

Using EMC SRDF/S and SRDF/A with Oracle8i Database, Oracle9i Database, and Oracle Database 10g

Abstract

This white paper provides a best practices guide to performing disaster restart, remote replication, and failure remediation operations for Oracle databases using the EMC® SRDF®/S and SRDF/A products on UNIX servers.

November 2005

Copyright © 2005 EMC Corporation. All rights reserved.

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

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

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

EMC², EMC, EMC ControlCenter, AlphaStor, ApplicationXtender, Catalog Solution, Celerra, CentraStar, CLARAlert, CLARiiON, ClientPak, Connectrix, Co-StandbyServer, Dantz, Direct Matrix Architecture, DiskXtender, Documentum, EmailXtender, EmailXtract, HighRoad, Legato, Legato NetWorker, Navisphere, OpenScale, PowerPath, RepliStor, ResourcePak, Retrospect, Smarts, SnapShotServer, SnapView /IP, SRDF, Symmetrix, TimeFinder, VisualSAN, VSAM Assist, Xtender, Xtender Solutions, and where information lives are registered trademarks and EMC Developers Program, EMC OnCourse, EMC Proven, EMC Snap, EMC Storage Administrator, Access Logix, ArchiveXtender, Authentic Problems, Automated Resource Manager, AutoStart, AutoSwap, AVALONidm, C-Clip, Celerra Replicator, Centera, CLARevent, Codebook Correlation Technology, Common Information Model, CopyCross, CopyPoint, DatabaseXtender, Direct Matrix, DiskXtender 2000, EDM, E-Lab, EmailXaminer, Enginuity, eRoom, FarPoint, FLARE, FullTime, Graphic Visualization, InfoMover, Invista, MirrorView, NetWin, NetWorker, OnAlert, Powerlink, PowerSnap, Rainfinity, RepliCare, SafeLine, SAN Advisor, SAN Copy, SAN Manager, SDMS, SnapImage, SnapSure, SnapView, StorageScope, SupportMate, SymmAPI, SymmEnabler, Symmetrix DMX, UltraPoint, Viewlets, VisualSRM, and WebXtender are trademarks of EMC Corporation. All other trademarks used herein are the property of their respective owners.

Part Number H1143.1

Table of Contents

Introduction	5
Oracle Storage Compatibility Program (OSCP).....	5
Scope	5
Audience	6
Related Documentation	6
 SRDF Family Overview	 7
Disaster Restart and Disaster Recovery.....	7
 SRDF Family Products	 8
SRDF/S Synchronous Replication	9
Using the SRDF/Consistency Groups (SRDF/CG) Option with SRDF/S Synchronous Replication.....	9
SRDF/S Adaptive Copy Replication	10
When to Use SRDF/S Adaptive Copy Replication	10
SRDF/A Asynchronous Replication	11
SRDF/A and Oracle Databases	11
SRDF/MS Multi-Session Consistency	12
SRDF/Automated Replication (SRDF/AR).....	12
 SRDF Family Operations.....	 13
 Best Practices for Using SRDF/S Synchronous Replication and Oracle RDBMS.....	 15
Disaster Restart Solutions Using SRDF/S Synchronous Replication.....	15
Restart Solution 1: Full Database Protection (No Data Loss).....	15
Restart Solution 2: Using a Remote TimeFinder/Mirror Split.....	15
Backup Solutions Using SRDF/S Synchronous Replication.....	16
Backup Solution 1: Using a Remote TimeFinder/Mirror Split.....	16
Backup Solution 2: Using an SRDF Split for Read-Write Databases	18
Backup Solution 3: Using an SRDF Split for Read-Only Databases	19
Backup Solution 4: Using an SRDF Split for Closed Databases	19
Failure Remediation Scenarios Using SRDF/S Synchronous Replication	20
Failure Remediation Scenario 1: Resuming Protection after a Link Failure	20
Failure Remediation Scenario 2: Failover to Remote Site	20
Failure Remediation Scenario 3: Failback to Local Site	21
 Best Practices for Using SRDF/A Asynchronous Replication and Oracle RDBMS.....	 22
Disaster Restart Solutions Using SRDF/A Asynchronous Replication	22
Restart Solution 1: Full Database Protection	22

Restart Solution 2: Using a Remote TimeFinder/Mirror Split	22
Restart Solution 3: Performing an SRDF Split	23
Backup Solutions Using SRDF/A Asynchronous Replication	24
Backup Solution 1: Using a Remote TimeFinder/Mirror Split	24
Backup Solution 2: Using an SRDF Split for Read-Write Databases	25
Failure Remediation Scenarios Using SRDF/A Asynchronous Replication	27
Failure Remediation Scenario 1: Resuming Protection after a Link Failure	27
Failure Remediation Scenario 2: Failover to Remote Site	27
Failure Remediation Scenario 3: Failback to Local Site	28

Introduction

The demand for database protection and availability increases as databases grow in size, systems consolidate, and the organization infrastructure expands. Continuous access to the database and applications is required, and having a valid and current backup or restartable image is essential. When disaster strikes, the organization is measured by its ability to resume operations quickly, seamlessly, and with the minimum amount of data loss. This timely response minimizes loss of revenue during system downtime and ensures customer satisfaction.

To achieve this response, EMC provides its Symmetrix® Remote Data Facility family of solutions (referred to as “SRDF” in the rest of this document) for remote replications, disaster restart, and disaster recovery. In a matter of minutes from the time of a disaster, the database and applications can be brought up on a remote site and connectivity can be re-established. SRDF can also be used to create a fast and reliable device-level replica in a remote location, such as another site, city, or continent. This replication can be used by reporting applications, perform data updates, offload backup operations to a safe location, import to the data warehouse, and more. For short distances, the remote replication can be synchronous (a campus solution), which provides realtime data mirroring between local and remote sites. When the distance grows, latency and performance considerations make an asynchronous solution more viable. This paper presents best practices for using the SRDF/Synchronous (SRDF/S) and SRDF/Asynchronous (SRDF/A) products with Oracle databases.

Oracle Storage Compatibility Program (OSCP)

This guide is based on tests that were performed in accordance with the Oracle Storage Compatibility Program (OSCP) for Remote Mirroring technologies. The Oracle-supplied OSCP test kit was used to validate database integrity; this test kit is designed to test the compatibility of storage vendor technologies with Oracle databases. Successful completion of these tests is required to validate EMC SRDF/S synchronous replication and SRDF/A asynchronous replication as an Oracle-certified solution.

Scope

This document describes best practices for EMC SRDF/S and SRDF/A with Oracle8i Database, Oracle9i Database, and Oracle Database 10g on UNIX servers and includes the following topics:

- SRDF family functionality included in versions of EMC Solution Enabler up to version 6.0 running on Symmetrix units using Enginuity™ Versions 5x65 to 5x67, 5568, 5669, 5670, and 5671.
- Best practices for disaster recovery, disaster restart, and remote replication solutions with SRDF/S and SRDF/A with Oracle databases, including data replication technologies for backup and recovery.
- Use of remote TimeFinder® family business continuance volumes (BCVs) to further enhance database protection.
- Best practices to renew database availability and protection after failure of RDF links or failures at the local or remote site.

Database integrity and validity are preserved by following the best practices described in this document. Topics not included in this document are interoperability between the SRDF family and Oracle Recovery Manager (RMAN) or features and functionality of other members of the EMC SRDF family of products.

Audience

The primary target audience of this document is database and system administrators, storage administrators, and system architects who are responsible for implementing, maintaining, and protecting robust databases and storage systems. It is assumed that readers are familiar with Oracle database software, basic UNIX system administration, and EMC products, and are interested in achieving higher database availability and protection.

Related Documentation

The following manuals and white papers provide information related to the concepts discussed in this paper:

- *EMC Solutions Enabler Symmetrix SRDF Family CLI Product Guide*
- *EMC Solutions Enabler Symmetrix TimeFinder Family CLI product Guide*
- *EMC Solutions Enabler Symmetrix SRM CLI Product Guide*

SRDF Family Overview

The Symmetrix Remote Data Facility family of solutions (referred to as “SRDF” in this paper) provides Symmetrix-based business continuance and disaster restart solutions. In simplest terms, SRDF is a configuration of Symmetrix units whose purpose is to maintain multiple, realtime copies of logical volume data in more than one location. The Symmetrix units can be in the same room, in different buildings within the same campus, or hundreds of kilometers apart. SRDF provides data mobility and disaster restart spanning multiple host platforms, operating systems, and applications.

The local SRDF device, known as the *source (R1) device*, is configured in a pairing relationship with a remote *target (R2) device*, forming an *SRDF pair*. While the R2 device is mirrored with the R1 device, the R2 device is Write Disabled or Not Ready to its host. (*Not Ready* means disabled for both reads and writes.) After the R2 device becomes synchronized with its R1 device, you can split the R2 device from the (R1) device at any time, making the R2 device fully accessible again to its host. After the split, the target (R2) device contains valid data and is available for performing business continuance tasks through its original device address or restoring (copying) data to the source (R1) device if there is a loss of data on that device. Figure 1 shows a typical SRDF configuration.

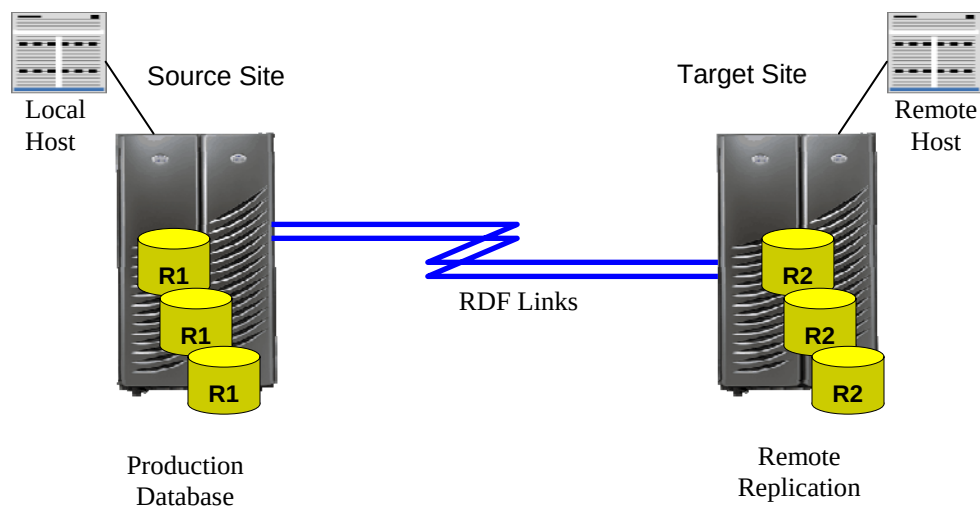


Figure 1. Typical SRDF Configuration

Disaster Restart and Disaster Recovery

Classical *disaster recovery* techniques have evolved over several decades. In most cases, a disaster recovery activity implies usage of data tapes that have been stored offsite in a secure location. *Full* backup tapes of disk data are usually taken periodically—often during a low transaction period, such as a Saturday or Sunday night. During the rest of the week, *incrementals* (or tapes capturing changed disk data since the last full or previous incremental backup) are gathered and sent offsite. By being stored offsite, these tapes use geographic separation to guard against any local disaster, such as a fire or a flood. In a disaster situation, the user must gather all the tapes and apply them in sequence. Considering the times associated with running the full and incremental tape backups from disk, packaging the tapes for offsite transport, gathering them in the event of a disaster, and recovering them back to disk storage at the backup site, 48 hours may be considered a realistic expectation of the duration of a disaster recovery activity. These activities are also susceptible to human error, such as the tapes incorrectly applied in the disaster recovery

sequence, lost tapes, damaged tapes, incompatible tapes, etc. Once the remote site is running, a similarly lengthy outage (usually involving tapes) occurs to *go home* once repairs have been made at the original data center.

Disaster restart, on the other hand, does not use computer tapes. Rather, data is transported by communications links to remote data storage. The remote replica of data serves as the restart point, and the user may restart the application using disk images at the remote site.

A major issue is whether the data in the remote location is logically consistent. Both SRDF/S synchronous replication and SRDF/A asynchronous replication ensure the dependent write order consistency of the replication by synchronizing each and every dependent I/O (in the case of SRDF/S synchronous replication) or by synchronizing cycled chunks of data (in the case of SRDF/A asynchronous replication). In addition, in certain situations, such as where there is no physical damage to the data center, the source side of the SRDF copy operation can operate on the remote mirror without having to relocate the workload to the remote site. In a true physical disaster at the source location, database restart operations can be completed at the remote site without the delays associated with finding and applying tapes in the correct sequence. Because the remote site has physical disk replicas, the *go home* activity is likewise very fast and easy.

In addition to disaster restart benefits, the SRDF product family significantly enhances disaster recovery operations by using fast and reliable replication technology to offload the Oracle backup operations to a remote site and later return the restored data to the local site.

When a disaster recovery solution is required, the only way to create valid Oracle backups with any split mirror or snapshot technology is to ensure that the database is in hot backup mode during the split or to use the Oracle Recovery Manager (RMAN). Refer to Oracle documentation for further details regarding RMAN.

SRDF Family Products

The SRDF/S product provides the following methods of data replication:

- *Synchronous* replication provides realtime mirroring of data between the source Symmetrix and the target Symmetrix systems. Data is written simultaneously to the cache of both systems in real time before the application I/O is completed, thus ensuring the highest possible data availability.
- *Adaptive Copy* replication transfers data from the source devices to the remote devices without waiting for an acknowledgement. This is especially useful when transferring large amounts of data during data center migrations, consolidations, and in data mobility environments.

SRDF/A asynchronous replication places host writes into chunks (or *delta sets*) and then transfers the entire delta set to the target system. When a complete delta set is received on the R2 side, the copy cycle is committed. If the RDF links are lost during data transfer, any partial delta set is discarded, preserving consistency on the R2. This method provides a consistent point-in-time R2 image that is not far behind the R1 side and results in minimal data loss if there is a disaster at the source site.

It is not the purpose of this paper to provide detailed information about the various data replication methods used in SRDF family products and options, or to contrast them with other traditional approaches to data replication. Refer to the *EMC SRDF Family Data Sheet* and other publications for such information.

The following sections provide further information on the operation of the SRDF/S and SRDF/A data replication methods.

SRDF/S Synchronous Replication

With SRDF/S synchronous replication, shown in Figure 2, each and every I/O from the local host is first written to the local Symmetrix cache (1) and is then sent over the RDF links to the remote Symmetrix unit (2). Once the remote Symmetrix unit reports that the data has reached its cache successfully (3), the I/O is acknowledged to the local host (4).

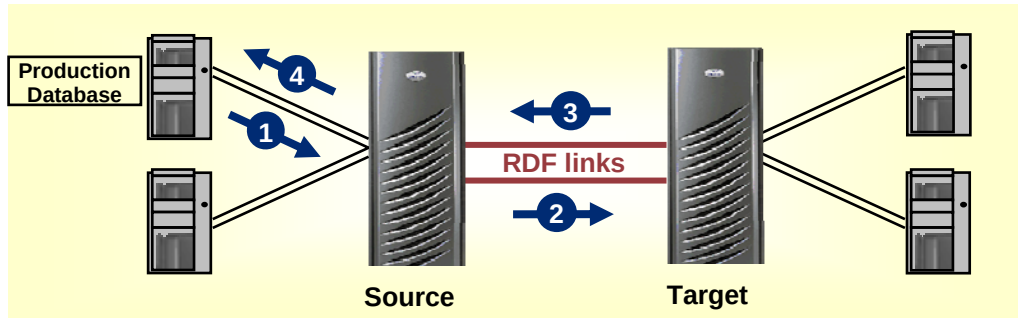


Figure 2. SRDF/S Synchronous Replication

SRDF/S synchronous replication guarantees that the remote image is a complete duplication of the source image.

Using the SRDF/Consistency Groups (SRDF/CG) Option with SRDF/S Synchronous Replication

A *consistency group* is a group of Symmetrix devices configured to act in unison to maintain the integrity of a database distributed across multiple Symmetrix units. Applications that are protected using the SRDF/CG option can be restarted at a target location almost immediately and with zero data exposure.

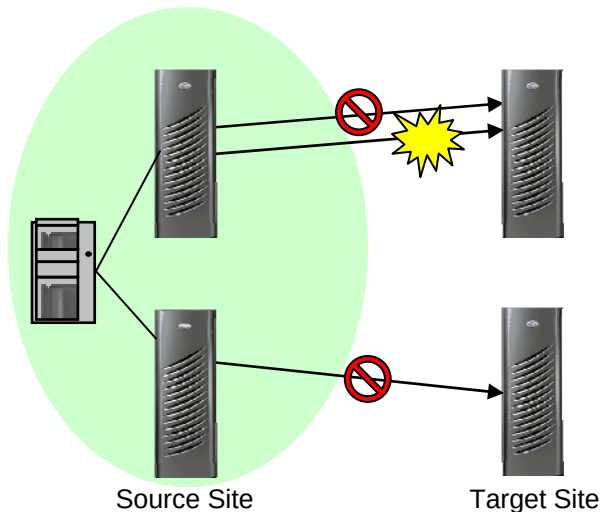


Figure 3. Failure in an SRDF/CG Environment

In Figure 3, a single database is distributed across two source Symmetrix units and replicated to two target Symmetrix units through three RDF links. If one of the RDF links fails and the other two continue to

transmit data, the remote replication becomes inconsistent and the database is not protected. By defining consistency groups, a failure to one of the links automatically and *immediately* blocks any transmission on the other links, thereby ensuring database consistency and protection.

SRDF/S Adaptive Copy Replication

SRDF/S adaptive copy replication facilitates long-distance data sharing and migration (see Figure 5). The two types of SRDF/S adaptive copy replication allow the primary and secondary volumes to be more than one I/O out of synchronization. The maximum number of I/Os that can be out of synchronization is known as the maximum skew value, and can be set using SRDF/S monitoring and control software. There is no attempt to preserve the ordering of write I/Os when using SRDF/S adaptive copy replication.

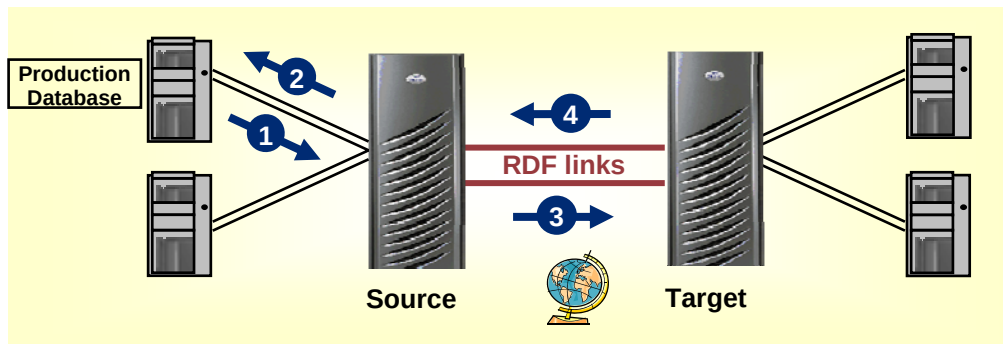


Figure 5. SRDF/S Adaptive Copy Replication

There are two types of SRDF/S adaptive copy replication:

- Adaptive Copy—Write Pending (acp_wp) mode
- Adaptive Copy—Disk (acp_disk) mode.

Both modes allow write tasks to accumulate on the local system before being sent to the remote system. Adaptive Copy—Write Pending mode allows write tasks to the target Symmetrix unit to accumulate in the cache of the source Symmetrix unit, while in Adaptive Copy—Disk mode they are allowed to accumulate on the source Symmetrix (R1) devices.

Adaptive Copy—Disk mode should not be used if the source volumes are not protected by local mirroring or Parity RAID.

When to Use SRDF/S Adaptive Copy Replication

SRDF/S adaptive copy replication is useful as an interim step before changing to an Oracle-supported SRDF/S synchronous or SRDF/A asynchronous replication. It is also used for the long-distance bulk transfer of data initiated by Solutions Enabler SRDF commands, or by other EMC automated solutions, such as SRDF/AR. For example, if the connection between the two sides is lost for a long period of time allowing the build up of a large number of changes to accumulate, resumption of the links can cause a heavy surge in link traffic (created by the backlog of changes added to those generated by normal production traffic). By using SRDF/S adaptive copy replication, the backlog of invalid tracks is synchronized using the SRDF low priority queue, while new writes are buffered in cache and sent across using the high priority SRDF queue without impacting the host application. Once the backlog of changes has been transferred, or the total amount of changed tracks has reached a specified number, the source database and applications are halted and the local and remote devices are synchronized and split. Optionally, instead of splitting the devices, you can change to SRDF/S synchronous or SRDF/A asynchronous replication to achieve database protection.

SRDF/S adaptive copy replication is *not* supported for database restart or database recovery solutions with Oracle databases. Using SRDF/S adaptive copy replication by itself for disaster protection of Oracle databases will lead to a corrupt and unusable remote database.

SRDF/A Asynchronous Replication

Beginning with Solutions Enabler Version 5.3 and Enginuity Version 5670, you can use EMC's innovative Delta Set asynchronous product SRDF/Asynchronous (SRDF/A) to copy data to remote devices using an all-or-nothing paradigm similar to consistency groups. SRDF/A provides a point-in-time image on the target (R2) device that is only slightly behind the source (R1) device. SRDF/A session data is transferred to the remote Symmetrix unit in predefined timed cycles, or chunks (also called *delta sets*), and is illustrated in Figure 4. This process eliminates redundant transfers of the same track changes, potentially reducing the bandwidth required to replicate data.

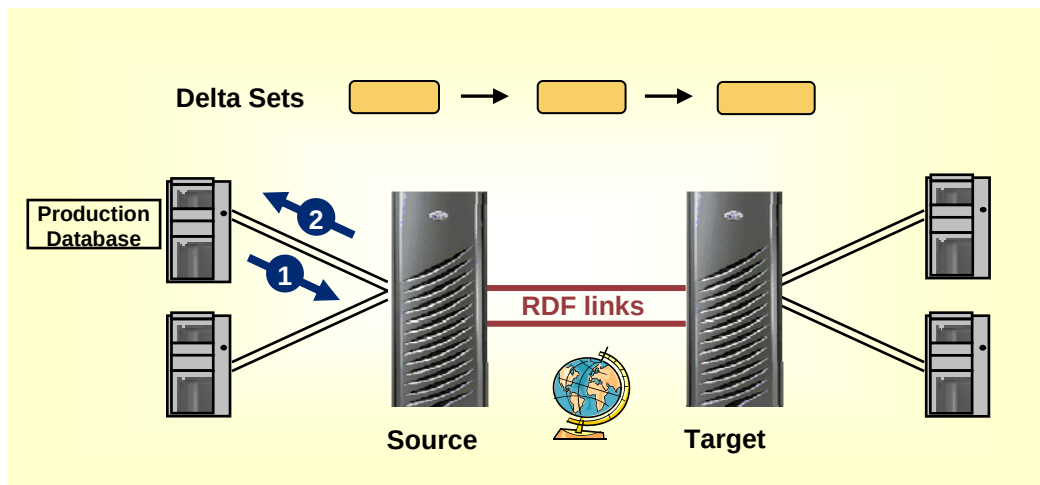


Figure 4. SRDF/A Asynchronous Replication

SRDF/A provides a long-distance replication solution with minimal impact on performance. This level of protection is intended for customers who require minimal host application impact while maintaining a consistent, restartable image of their data on the remote site. In the event of a disaster at the local site, or if the RDF links are lost during data transfer, the partial chunk of data is discarded, preserving consistency on the remote site with a data loss of no more than two SRDF/A delta sets. Note that there will always be some loss of committed transactions when using SRDF/A to replicate a live production database.

SRDF/A utilizes the system cache to buffer I/Os. Data flows into the system at a certain rate (the channel write I/O bandwidth), where it is buffered (creating an additional cache requirement) and sent out through the inter-controller links (outgoing bandwidth) to the remote controller, where it is buffered again (additional cache) before being written to the remote disk. The key to a successful implementation requires configuring the amount of additional cache to sustain SRDF/A operation. Cache requirements depend on the workload in the customer environment and available bandwidth (network infrastructure), and the system must have enough cache to buffer the incoming I/Os during periods of heavy workloads. For specific requirements, please refer to your local EMC contact.

SRDF/A and Oracle Databases

Similar to the behavior of SRDF/S synchronous replication, SRDF/A is also required to maintain dependent write order consistency of an Oracle database at the remote site. However, instead of keeping the order of every dependent write I/O, SRDF/A preserves dependent write consistency between delta sets. On the source Symmetrix unit, two consecutive I/Os will be included in the same delta set, or the later one will

enter the next delta set, thereby maintaining I/O ordering between delta sets. Each delta set is committed **in full** at the remote site to R2 devices before the next delta set starts the commit process. The SRDF/A mechanism guarantees successful completion of each commit process and the remote data image always maintains dependent write consistency when the SRDF state is Consistent.

SRDF/MSC Multi-Session Consistency

Beginning with Enginuity 5671, consistency protection for SRDF/A asynchronous devices is provided using Multi-Session Consistency (MSC). If one or more source (R1) devices in an SRDF/A MSC-enabled RDF consistency group cannot propagate data to their corresponding target (R2) devices, the MSC process suspends data propagation from *all* R1 devices in the consistency group, halting all data flow to the R2 targets. MSC enhances SRDF/A by providing consistency protection across multiple SRDF groups and/or multiple Symmetrix units. Refer to the white paper *Using SYMCLI to Implement RDF Consistency Protection with SRDF Family Products* (P/N 300-000-284) for further information on MSC.

SRDF/Automated Replication (SRDF/AR)

SRDF/Automated Replication (SRDF/AR) provides disaster restart protection for Oracle and other databases. This SRDF family option automates EMC TimeFinder and SRDF replication technology to provide restartable images of databases. SRDF/AR is not considered to be an Oracle-certified SRDF replication operation since it is the TimeFinder/Mirror consistent split process that is actually responsible for creating the restartable database image. Refer to the white papers *Validating EMC TimeFinder with Oracle8i, Oracle9i, and Oracle 10g* (P/N H1090) and *Using SYMCLI to Perform SRDF/AR* (P/N 300-000-078) for further information on SRDF/AR.

SRDF Family Operations

Using the EMC Solutions Enabler CLI (Command Line Interface), some of the basic SRDF operations between source and target devices are:

- **Full Establish:** Fully synchronize R1 and R2 devices. While established, R1 devices are accessible for R/W operations by local hosts, while R2 devices are inaccessible to any remote hosts.

```
root:> symrdf -full establish
```

- **Split:** Break the synchronization and make R2 devices accessible to any remote hosts. R1 devices remain accessible to local hosts.

```
root:> symrdf split
```

- **Re-establish:** Incrementally refresh R2 devices from their R1 devices (only changed device tracks are copied). While R1 devices remain accessible to local hosts, the R2 devices are made inaccessible to any remote hosts.

```
root:> symrdf establish
```

- **Restore:** After a split, incrementally refresh R1 devices from their remote R2 devices. The R2 devices are made inaccessible to any remote hosts. R1 devices are made accessible to local hosts as soon as the restore operation is started.

```
root:> symrdf restore
```

- **Suspend:** Halt the synchronization (usually temporarily). Unlike a split operation, the suspend operation will not make R2 devices accessible to any remote hosts. R1 devices remain accessible to local hosts.

```
root:> symrdf suspend
```

- **Resume:** After a suspend operation, re-enable SRDF synchronization.

```
root:> symrdf resume
```

- **Checkpoint:** When using SRDF/A, guarantee that all data from R1 devices has reached the remote site. This command is especially important after entering Oracle Hot Backup mode using SRDF/A replication. The checkpoint operation ensures that hot backup information has reached the remote site before a split operation can take place.

```
root:> symrdf checkpoint
```

After a failure or a disaster, some of the basic states and operations that can take place are:

- **Failover:** If the source site becomes unavailable, perform a failover. The failover operation will make R2 devices accessible to any remote hosts until the source site becomes available.

```
root:> symrdf failover
```

Note: Using SRDF/S or SRDF/A, if any of the R1 devices fail, then I/O requests by local hosts are satisfied by the corresponding R2 devices to prevent unwanted failover.

- **Update:** When the local site becomes available after a disaster, the update operation can be used to refresh R1 devices to be as close as possible to their active R2 devices before a failback operation is initiated (see below).

```
root:> symrdf update
```

-
- **Failback:** When operations are ready to resume at the local site, remote applications can be shut down and a failback operation can take place. At that time, R2 devices will become unavailable to any remote hosts and R1 devices will get refreshed with the current information from their R2 devices. Local hosts can access R1 devices as soon as the failback operation starts, giving transfer priority to requested data that has not yet reached R1 devices.

```
root:> symrdf failback
```

- In addition, the status of the RDF link and the R1 and R2 devices can be queried.

```
root:> symrdf query
```

Best Practices for Using SRDF/S Synchronous Replication and Oracle RDBMS

In addition to the best practices described for each of the following solutions, it is recommended that you keep an up-to-date copy of the Oracle database parameters file, any network configuration files, and any password files at the remote site. This will save time if the local site becomes unavailable and the database is started at the remote site.

In the procedures described next, Solutions Enabler commands issued from a host connected to the local Symmetrix unit are designated `local:>`, while those issued from a host connected to the remote Symmetrix unit are designated `remote:>`.

Refer to Appendix A for details of the configuration used in the following solutions and scenarios.

Disaster Restart Solutions Using SRDF/S Synchronous Replication

Restart Solution 1: Full Database Protection (No Data Loss)

This solution uses SRDF/S synchronous replication and positions the entire database on R1 devices. In the event of a disaster, the R2 image of the database can be restarted in the same manner as after a power failure. There is no need for manual media recovery.

This is a disaster restart solution and should never be used for backup of the remote database image (refer to the following section on *Backup Solutions*).

Best Practices

Position all local Oracle data files, online logs, and control files on R1 devices:

```
local:> symdg create AllDB -type RDF1
local:> symld -g AllDB addall -range 000:008
local:> symld -g AllDB add dev 03A
```

Enable SRDF/S synchronous replication and synchronize the R1 and R2 devices:

```
local:> symrdf -g AllDB set mode sync
local:> symrdf -g AllDB establish -full
local:> symrdf -g AllDB verify -synchronized -i 30
```

Restart Solution 2: Using a Remote TimeFinder/Mirror Split

Using SRDF/S synchronous replication as a restart solution protects the Oracle database as long as the SRDF status is Synchronized. However, if an SRDF split operation is performed, then the protection is temporarily compromised. By using TimeFinder/Mirror BCVs on the remote site, SRDF/S protection is not interrupted.

While Oracle is open for read/write on the local site, a manual (planned) TimeFinder/Mirror consistent split is performed on established TimeFinder/Mirror BCVs at the remote site to create a point-in-time restartable database image. For example, use this process to start up a remote database instance for reporting.

This is a disaster restart solution and should never be used for backup of the remote database image (refer to the following section on *Backup Solutions*).

Best Practices

Position all local Oracle data files, online logs, and control files on R1 devices:

```
local:> symdg create AllDB -type RDF1
local:> symld -g AllDB addall -range 000:008
local:> symld -g AllDB add dev 03A
```

Associate all R2 devices with TimeFinder/Mirror BCVs on the remote Symmetrix unit:

```
remote:> symdg create AllDB -type RDF2
remote:> symld -g AllDB addall dev -range 000:008
remote:> symld -g AllDB add dev 03A
remote:> symbcv -g AllDB associateall dev -range 196:19E
remote:> symbcv -g AllDB associate dev 1A0
```

Enable SRDF/S synchronous replication and synchronize the R1 and R2 devices:

```
local:> symrdf -g AllDB set mode sync
local:> symrdf -g AllDB establish -full
local:> symrdf -g AllDB verify -synchronized -i 30
```

Establish the remote TimeFinder/Mirror BCVs:

```
remote:> symmir -g AllDB establish -full
remote:> symmir -g AllDB verify -synched -i 30
```

Perform a consistent split on the remote TimeFinder/Mirror BCVs:

```
remote:> symmir -g AllDB split -consistent
```

Note that any further TimeFinder/Mirror or SRDF establish operations can be incremental.

Backup Solutions Using SRDF/S Synchronous Replication

In the following backup solutions, all Oracle SQL commands are issued from a host connected to the local Symmetrix unit.

Backup Solution 1: Using a Remote TimeFinder/Mirror Split

While Oracle is open for read/write on the local site, a manual (planned) TimeFinder/Mirror instant split is performed on established TimeFinder/Mirror BCVs at the remote site to create a remote mirror image of the active database that can serve as a backup source.

Best Practices

Position all local Oracle data files on R1 devices. In this example, Oracle archive logs are also positioned on R1 devices that will be split later. (Optionally, `ftp`, an additional archive log destination, or third-party software can be used to send the archive logs to the remote site.)

```
local:> symdg create AllDB -type RDF1
local:> symld -g AllDB addall -range 000:008
local:> symld -g AllDB add dev 03A
```

Associate all R2 devices with TimeFinder/Mirror BCVs on the remote Symmetrix unit:

```
remote:> symdbg create datafiles -type RDF2
remote:> symld -g datafiles addall dev -range 000:005
remote:> symbcv -g datafiles associateall dev -range 196:19B
remote:> symdbg create archivlogs -type RDF2
remote:> symld -g archivlogs add dev 039
remote:> symbcv -g archivlogs associate dev 19F
```

Enable SRDF/S synchronous replication and synchronize the R1 and R2 devices:

```
local:> symrdf -g AllDB set mode sync
local:> symrdf -g AllDB establish -full
local:> symrdf -g AllDB verify -synchronized -i 30
```

Establish the remote TimeFinder/Mirror BCVs:

```
remote:> symmir -g datafiles establish -full
remote:> symmir -g archivlogs establish -full
remote:> symmir -g datafiles verify -synched -i 30
remote:> symmir -g archivlogs verify -synched -i 30
```

Enable Archivelog mode (enable automatic archival) for the Oracle database:

```
SQL:> archive log list;
##### If log archiving is not enabled, then #####
##### follow Oracle procedures to enable it #####
```

Put all Oracle tablespaces in Hot Backup mode:

```
SQL:> alter tablespace SYSTEM begin backup;
SQL:> alter tablespace UNDO_1 begin backup;
SQL:> alter tablespace SP_0 begin backup;
```

Split the remote TimeFinder/Mirror BCVs that contain the data files:

```
remote:> symmir -g datafiles split -instant
```

Note that any further TimeFinder/Mirror or SRDF establish operations can be incremental.

Take all Oracle tablespaces out of Hot Backup mode:

```
SQL:> alter tablespace SYSTEM end backup;
SQL:> alter tablespace UNDO_1 end backup;
SQL:> alter tablespace SP_0 end backup;
```

Archive the current online redo log:

```
SQL:> alter system archive log current;
```

Copy the backup control file:

```
SQL:> alter database backup controlfile to '/oracle9/dbs/arch/bckcontrol.ctl';
```

Split the archive logs together with the backup control file. (Optionally, FTP, an additional archive log destination, or third-party software can be used to send the archive logs to the remote site.)

```
remote:> symmir -g archivlogs split -instant
```

Backup Solution 2: Using an SRDF Split for Read-Write Databases

While Oracle is open for read/write on the local site, a manual (planned) SRDF split is used to create a point-in-time remote mirror image of the active database that can serve as a backup source.

Best Practices

Position all local Oracle data files on R1 devices:

```
local:> symdg create datafiles -type RDF1
local:> symld -g datafiles addall -range 000:005
```

Enable SRDF/S synchronous replication and synchronize the R1 and R2 devices:

```
local:> symrdf -g datafiles set mode sync
local:> symrdf -g datafiles establish -full
local:> symrdf -g datafiles verify -synchronized -i 30
```

Enable Archivelog mode (enable automatic archival) for the Oracle database:

```
SQL:> archive log list;
##### If log archiving is not enabled, then #####
##### follow Oracle procedures to enable it #####
```

If you do not plan to split archive logs, then they have to be available on the remote site. (Use FTP, an additional archive log destination, or third-party software to send them to the remote site.) In this example, archive logs are on R1 devices and will be split later.

```
local:> symdg create archivelogs -type RDF1
local:> symld -g archivelogs add dev 039
local:> symrdf -g archivelogs establish -full
local:> symrdf -g archivelogs verify -synchronized -i 30
```

Put all Oracle tablespaces in Hot Backup mode:

```
SQL:> alter tablespace SYSTEM begin backup;
SQL:> alter tablespace UNDO_1 begin backup;
SQL:> alter tablespace SP_0 begin backup;
```

Split the R2 devices containing the Oracle data files:

```
local:> symrdf -g datafiles split
```

Take all Oracle tablespaces out of Hot Backup mode:

```
SQL:> alter tablespace SYSTEM end backup;
SQL:> alter tablespace UNDO_1 end backup;
SQL:> alter tablespace SP_0 end backup;
```

Archive the current online redo log:

```
SQL:> alter system archive log current;
```

Create a backup control file and send it to the remote site. Optionally, if you plan to split the archive logs as well, the backup control file can be saved to the directory containing the archive logs.

```
SQL:> alter database backup controlfile to '/oracle9/dbs/arch/bckcontrol.ctl';
```

If you plan to split the archive logs, then split them now. You may use FTP to send them to the remote site. If an Oracle additional archive log destination is used, then the archive logs are shipped via the network interface to the remote site automatically. Third-party software can also be used to transfer archive logs.

```
local:> symrdf -g archive logs split
```

If the database is on a file system, it is recommended that you flush the file system as close to the split as possible (use UNIX: sync; OCFS: Flush; etc.).

Backup Solution 3: Using an SRDF Split for Read-Only Databases

If the local database is in Read-Only mode, then data is not modified, redo logs and archive logs are not generated, and Hot Backup mode is not involved.

Best Practices

Position all local Oracle data files on R1 devices:

```
local:> symdg create datafiles -type RDF1
local:> symld -g datafiles addall -range 000:005
```

Enable SRDF synchronous replication and synchronize the R1 and R2 devices:

```
local:> symrdf -g datafiles set mode sync
local:> symrdf -g datafiles establish -full
local:> symrdf -g datafiles verify -synchronized -i 30
```

Split the R2 devices containing the Oracle data files:

```
local:> symrdf -g datafiles split
```

Copy the backup control file and send it to the remote site:

```
SQL:> alter database backup controlfile to '/oracle9/dbs/arch/bckcontrol.ctl';
local:> rcp /oracle9/dbs/bckcontrol.ctl remote:/oracle9/dbs/bckcontrol.ctl
```

Backup Solution 4: Using an SRDF Split for Closed Databases

When the local database is shut down “gracefully” (i.e., shut down using the Normal, Immediate, or Transactional option) before the split mirror image is taken, the process is called a “Consistent Whole Database Backup”. The database is made available remotely and can be either backed up or started. Note that once the remote database is opened for read/write, it becomes no longer valid for backup of the production database.

Best Practices

Position all local Oracle data files, online logs, and control files on R1 devices:

```
local:> symdg create AllDB -type RDF1
local:> symld -g AllDB addall -range 000:008
local:> symld -g AllDB add dev 03A
```

Enable SRDF/S synchronous replication and synchronize the R1 and R2 devices:

```
local:> symrdf -g AllDB set mode sync
local:> symrdf -g AllDB establish -full
local:> symrdf -g AllDB verify -synchronized -i 30
```

Shut down the database using the Normal, Immediate, or Transactional option:

```
SQL:> shutdown immediate;
```

Unmount the data files if they are mounted on a file system and split the R2 devices:

```
local:> symrdf -g AllDB split
```

Mount the data files on the file system (if applicable) and restart the database:

```
SQL:> startup;
```

Failure Remediation Scenarios Using SRDF/S Synchronous Replication

Failure Remediation Scenario 1: Resuming Protection after a Link Failure

If RDF link failure occurs, the SRDF/S status will be Partitioned. The R1 devices are still available and the production database is undisturbed. The following procedure may be used to achieve database protection once the RDF link failure is corrected.

Best Practices

Enable SRDF/S synchronous replication:

```
local:> symrdf -g AllDB set mode sync
```

Re-establish the SRDF relationships:

```
local:> symrdf -g AllDB establish  
local:> symrdf -g AllDB verify -synchronized -i 30
```

Note that the database is not protected until resynchronization is completed. To maintain database protection at all times, it is recommended that you use a remote TimeFinder/Mirror BCV split before resynchronization starts.

When the RDF links become available, in some cases (determined by the status of devices) synchronization will start automatically as SRDF reverts to its state before the failure.

Failure Remediation Scenario 2: Failover to Remote Site

After a failure at the local site, the R1 devices are not reachable and the SRDF status will be Partitioned. To resume operations at the remote site, the R2 devices need to be made available to a remote host so that the database can be started. The following procedure may be used to restart the database on the remote host.

All operations are done from the remote host.

Best Practices

Read/Write enable the R2 devices by using the `symrdf failover` command. **Note that the device group AllDB must be defined on the remote Symmetrix unit.**

```
remote:> symrdf -g AllDB failover
```

If the database was using a volume manager and/or file systems, then import the volumes to the remote host and/or run FSCK for the file systems before mounting them.

Start the Oracle database on the remote host without any media recovery:

```
SQL:> startup;
```

Failure Remediation Scenario 3: Failback to Local Site

After a failure at the local site, start the Oracle database at the remote site (see previous scenario). Once the problem has been investigated and fixed and the local site is ready to continue operations, a failback operation should be initiated. The following procedure resumes operations at the local site.

Best Practices

Ensure the R1 devices are reachable and the RDF links are operational.

While the database is still running at the remote site, start an update operation of R1 devices with recent data using the `-until` option to continuously update the R1 devices to a specified level of changed tracks:

```
remote:> symrdf -g AllDB set mode sync
remote:> symrdf -g AllDB update R1 -until 1000
```

When the update has completed, on the remote host, stop Oracle and any other applications using the R2 devices:

```
SQL:> shutdown immediate;
```

If the database was using a volume manager and/or file systems, then export the volumes and/or unmount the file systems.

Perform an SRDF failback operation to the local site:

```
remote:> symrdf -g AllDB failback
remote:> symrdf -g AllDB verify -synchronized -i 30
```

On a host connected to the local Symmetrix unit, start the database and applications on the local site (import volumes and/or mount file systems first if applicable):

```
SQL:> startup;
```

It is possible to start using the R1 volumes as soon as the failback operation starts. In such a case, the database on the local site will not be protected until full synchronization is reached. It is recommended that you first take a TimeFinder/Mirror BCV image of the database at the remote site before starting the failback operation.

Best Practices for Using SRDF/A Asynchronous Replication and Oracle RDBMS

In addition to the best practices described for each of the following solutions, it is recommended that you keep an up-to-date copy of the Oracle database parameters file, any network configuration files, and any password files at the remote site. This will save time if the local site becomes unavailable and the database is started at the remote site.

In the procedures described next, Solutions Enabler commands issued from a host connected to the local Symmetrix unit are designated `local:>`, while those issued from a host connected to the remote Symmetrix unit are designated `remote:>`.

Refer to Appendix A for details of the configuration used in the following solutions and scenarios.

Disaster Restart Solutions Using SRDF/A Asynchronous Replication

Restart Solution 1: Full Database Protection

This solution uses SRDF/A asynchronous replication and positions the entire database on R1 devices. In the event of a disaster, a predicted data loss of two cycle chunks may occur. The R2 image of the database can be restarted in the same manner as after a power failure. There is no need for manual media recovery.

This is a disaster restart solution and should never be used for backup of the remote database image (refer to the following section on *Backup Solutions*).

Best Practices

Position all local Oracle data files, online logs, and control files on R1 devices:

```
local:> symdg create All -type RDF1
local:> symld -g All addall -range 000:008
local:> symld -g All addall -range 039-03A
```

Enable SRDF/A asynchronous replication and synchronize the R1 and R2 devices. Note that the database is not protected until the SRDF state becomes Consistent.

```
local:> symrdf -g All set mode async
local:> symrdf -g All establish -full
local:> symrdf -g All verify -consistent -i 30
```

Restart Solution 2: Using a Remote TimeFinder/Mirror Split

Using SRDF/A as a restart solution protects the Oracle database as long as the SRDF status is Consistent. However, if an SRDF split operation is performed, then the protection is temporarily compromised. By using TimeFinder/Mirror BCVs on the remote site, SRDF/A protection is not interrupted.

While Oracle is open for read/write on the local site, a manual (planned) TimeFinder/Mirror consistent split is performed on established TimeFinder/Mirror BCVs at the remote site to create a point-in-time restartable database image. A TimeFinder/Mirror consistent split coordinates the split with SRDF/A cycles. For example, use this process to start up a remote database instance for reporting.

This is a disaster restart solution and should never be used for backup of the remote database image (refer to the following section on *Backup Solutions*).

Best Practices

The remote Oracle image can be successfully started after performing the split. Due to the asynchronous nature of SRDF/A, a predicted data loss of up to two delta sets may occur. In addition, the TimeFinder/Mirror consistent split option must be used to coordinate the split with SRDF/A. Note that the SRDF state must be Consistent before making the split.

Currently, the `-consistent` option must be used when performing remote TimeFinder/Mirror split operations. Note that TimeFinder/Snap and TimeFinder/Mirror clones are not currently supported for remote split operations.

Position all local Oracle data files, online logs, and control files on R1 devices:

```
local:> symdg create All -type RDF1
local:> symld -g All addall -range 000:008
local:> symld -g All addall -range 039:03A
```

Associate the appropriate R2 devices with TimeFinder/Mirror BCVs on the remote Symmetrix unit:

```
remote:> symdg create AllDB -type RDF2
remote:> symld -g AllDB addall dev -range 000:008
remote:> symld -g AllDB add dev 03A
remote:> symbcv -g AllDB associateall dev -range 196:19E
remote:> symbcv -g AllDB associate dev 1A0
```

Enable SRDF/A asynchronous replication and synchronize the R1 and R2 devices:

```
local:> symrdf -g All set mode async
local:> symrdf -g All establish -full
local:> symrdf -g All verify -consistent -i 30
```

Establish the remote TimeFinder/Mirror BCVs:

```
remote:> symmir -g AllDB establish -full
remote:> symmir -g AllDB verify -synched -i 30
```

Perform a consistent split on the remote TimeFinder/Mirror BCVs:

```
remote:> symmir -g AllDB split -consistent
```

Note that any further TimeFinder/Mirror or SRDF establish operations can be incremental.

Restart Solution 3: Performing an SRDF Split

While Oracle is open for read/write on the local site, a manual (planned) SRDF split is performed to create a point-in-time restartable database image.

This is a disaster restart solution and should never be used for backup of the remote database image (refer to the following section on *Backup Solutions*).

Best Practices

The remote Oracle image can be successfully started after performing the split. Due to the asynchronous nature of SRDF/A, a predicted data loss of up to two delta sets may occur.

Position all local Oracle data files, online logs, and control files on R1 devices:

```
local:> symdg create All -type RDF1
local:> symld -g All addall -range 000:008
local:> symld -g All addall -range 039:03A
```

Enable SRDF/A asynchronous replication and synchronize the R1 and R2 devices:

```
local:> symrdf -g All set mode async
local:> symrdf -g All establish -full
local:> symrdf -g All verify -consistent -i 30
```

Split the R2 devices:

```
local:> symrdf -g All split
```

To re-establish the R2 devices at a later time, ensure that any remote applications (for example, the Oracle RDBMS) using the R2 devices are stopped and unmount any file systems. Then re-establish the R2 devices:

```
local:> symrdf -g All establish
local:> symrdf -g All verify -consistent -i 30
```

Backup Solutions Using SRDF/A Asynchronous Replication

In the following backup solutions, all Oracle SQL commands are issued from a host connected to the local Symmetrix unit.

Backup Solution 1: Using a Remote TimeFinder/Mirror Split

While Oracle is open for read/write on the local site, a manual (planned) TimeFinder/Mirror consistent split is performed on established TimeFinder/Mirror BCVs at the remote site to create a remote mirror image of the active database that can serve as a backup source.

Best Practices

Position all local Oracle data files on R1 devices. In this example, Oracle archive logs are also positioned on R1 devices that will be split later. (Optionally, FTP, an additional archive log destination, or third-party software can be used to send the archive logs to the remote site.)

```
local:> symdg create All -type RDF1
local:> symld -g All addall -range 000:008
local:> symld -g All addall -range 039:03A
```

Associate the appropriate R2 devices with TimeFinder/Mirror BCVs on the remote Symmetrix unit:

```
remote:> symdg create datafiles -type RDF2
remote:> symld -g datafiles addall dev -range 000:005
remote:> symbcv -g datafiles associateall dev -range 196:19B
remote:> symdg create archivelogs -type RDF2
remote:> symld -g archivelogs add dev 039
remote:> symbcv -g datafiles associate dev 19F
```

Enable SRDF/A asynchronous replication and synchronize the R1 and R2 devices:

```
local:> symrdf -g All set mode async
local:> symrdf -g All establish -full
local:> symrdf -g All verify -consistent -i 30
```

Establish the remote TimeFinder/Mirror BCVs:

```
remote:> symmir -g datafiles establish -full
remote:> symmir -g archivelogs establish -full
remote:> symmir -g datafiles verify -synched -i 30
remote:> symmir -g archivelogs verify -synched -i 30
```

Note that any further TimeFinder/Mirror or SRDF establish operations can be incremental.

Enable Archivelog mode (enable automatic archival) for the Oracle database:

```
SQL:> archive log list;
##### If log archiving is not enabled, then #####
##### follow Oracle procedures to enable it #####
```

Put all Oracle tablespaces in Hot Backup mode:

```
SQL:> alter tablespace SYSTEM begin backup;
SQL:> alter tablespace UNDO_1 begin backup;
SQL:> alter tablespace SP_0 begin backup;
```

Issue an SRDF checkpoint to ensure that changes to the R1 devices have reached the R2 devices:

```
local:> symrdf -g All checkpoint
```

Perform a consistent split on the remote TimeFinder/Mirror BCVs that contain the data files:

```
remote:> symmir -g datafiles split -consistent
```

Take all Oracle tablespaces out of Hot Backup mode:

```
SQL:> alter tablespace SYSTEM end backup;
SQL:> alter tablespace UNDO_1 end backup;
SQL:> alter tablespace SP_0 end backup;
```

Archive the current online redo log:

```
SQL:> alter system archive log current;
```

Copy the backup control file:

```
SQL:> alter database backup controlfile to '/oracle9/dbs/arch/bckcontrol.ctl';
```

Perform a second SRDF checkpoint operation to update the R2 devices, and then split the archive logs together with the backup control file. (Optionally, FTP, an additional archive log destination, or third-party software can be used to send the archive logs and backup control file to the remote site.)

```
local:> symrdf -g All checkpoint
remote:> symmir -g archivelogs split -consistent
```

Backup Solution 2: Using an SRDF Split for Read-Write Databases

While Oracle is open for read/write on the local site, a manual (planned) SRDF split is used to create a point-in-time remote mirror image of the active database that can serve as a backup source.

Best Practices

Position all local Oracle data files on R1 devices:

```
local:> symdg create All -type RDF1
local:> symld -g All addall -range 000:008
local:> symld -g All addall -range 039:03A
```

Enable SRDF/A asynchronous replication and synchronize the R1 and R2 devices:

```
local:> symrdf -g All set mode async
local:> symrdf -g All establish -full
local:> symrdf -g All verify -consistent -i 30
```

Enable Archivelog mode (enable automatic archival) for the Oracle database:

```
SQL:> archive log list;
##### If log archiving is not enabled, then #####
##### follow Oracle procedures to enable it #####
```

Put all Oracle tablespaces in Hot Backup mode:

```
SQL:> alter tablespace SYSTEM begin backup;
SQL:> alter tablespace UNDO_1 begin backup;
SQL:> alter tablespace SP_0 begin backup;
```

Issue an SRDF checkpoint to ensure that changes to the R1 devices have reached the R2 devices:

```
local:> symrdf -g All checkpoint
```

Split the R2 devices containing the Oracle data files:

```
local:> symrdf -g All split
```

Take all Oracle tablespaces out of Hot Backup mode:

```
SQL:> alter tablespace SYSTEM end backup;
SQL:> alter tablespace UNDO_1 end backup;
SQL:> alter tablespace SP_0 end backup;
```

Archive the current online redo log:

```
SQL:> alter system archive log current;
```

Copy the backup control file:

```
SQL:> alter database backup controlfile to '/oracle9/dbs/arch/bckcontrol.ctl';
```

Send the archive logs and backup control files to the remote site:

```
local:> ftp remote
ftp> bin
ftp> prompt
ftp> cd /oracle9/dbs/arch
ftp> put bckcontrol.ctl
ftp> mput *.arc
ftp> bye
```

To re-establish the R2 devices at a later time, ensure that any remote applications (for example, the Oracle RDBMS) using the R2 devices are stopped and unmount any file systems. Then re-establish the R2 devices.

```
local:> symrdf -g All establish
local:> symrdf -g All verify -consistent -i 30
```

Failure Remediation Scenarios Using SRDF/A Asynchronous Replication

Failure Remediation Scenario 1: Resuming Protection after a Link Failure

If RDF link failure occurs, the SRDF status will be Partitioned. The R1 devices are still available and the production database is undisturbed. The following procedure may be used to achieve database protection once the RDF link failure is corrected.

Best Practices

If a large number of changes have accumulated on the R1 devices during the link failure, then start low-priority synchronization using SRDF/S adaptive copy replication. (If synchronization is a priority, then use SRDF/S synchronous replication.)

```
local:> symrdf -g All set mode acp_wp
local:> symrdf -g All establish
```

Enable SRDF/A asynchronous replication:

```
local:> symrdf -g All set mode async
```

Re-establish the SRDF relationships:

```
local:> symrdf -g All establish
local:> symrdf -g All verify -consistent -i 30
```

Note that the database is not protected until resynchronization is completed. To maintain database protection at all times, it is recommended that you use a remote TimeFinder/Mirror BCV split before resynchronization starts.

When the RDF links become available, in some cases (determined by the status of devices) synchronization will start automatically as SRDF reverts to its state before the failure.

Failure Remediation Scenario 2: Failover to Remote Site

After a failure at the local site, the R1 devices are not reachable and the SRDF status will be Partitioned. To resume operations at the remote site, the R2 devices need to be made available to a remote host so that the database can be started. The following procedure may be used to restart the database on the remote host.

All operations are done from the remote host.

Best Practices

Read/Write enable the R2 devices by using the `symrdf failover` command. **Note that the device group All must be defined on the remote Symmetrix unit.**

```
remote:> symrdf -g All failover
```

If the database was using a volume manager and/or file systems, then import the volumes to the remote host and/or run `FCK` for the file systems before mounting them.

Start the Oracle database on the remote host without any media recovery:

```
SQL:> startup;
```

Failure Remediation Scenario 3: Failback to Local Site

After a failure at the local site, start the Oracle database at the remote site (see previous scenario). Once the problem has been investigated and fixed and the local site is ready to continue operations, a failback operation should be initiated. The following procedure resumes operations at the local site.

Best Practices

Ensure the R1 devices are reachable and the RDF links are operational.

While the database is still running at the remote site, start an update operation of R1 devices with recent data using the `-until` option to continuously update the R1 devices to a specified level of changed tracks. Use SRDF/S adaptive copy replication for the update operation so that database activity on the R2 devices is not adversely affected. (If the update is a priority, then use SRDF/S synchronous replication.)

Note that the device group All must be defined on the remote Symmetrix unit.

```
remote:> symrdf -g All set mode acp_wp
remote:> symrdf -g All update R1 -until 1000
```

When the update has completed, on the remote host, stop Oracle and any other applications using the R2 devices:

```
SQL:> shutdown immediate;
```

If the database was using a volume manager and/or file systems, then export the volumes and/or unmount the file systems.

Perform an SRDF failback operation to the local site. **EMC recommends using SRDF/S synchronous replication (rather than SRDF/S adaptive copy replication) during the failback operation so that the SRDF devices are fully synchronized before restarting SRDF/A operation.** Using SRDF/S synchronous replication prioritizes the synchronization and the SRDF state will become Consistent immediately when SRDF/A operation is resumed, thereby ensuring full database protection.

```
remote:> symrdf -g All set mode sync
remote:> symrdf -g All failback
remote:> symrdf -g All verify -synchronized -i 30
```

It is possible to start using the R1 volumes as soon as the failback operation starts. In such a case, the database on the local site will not be protected until full synchronization is reached, and it is recommended that you first take a TimeFinder/Mirror BCV image of the database at the remote site before starting the failback operation.

Enable SRDF/A asynchronous replication:

```
local:> symrdf -g All set mode async
local:> symrdf -g All verify -consistent -i 30
```

On a host connected to the local Symmetrix unit, start the database and applications on the local site (import volumes and mount file systems first if applicable):

```
SQL:> startup;
```