication is finished, Site-A and Site-B resume the normal replicating mode.

Failing Over to Site-B

Note – This recovery scenario describes switching the production database to Site-B.

If Site-A is not accessible, business users must connect to an alternate site, Site-B, to continue operations. Use the following procedures a failover scenario to Site-B.

After you perform these steps, Site-B is the failover site. As soon as Site-A is accessible, you must perform a switch back or reverse-role scenario. To switch Site-A back as the production database, see [“Switching Back to Site-A as the Production Database By Copying Files” on page 49](#) or [“Switching Back to Site-A as the Production Database By Restoring a Backup Copy” on page 53](#).

▼ To Shut Down the Databases

1. At Site-A, shut down the production database if it is running by using one of the following Oracle commands.

```
shutdown immediate
or
shutdown abort
or
shutdown transactional
```

2. At Site-B, shut down the standby database if it is running by using one of the following Oracle commands.

```
shutdown immediate
or
shutdown transactional
```

▼ To Fail Over to Site-B

Note – Perform these steps from Site-B.

1. Place the remote mirror volumes into logging mode so that changes can be applied to Site-A later.

```
# sndradm -ln -f config-file
```

When you issue this command from Site-B (the remote host), it does not affect volumes that are currently synchronizing. Wait until all volumes have finished synchronizing before issuing this command.

2. If you are replicating a file system, run the `fsck(1M)` command for the replicated volumes before mounting them.

```
# fsck /dev/vx/rdisk/diskgroup/volume
# mount mount-point
```

3. Full database replication only: Recover the database and open it as a production database using the following Oracle commands. Otherwise, skip to [Step 4](#).

- a. Start the database in `mount` mode.

```
startup pfile=path/initsid.ora mount
```

- b. Recover the database.

```
alter database recover automatic database
```

- c. Open the database as the production database.

```
alter database open noresetlogs
```

4. For Standby Databases Only: Open the Site-B standby database as a production database using the following Oracle commands.

- a. Start the database in `nomount` state.

```
startup pfile=path/initsid.ora nomount
```

- b. Mount the standby database.

```
alter database mount standby database
```

- c. Recover the database.

```
alter database recover automatic standby database
```

- d. Open the database as the production database.

```
alter database activate standby database
```

5. Shut down the database and make a cold backup copy.

```
shutdown immediate  
perform the cold back up
```

6. Open the database.

```
startup pfile=path/initsid.ora mount
alter database read write
```

Switching Back to Site-A as the Production Database By Copying Files

Note – This recovery scenario describes copying files from Site-B to Site-A and restore the sites to their original roles.

After the failover is activated, Site-B is used for the production database. To fail over to Site-B, execute the failover steps described in [“Failing Over to Site-B” on page 46](#).

After the network or Site-A failure problem is fixed, these steps describe how to switch back to Site-A as the production database and return Site-B as the standby or replicated database.

▼ To Shut Down the Site-A Database

1. At Site-A, shut down the database if it is running by using one of the following Oracle commands.

```
shutdown immediate
or
shutdown transactional
```

2. If you have been replicating only the log files, remove the old archive log files using the `rm (1M)` command.
3. If you are replicating a file system, run the `fsck(1M)` command for the replicated volumes before mounting them.

```
# fsck /dev/vx/rdisk/diskgroup/volume
# mount mount-point
```

▼ Resynchronize Files from Site-B

1. Copy the control file to Site-A.

```
alter database backup control file to filename
```

2. If a cold backup is needed, shut down the database and take a cold backup of the database. If it is not needed, skip to [Step 3](#).

```
shutdown immediate  
or  
shutdown transactional  
perform the cold back up
```

3. If a cold backup is not needed, place the tablespaces into Hot Backup mode.

```
alter tablespace tablespacename begin backup
```

4. Perform one of the following steps.

- a. If only log files are being replicated, copy the data files to Site-A.
- b. If the full database is being replicated, perform the reverse synchronization (also known as *resilvering*) for *data file volumes only*.

Make sure that your configuration file specified by `-f config-file` contains the correct volumes.

Note – Perform the following command at Site-A.

```
# sndradm -un -f config-file
```

5. If a backup operation was performed in Hot Backup mode, end the backup mode.

```
alter tablespace tablespacename end backup
```


6. Shut down the database.

```
shutdown transactional
or
shutdown immediate
```

7. If a file system is being replicated, unmount all the remote mirror volumes.

```
# umount mount-point
```

Note – If you are replicating only online logs, perform the following steps.

8. At Site-A, create a standby control file and set the archive destination.

```
alter database create standby control file
alter system archive log current
alter system set log_archive_dest_state_2 = enable
alter system set log_archive_min_succeed_dest = 2
```

9. At Site-A, copy the standby control file to Site-B.

▼ To Reverse Synchronize Site-A (Full Database Replication)

1. If you are replicating the full database, perform a reverse update synchronization (also known as *resilvering*) for *archived logs, online logs, and control file volumes*. Make sure that your configuration file specified by `-f config-file` contains the correct volumes.

Note – Perform the following command at Site-A.

```
# sndradm -un -f config-file
```

After this step, Site-A returns to the normal replicating mode.

2. If you are replicating a file system, run the `fsck(1M)` command for the replicated volumes before mounting them.

```
# fsck /dev/vx/rdisk/diskgroup/volume
# mount mount-point
```

3. Start up and mount the database as a production database with the necessary secondary archive log destination (specified in `initsid.ora`).

```
startup pfile=path/initsid.ora mount
alter database recover automatic database
alter database open
```

4. Perform the following steps at Site-B.

- a. Start the database in `nomount` state.

```
startup pfile=path/initsid.ora nomount
```

- b. Mount the standby database and recover the database.

```
alter database mount standby database
recover managed standby database
```

Switching Back to Site-A as the Production Database By Restoring a Backup Copy

Note – This recovery scenario describes restoring a backup database copy from Site-B to Site-A and designating Site-A as the production database host.

After the failover is activated, Site-B is used for the production database. To fail over to Site-B, execute the failover steps described in “[Failing Over to Site-B](#)” on page 46.

After the network or Site-A failure problem is fixed, these steps describe how to switch back to Site-A as the production database and return Site-B as the standby or replicated database by restoring a pre-failure backup Site-A database.

▼ To Prepare Site-A and Site-B

1. At Site-B, create the control file script using the following Oracle commands.

```
alter database backup controlfile to trace
```

Note – Perform the following procedures at Site-A.

2. Remove the old archive logs using the `rm(1M)` command.
3. Restore the data files and control files from the backup performed before the disaster.
4. If only log files are being replicated or if the archive logs are not a remote mirror volume, copy the archived log files from Site-B using the `cp(1M)` command.
5. If you are replicating the full database and the archive logs are in a replicating volume, reverse update synchronization of only the archived log file volume.
 - a. If a file system is part of the replication, unmount the archive logs volumes.

```
# umount mount-point
```

b. Perform the reverse update synchronization (also known as resilvering).

Make sure that your configuration file specified by `-f config-file` contains the correct volumes.

```
# sndradm -u -r -f config-file
```

6. If you are replicating a file system, run the `fsck(1M)` command for the replicated archive log volume before mounting it.

```
# fsck /dev/vx/rdisk/diskgroup/volume  
# mount mount-point
```

7. Start the database and mount the database.

```
startup pfile=path/initsid.ora mount
```

8. Recover the database.

```
alter database recover automatic database
```

Next, apply the log files to bring the database to the current level.

9. Shut down the databases at Site-A and Site-B.

Perform this command from both sites.

```
shutdown transactional  
or  
shutdown immediate
```

▼ To Return the Sites to Their Original Roles

1. If a file system is part of the replication, unmount the online redo logs volumes.

```
# umount mount-point
```

2. Perform the reverse update synchronization (also known as resilvering).

Make sure that your configuration file specified by `-f config-file` contains the correct online logs volumes.

```
# sndradm -un -r -f config-file
```

3. If you are replicating a file system, run the `fsck(1M)` command for the replicated online redo log volume before mounting it.

```
# fsck /dev/vx/rdisk/diskgroup/volume  
# mount mount-point
```

- a. Start the database in `nomount` state.

```
startup pfile=path/initsid.ora no mount
```

4. Create the control file using the script created in [Step 1](#).

```
create controlfile ...
```

5. Mount the database.

```
alter database mount
```

6. Recover the database and open it.

```
alter database recover automatic database  
alter database open
```

7. Create the standby control file and set the archive destination.

```
alter database create standby control file
alter system archive log current
alter system set log_archive_dest_state_2 = enable
alter system set log_archive_min_succeed_dest = 2
```

8. Copy the control file to Site-B.
9. At Site-B, start the database in `nomount` state.

```
startup pfile=path/initsid.ora no mount
```

10. At Site-B, mount the database as the standby database.

```
alter database mount standby database
```

11. At Site-B, recover the standby database

```
recover managed standby database
```

12. At Site-A, open the database.

Swapping Host Roles (Database Copy)

Note – This recovery scenario fails Site-A over to Site-B. Site-B then becomes the primary remote mirror host and Site-A becomes the secondary remote mirror host.

After the failover is activated, Site-B is used for the production database. To fail over to Site-B, execute the failover steps described in [“Failing Over to Site-B” on page 46](#).

After the network or Site-A failure problem is fixed, these steps describe how to reverse the roles of the sites; Site-A will store the standby replicated database and Site-B will store the production database. In this scenario, you must edit the configuration files for both hosts.

Volume Set File Used in This Example

In this section, the example remote mirror volume set file is named `/etc/opt/SUNWrdc/rdc.cf`. Following is the entry in the `rdc.cf` file showing Site-A (atm10) and Site-B (atm20) details. See also [“Creating a Remote Mirror Volume Set File” on page 32](#).

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

Site-A (primary host)	atm10
Site-A host volume	/dev/vx/rdsk/oracle816/oratest
Site-A host bitmap volume	/dev/vx/rdsk/oracle816/oratest_bm
Site-B (secondary host)	atm20
Site-B host volume	/dev/vx/rdsk/oracle816/oratest
Site-B host bitmap volume	/dev/vx/rdsk/oracle816/oratest_bm
Transmission protocol	ip
Replication mode	sync

I/O groupname

An I/O group name can be specified using the `g` character. In this example, it is `oragroup`.

▼ To Shut Down the Site-A Database

1. At Site-A, shut down the production database if it is running by using the following Oracle command.

```
shutdown immediate
or
shutdown transactional
```

2. Disable the remote mirror software and discard the remote mirror bitmap.

```
# sndradm -dn -f /etc/opt/SUNWrdc/rdc.cf
```

3. Edit the `rdc.cf` file to swap the Site-A primary host information and Site-B secondary host information.

For example, in the example entry shown in [“Volume Set File Used in This Example” on page 57](#), change `atm10` to `atm20` and `atm20` to `atm10`.

4. Unmount the replicating volume at the mount point.

If you are replicating a file system and are performing full database replication, unmount all remote mirror volumes. If you are replicating online logs only, unmount those volumes only.

```
# umount mount-point
```

▼ To Shut Down Site-B and Enable the remote mirror Software

1. At Site-B, shut down the database if it is running by using the following Oracle command.

```
shutdown immediate
or
shutdown transactional
```


2. Disable the remote mirror software and discard the remote mirror bitmap.

```
# sndradm -dn -f /etc/opt/SUNWrdc/rdc.cf
```

3. Edit the `rdc.cf` file to swap the Site-A primary host information and Site-B Edit the `rdc.cf` file to swap the Site-A primary host information and Site-B secondary host information.

For example, in the example entry shown in [“Volume Set File Used in This Example” on page 57](#), change `atm10` to `atm20` and `atm20` to `atm10`.

4. Enable the remote mirror software.

```
# sndradm -en -f /etc/opt/SUNWrdc/rdc.cf
```

- If you are performing a full database replication, see [“To Perform a Full Database Replication and Start the Database” on page 60](#).
- If you replicating the online logs only, see [“To Replicate the Online Log Files Only” on page 60](#).

▼ To Perform a Full Database Replication and Start the Database

Note – Perform these steps at Site-B.

1. Perform a full synchronization from Site-B to Site-A.

```
# sndradm -mn -f /etc/opt/SUNWrdc/rdc.cf
```

2. Start the database in open mode using the following Oracle command.

```
startup pfile=path/initsid.ora
```

3. Proceed to [“To Start the Database at Site-A” on page 61.](#)

▼ To Replicate the Online Log Files Only

Note – Perform these steps at Site-B.

1. Create a standby control file, make the archive log current, and enable the secondary destination of the archived logs.

```
alter database create standby control file filename
alter system archive log current
alter system set log_archive_dest_state_2 = enable
alter system set log_archive_min_succeed_dest = 2
```

2. Copy the data files and standby control files to Site-A using the `cp(1M)` copy command.
3. Proceed to [“To Start the Database at Site-A” on page 61.](#)

▼ To Start the Database at Site-A

1. At Site-A, start the database in the `nomount` state.

```
startup pfile=path/initsid.ora nomount
```

2. Open it as a standby database.

```
alter database mount standby database
```

3. Recover the database.

- a. If you use managed recovery, type:

```
recover managed standby database
```

- b. If you use automatic recovery, type the following after the archive logs are shipped:

```
alter database recover automatic standby database
```

Swapping Host Roles (Restoring Using a Backup Copy and Archive Logs)

Note – This recovery scenario fails Site-A over to Site-B. Site-B then becomes the primary remote mirror host and Site-A becomes the secondary remote mirror host. The Site-A database is restored by using a backup copy taken before the failure. The archive logs from Site-B are then copied to Site-A.

After the failover is activated, Site-B is used for the production database. To fail over to Site-B, execute the failover steps described in [“Failing Over to Site-B” on page 46](#).

After the network or Site-A failure problem is fixed, these steps describe how to reverse the roles of the sites by restoring a Site-A backup copy taken before the disaster. After restoring the backup copy, the database is updated by using the archive logs shipped from Site-B. In this scenario, you must edit the configuration files for both hosts.

Configuration File Used in This Example

In this section, the example remote mirror configuration file is named `/etc/opt/SUNWrdc/rdc.cf`. Following is the entry in the `rdc.cf` file showing Site-A (atm10) and Site-B (atm20) details. See also [“Creating a Remote Mirror Volume Set File” on page 32](#).

```
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
```

Site-A (primary host)	atm10
Site-A host volume	/dev/vx/rdisk/oracle816/oratest
Site-A host bitmap volume	/dev/vx/rdisk/oracle816/oratest_bm
Site-B (secondary host)	atm20
Site-B host volume	/dev/vx/rdisk/oracle816/oratest
Site-B host bitmap volume	/dev/vx/rdisk/oracle816/oratest_bm

Transmission protocol	ip
Replication mode	sync
I/O groupname	An I/O group name can be specified using the <code>g</code> character. In this example, it is <code>oragroup</code> .

▼ To Shut Down the Site-A Database

1. At Site-A, shut down the production database if it is running by using the following Oracle command.

```
shutdown immediate
or
shutdown transactional
```

2. Disable the remote mirror software and discard the remote mirror bitmap.

```
# sndradm -dn -f /etc/opt/SUNWrdc/rdc.cf
```

3. Edit the `rdc.cf` file to swap the Site-A primary host information and Site-B secondary host information.

For example, in the example entry shown in [“Configuration File Used in This Example” on page 62](#), change `atm10` to `atm20` and `atm20` to `atm10`.

4. Restore the backup copy taken before the disaster.
5. If you are replicating a file system and are replicating online logs only, unmount those volumes only.

```
# umount mount-point
```

▼ To Shut Down Site-B and Enable the Remote Mirror Software

1. At Site-B, shut down the database if it is running by using the following Oracle command.

```
shutdown immediate
or
shutdown transactional
```

2. Disable the remote mirror software and discard the remote mirror bitmap.

```
# sndradm -dn -f /etc/opt/SUNWrdc/rdc.cf
```

3. Edit the `rdc.cf` file to swap the Site-A primary host information and Site-B Edit the `rdc.cf` file to swap the Site-A primary host information and Site-B secondary host information.

For example, in the example entry shown in [“Configuration File Used in This Example”](#) on page 62, change `atm10` to `atm20` and `atm20` to `atm10`.

4. Enable the remote mirror software.

```
# sndradm -en -f /etc/opt/SUNWrdc/rdc.cf
```

- If you are performing a full database replication, perform a full synchronization for the archive log and online log volumes specified in the `rdc.cf` file. (You might need separate configuration files to specify each volume type.)

Continue to [“To Perform a Full Database Replication and Start the Database”](#) on page 60.

▼ To Perform a Full Database Replication and Start the Database

Note – Perform these steps at Site-B.

1. Perform a full synchronization from Site-B to Site-A.

```
# sndradm -mn -f /etc/opt/SUNWrdc/rdc.cf
```

2. Start the database in open mode using the following Oracle command.

```
startup pfile=path/initsid.ora
```

3. Create a standby control file, make the archive log current, and enable the secondary destination of the archived logs.

```
alter database create standby control file filename
alter system archive log current
alter system set log_archive_dest_state_2 = enable
alter system set log_archive_min_succeed_dest = 2
```

4. Copy the data files and standby control files to Site-A using the `cp(1M)` copy command.

Continue to [“To Start the Database at Site-A”](#) on page 66.

▼ To Start the Database at Site-A

1. At Site-A, start the database in the `nomount` state.

```
startup pfile=path/initsid.ora nomount
```

2. Open it as a standby database.

```
alter database mount standby database
```

3. Recover the database.

- a. If you use managed recovery, type:

```
recover managed standby database
```

- b. If you use automatic recovery, type the following after the archive logs are shipped:

```
alter database recover automatic standby database
```

Implementing One-To-Many Replication

Note – See “[One-to-Many, Many-to-One, and Multihop Volume Sets](#)” on page 7.

The concepts and failover scenarios of one-to-one replication (described in this guide) and one-to-many replication are similar. One-to-many replication introduces the concept that you can perform replication in parallel.

For example, consider the following scenario; it enables you to test and develop live data at two remote sites:

- A production database is located in California (Site-A) and is the remote mirror primary host location. Its configuration requires one volume set definition each for:
 - A test database located in Massachusetts (Site-B), secondary host 1
 - A development database located in Florida (Site-C), secondary host 2

The remote mirror configuration would include two entries, each with a unique Site-A bitmap volume. (You cannot use the same bitmap at Site-A for the Site-B and Site-C hosts because you would risk data corruption when you synchronize all volume sets.)

If you configure the remote mirror software to operate in this one-to-many mode, the database can be replicated to Site-B and Site-C simultaneously. Also, if the network link in Massachusetts has failed, the Site-A to Site-C link is unaffected and remains live and replicating.

One-to-many replication helps provide many creative replication and disaster recovery solutions. For example, depending on distance and the network link type, you can replicated one volume set in synchronous mode and another in asynchronous mode. You could also reverse-update volumes from Site-B to Site-A, then replicate the Site-A volumes to Site-C.

Implementing Multihop Replication

Note – See [“One-to-Many, Many-to-One, and Multihop Volume Sets” on page 7.](#)

The concept difference between one-to-many and multihop replication is that you can replicate volumes in parallel using one-to-many; multihop replication typically occurs serially.

This guide describes how to swap host roles: Site-A becomes the secondary site location for data while Site-B becomes the primary site location. With one-to-many replication, Site-B can be a primary and secondary site location simultaneously (exclusive of Oracle database configuration, of course).

Using the site example in [“Implementing One-To-Many Replication” on page 67](#), you can replicate volumes from primary Site-A to secondary Site-B. Then primary Site-B can replicate to secondary Site-C. This setup requires you to define two volumes sets for Site-B, with each volume set containing a unique bitmap volume, one for Site-A-to-Site-B replication and one for Site-B-to-Site-C replication.

If the Site-A volumes become corrupted or unavailable, later you can replicate the Site-B volumes to Site-A. Depending on your volume needs and your site requirements, you can also replicate Site-C to Site-B to Site-A. When replicating between Site-A and Site-B, place the Site-B-to-Site-C volumes in [logging mode](#).

Conclusions

1. For both log file replication and full database replication, the remote mirror software might not lose any committed data while running in synchronous mode. Log file replication, as used here, had the least impact on performance.
2. Running the remote mirror software in asynchronous mode might further improve performance, since the application is not required to wait for the acknowledgement from the secondary site.
3. Replicating the entire database might impact system performance, especially as the distance between sites increases.

Note – Both replication modes offer different fall-back and failover procedures. The choice of one mode or the other depends on the nature of the business, the transaction volume, and the allowable down time during disaster recovery.

References

1. *Sun Test Results for Sun StorEdge Availability Suite 3.2 Remote Mirror Software with Oracle Databases*
by Sridhar Ranganathan, Sun Microsystems.
2. *Guidelines for Using Remote Mirroring Storage Systems for Oracle Database*
by J. Bill Lee, Oracle
3. *Guidelines for Testing Remote Mirroring Storage Systems for Oracle Database*
by J. Bill Lee, Cynthia Yip, Oracle
4. *Sun StorEdge Network Data Replicator Oracle Database Implementation Guidelines*
by Judy Leach, Philip Crane, Sun Microsystems, Inc.

5. *Oracle Installation Manual for 8.1.7 - Solaris*
6. *Sun StorEdge Availability Suite 3.2 Software Installation Guide*
7. *Sun StorEdge Availability Suite 3.2 Remote Mirror Software Administration and Operations Guide*

Glossary

Asynchronous replication	Asynchronous 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.
Configuration location	Location where the Sun StorEdge Availability Suite software stores configuration information about all enabled volumes used by the software.
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.
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.
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.

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.