

SRDF/Asynchronous (SRDF/A)
Open Systems
Version 1.0

Technical Note
P/N 300-001-346
REV A03
April 30, 2004

This technical note contains supplemental information about EMC SRDF/A Version 1.0. Topics include:

♦ Introduction	2
♦ SRDF/A Product Description	2
♦ Behavioral Differences between SRDF and SRDF/A	6
♦ SRDF/A Operational Considerations	8
♦ SRDF/A and Databases	11
♦ Conclusion	13

Introduction

EMC SRDF®/A is a new architectural approach to asynchronous extended-distance remote replication. Using *delta set* technology, SRDF/A enables customers to maintain a restartable remote copy of data at all times, at long distances, with no host application impact. SRDF/A provides customers with an economical and high-performance replication solution for business continuity. Customers can meet greater service levels that cannot be obtained with traditional asynchronous architectures.

Built with the underpinnings of the proven SRDF software, SRDF/A is the optimal extended-distance replication solution when a customer's service-level requirements dictate that economics and application performance are more critical than zero data exposure.

Purpose and Scope

The SRDF/A user interface is designed to make it behave similarly to SRDF on open systems. It is invoked through the `symrdf` command and permits the use of SRDF actions such as split, establish, failover, failback, etc. In limited cases, the differences between SRDF and SRDF/A lead to different results and information displays as an outcome of the same commands.

The purpose of this document is to highlight the differences between SRDF and SRDF/A and to recommend procedures to facilitate the use of SRDF/A.

Related Documentation

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

- ♦ *WideSky Solutions Enabler Symmetrix SRDF CLI Product Guide*, P/N 300-000-877
- ♦ *Using SYMCLI to Perform SRDF Control Operations*, P/N 300-000-076

SRDF/A Product Description

Beginning with Solutions Enabler Version 5.3 and Enginuity™ Version 5670, the Symmetrix® system supports the new asynchronous replication product named SRDF/Asynchronous (SRDF/A). Asynchronous mode

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 system in pre-defined timed cycles or *delta sets*, eliminating the redundancy of same track changes being transferred over the link, potentially reducing the required bandwidth.

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 R2 side. In the event of a disaster at the R1 site or if RDF links are lost during data transfer, a partial delta set of data can be discarded, preserving consistency on the R2 with a data loss of no more than two SRDF/A cycles.

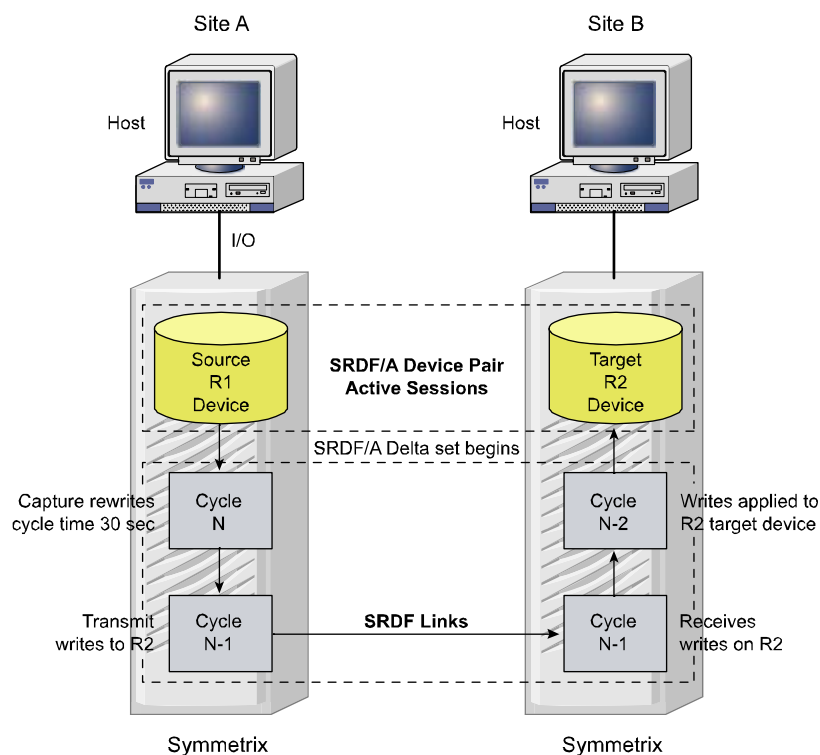


Figure 1. SRDF/A Transfers Data Using Cycles (Delta Sets)

Write Ordering in SRDF/A

Different from traditional ordered-write approaches, the Symmetrix system implements Asynchronous mode host writes from the source to the target using pre-determined timed cycles (delta sets). Each delta set contains groups of I/Os for processing. Using cycles of operation,

SRDF/A transfers sets of data one cycle at a time between the R1 and the R2. If the same track is written to more than one time within an active set, SRDF/A will send the update over the link only once. This approach lowers the use of link bandwidth as compared with other ordered-write processing approaches that transfer each and every write separately. Refer to Figure 1 for a depiction of SRDF/A delta sets.

Dependent Write Consistency

All commonly used database management systems are inherently dependent write consistent. For instance, a DBMS will not perform a log write, indicating that a transaction is complete, until it has received an acknowledgement from the storage subsystem that the log data pertaining to the transaction itself was completely written to disk. The Symmetrix system honors this logic to maintain write ordering in SRDF/A by treating any successor I/O (B), which arrives after a predecessor I/O (A), as a dependent I/O.

Dependent write consistency is achieved through the processing of ordered SRDF/A delta sets between the source (R1) and the target (R2). Figure 1 depicts the SRDF/A cycles. The Symmetrix system ensures that if a dependent write (B) is in cycle (N), then its predecessor write (A) is in cycle (N) or in an earlier cycle such as (N-1). Dependent write consistency ensures that all writes to the R2 are processed in sequential numbered sets to maintain a consistent copy of data between the R1 and the R2.

When the SRDF/A cycle (N) is active on the source Symmetrix system, it collects any new writes in the R1 Symmetrix cache, overwriting any duplicate tracks intended for data transfer over the link. The cycle is active for a pre-determined amount of time that can be configured on the Symmetrix system at the time of the initial configuration of the SRDF/A environment; the default time is 30 seconds. After the set time has been reached, the delta set data inherits the next cycle position (N-1) and begins transferring the delta set over the link to the R2. A new cycle N then begins collecting new writes again for the next delta set transfer.

In cycle (N-1), the delta set is temporarily collected on the R2 side for destaging. When the (N-1) cycle has finished transferring data into the R2 and the minimum cycle time has elapsed, the delta set inherits the next cycle position (N-2) and begins destaging the data to the R2 storage devices. The delta set data is considered committed to the R2 in cycle (N-2).

Three conditions must be fulfilled before a cycle switch occurs:

- ♦ The cycle time (default value: 30 sec.) must have elapsed.
- ♦ The (N-1) cycle must have been fully received at the target site.
- ♦ The (N-2) cycle must have been marked as device write pendings or destaged to disk.

The cycle is elongated if the write transfer or destaging exceeds the set cycle time. A new cycle cannot begin until all three events have occurred. All data is transferred at the block level.

When all writes belonging to delta set (N-2) data have been committed to the R2 target device, the R1 and R2 are considered to be in the *consistent-pair state*.

Note that the movement of delta sets inside a Symmetrix system is performed by a switching of pointers to different regions of cache, instead of actual copying of data from one location to another inside Symmetrix cache. This is a key contributing feature to the high performance of SRDF/A.

SRDF/A Restrictions

There are certain current restrictions and limitations for running SRDF/A-capable devices in Asynchronous mode. The following is a list of current restrictions that apply:

- ♦ Only one SRDF/A group is permitted per Symmetrix DMX™. All SRDF/A-capable devices must be managed together by one device group or device list in an SRDF/A session.
- ♦ SRDF/A-capable devices that are enabled for Consistency Group protection must be disabled before attempting to change from Asynchronous mode to another mode.
- ♦ Asynchronous mode is currently limited to single-hop configurations using Enginuity Version 5670.

The following RDF control operations and configurations are currently **not** supported for SRDF/A-capable devices:

- ♦ SRDF/A devices cannot be part of a concurrent RDF configuration.
- ♦ SRDF Automated Replication (SRDF/AR).
- ♦ Dynamic RDF operations including the `createpair`, `deletpair`, and `swap` commands. Swapping RDF personalities such that all R1s become R2s and all R2s become R1 devices using the Symmetrix Configuration Change CLI (also known as the Config Manager) are allowed.
- ♦ Snap and clone operations are not permitted on the R2; they are permitted on the R1.

Behavioral Differences between SRDF and SRDF/A

Synchronization of Invalid Tracks

Whenever the tracks on the source and target sides of an SRDF configuration are not identical, SRDF marks the tracks that were created on the local side without having been transmitted to the remote side as “remote invalids.” For example, if the links between R1 and R2 are logically suspended and new writes are generated on the R1 side, these tracks are marked as R2 invalids (i.e., when the links come back up again, those tracks will need to be sent across from the R1 to the R2 side).

When running SRDF in Synchronous, Semi-synchronous, or Adaptive Copy - Disk mode, invalid tracks are sent across as full track writes via the low-priority RDF queue maintained on each Symmetrix system. This queue has a limit to the number of I/Os it will accept. It is fed by the back-end disk directors that are also responsible for destaging host I/O from cache to disk. The SRDF synchronization process is carried out in the background, while giving priority to the front end I/O traffic on the Symmetrix system. On a Symmetrix system that has a heavy front-end load, the rate of SRDF synchronization can be slowed down even further by the use of Quality of Service settings, accessible through the `symqos` command and documented in the *Symmetrix Base Control CLI* manual.

If SRDF/A is active, invalid tracks are sent across in discrete quantities through the SRDF/A cycles or delta sets without using either the high-priority or low-priority SRDF queues.

SRDF/A Behavior after Unexpected Loss of Links

The design goal of SRDF/A is to minimize application impact in the event of link-related problems. There is a timer within SymWin known as the *link limbo* parameter that will allow SRDF to tolerate the loss of physical connectivity for a certain amount of time (default value: 10 seconds) before concluding that the links have failed.

The logical connection between the source and target can be unexpectedly interrupted due to one of several reasons:

- ♦ Loss of physical connection between source and target due to network problems
- ♦ Symmetrix system dropping the link because of link saturation and its inability to keep up with write traffic
- ♦ User-initiated action such as the use of `symrdf -immediate split`

When the logical connection between source and target is unexpectedly interrupted:

- ♦ I/Os that are part of the Apply delta set ($N-2$ cycle) are safely stored on disk. This is a dependent write consistent image of data.
- ♦ I/Os that are part of the Receive delta set ($N-1$ cycle on the R2 side) are marked as remote invalid tracks on the R2 side (i.e., the R2 marks those tracks as owed to the R1).
- ♦ I/Os that are part of the Transmit delta set ($N-1$ cycle on the R1 side) and the Capture delta set (N cycle on the R1 side) are marked as remote invalid tracks on the R1 side (i.e., the R1 marks those tracks as owed to the R2).
- ♦ New incoming writes are marked as remote invalids (i.e., they are marked by the R1 as owed to the R2).

In SRDF, remote invalids are seen on both sides after new writes have occurred on both sides subsequent to the links being logically split. With SRDF/A, remote invalids will occur on both sides without new data having been created on the R2. When the links come back up and a resynchronization is initiated, the track tables are first merged, and then the appropriate tracks are copied from one side to the other depending on whether an establish or a restore is undertaken.

Planned Suspension of Links

Sometimes it may be necessary to suspend SRDF/A traffic for the purpose of planned activities, such as server maintenance, disaster drill, testing with the remote image of data, etc. The first step necessary before terminating the SRDF/A session is to disable consistency protection using the command:

```
symrdf disable
```

After consistency protection is disabled, is it possible to issue commands such as:

```
symrdf suspend
symrdf split
symrdf failover
```

without the `-force` option.

Switching from SRDF/A to Another Mode

It is sometimes necessary to change from SRDF/A mode to Synchronous, Semi-synchronous, or Adaptive Copy mode due to temporary operational necessities. To accomplish this, consistency protection must be first disabled as shown earlier.

If mode switching is performed while the links are intact and traffic is flowing across the links, the commands:

```
symrdf set mode sync
symrdf set mode semi
symrdf set mode acp_disk
symrdf set mode acp_wp
```

will cause the current Apply and Transmit/Receive delta sets ($N-2$ and $N-1$ cycles) to be completed and the outstanding I/Os in the Capture delta set (N cycle) to be marked as R2 invalids on the R1 side. As long as the two sides are not fully synchronized, consistency is no longer preserved on the R2 side.

SRDF/A Operational Considerations

The following considerations are important when planning an SRDF/A implementation.

SRDF/A and the Link Limbo Parameter

The link limbo parameter is the length of time that the Symmetrix system will tolerate the loss of links before concluding that the link has failed. It can be adjusted for each individual RDF group from the service processor. By default, the link limbo timer is set to 10 seconds, measured from the time that the RDF director discovers that the link is lost, till the time that that the physical connectivity is restored *and* the RDF directors complete the full initialization of the link. If the directors cannot complete full initialization of the link within this time, the link limbo timer expires and the RDF director communicates to the other directors in the Symmetrix that the link is down. Thus even a short physical link outage of 1 second could result in the timer expiring and the links being dropped by the Symmetrix.

From the software perspective, after the link is down, a query of the SRDF group will show the state as *Partitioned*. By default, most Symmetrix systems have their *Prevent Automatic RDF Link Recovery* parameter set to *Enabled*. Consequently, once the network has been re-initialized and connectivity has been reinstated, the SRDF state will appear to be *Suspended*. For a discussion on steps to recover from a link failure, please refer to *SRDF/A Behavior after Unexpected Loss of Links* on page 6.

Raising the value of the link limbo parameter is generally not recommended but can be permitted for SRDF/A implementations in special cases if the customer determines that it would be beneficial in his

environment. A change in the link limbo parameter has to be requested through the pre-sales qualifier (PSQ) document that precedes every SRDF/A implementation. The amount of cache configured for use by SRDF/A will need to be higher for a higher link limbo setting.

Enabling and Disabling Consistency

The primary value proposition of SRDF/A is to provide a consistent restartable image of user data at the remote site without affecting application performance. To enable consistency protection, it is necessary to issue the command:

```
symrdf enable
```

Enabling consistency guarantees consistency of the R2 copy of data at all times. If the logical connection between the two sides is interrupted, either due to a planned or unplanned event, consistency must be explicitly enabled after the resumption of link traffic. When an exit from SRDF/A mode is planned, consistency can be disabled by using the command:

```
symrdf disable
```

Recovering after Loss of Links

Prior to starting a resynchronization after link recovery, EMC recommends that a BCV copy of the R2 volumes be split off and set aside to guarantee the availability of a consistent image of data, to guard against an unforeseen interruption of the resynchronization process.

If logical connection between the two sides is lost for a long period of time allowing for the buildup of a large number of R2 invalids on the R1 side, resumption of the links with SRDF/A enabled (the default behavior in open systems) can cause a heavy surge of link traffic, created by the backlog of invalid tracks being added to the cycles generated by the production traffic. This can lead to an unacceptably high demand being placed on Symmetrix cache, causing SRDF/A to drop again. To avoid this risk, it is best to briefly disable SRDF/A when restarting after the build up of invalid tracks. It is suggested that resynchronization be undertaken at a time when there is a relatively low amount of incoming host writes. Recommended methods of getting back to steady state in such an event are:

- ♦ Set the SRDF mode to Adaptive Copy - Write Pending mode by using the command:

```
symrdf set mode acp_wp
```

In this mode, the invalid tracks will be synchronized and new writes will be buffered in cache and sent across without impacting the host

application. As long as the rate of incoming writes is less than the link bandwidth, this mode of operation will allow production to continue while continuing to reduce the backlog of invalid tracks. The resynchronization process may be relatively slow. While invalid tracks are being applied during the resynchronization process, the R2 data is not consistent.

- ♦ Set the SRDF mode to Synchronous and allow the resynchronization to occur. This will have the effect of slowing down the host application, while resynchronization is going on. The data on the R2 is not consistent until the invalid tracks have been reduced to zero.
- ♦ If the potentially extended resynchronization time and the consequent exposure to data loss in the event of a complete source site failure during resynchronization are deemed unacceptable, then the production application will have to be halted while the resynchronization process completes.

It is recommended that SRDF/A be enabled only after the two sides are synchronized.

SRDF/A and Rolling Disasters

EMC defines a rolling disaster as two successive failures, where the second failure occurs before recovery procedures related to the first one were completed. SRDF/A ensures that the data on the R2 side is consistent at all times, even when the source site is unavailable. As mentioned previously, once the source and target have lost connectivity, it is important to split off a BCV copy of the R2 data, *before* a resynchronization is attempted. The scenario described next is one that could happen with SRDF/A.

Let us presume that there is a buildup of invalid tracks because the R1 was being written to while the target was unreachable. This can be the result of a planned or unplanned logical suspension of links. Next, the links are resumed and a resynchronization is initiated from the R1 to the R2. Immediately after the start of resynchronization, the tracks to be copied from the R1 to the R2 get marked as local invalids on the R2 side—meaning that they need to be brought across from the R1. If SRDF/A is enabled prior to both sides reaching synchronization (not recommended), the invalid tracks are sent across in delta sets along with the write I/Os. If an unexpected loss of links now occurs before the state of the SRDF volumes is *consistent*, the R2 will now contain local invalids (caused during the prior resynchronization) and remote invalids (caused by the tracks belonging to the incomplete receive delta set). In SRDF, this is a problem requiring human intervention. In SRDF/A, this is an acceptable, though not desirable, state of affairs. This is why it is

recommended that SRDF/A be enabled only *after* both sides are synchronized.

SRDF/A and Configuration Manager

Using the SYMCLI-based Symmetrix Configuration Utility, also known as the Configuration Manager, it is possible to perform an RDF personality swap, causing all R1 devices to become R2s and R2 devices to become R1s. To facilitate such swaps, Cache Only Virtual Devices (COVDs) must be configured on both Symmetrix units. In the IMPL INIT RDF section of SymmWin, the "Reserve SNOW Resources:" value has to be marked as "YES".

However, SRDF/A devices cannot be configured by the Configuration Manager. Non-SRDF/A devices may not be converted to SRDF/A devices using Configuration Manager. COVDs may not be configured with the Configuration Manager. The help screen in Configuration Manager describes the syntax for these operations; but current restrictions in the software will not permit these actions.

SRDF/A and Databases

SRDF/A ensures that the image of the data on the R2 volumes is always consistent and restartable in the event of a failure on the source Symmetrix system. As made clear in the preceding pages, that guarantee does not apply during the resynchronization process, which is why a BCV copy of data should be split off from the R2 before recovery operations following a failure are undertaken.

After SRDF/A is enabled, at least two cycles are required after the invalid track count goes to zero before consistency can be guaranteed. The consistency status can be displayed by using the command:

```
symrdf -rdfa query
```

In some cases, a customer may want to create additional consistent copies of their data on the target Symmetrix system for various business reasons.

Consistent Database Copy with Rollback Capability

The ability to create a restartable point-in-time database image is required if you want to restart the database but do not need to create backups. The following are the steps undertaken from the host connected to the source Symmetrix system to create a restartable point-in-time database image on the remote BCV:

1. Establish a TimeFinder® BCV with the target R2 devices.
2. Wait until the BCV is fully synchronized.
3. Issue a consistent split command to split off the BCVs.

The BCVs now contain consistent images of data. It is worth noting that though the command syntax to issue a consistent TimeFinder split from a remotely connected host is the same as traditional SRDF (i.e., `symmir -rdf -consistent split`), the nature of the split is different. Instead of using Engineuity Consistency Assist (ECA) to hold the I/O to the affected Symmetrix volumes, this command will issue the TimeFinder split at a cycle boundary, when consistency is guaranteed on the R2 volumes.

Database Backups Using SRDF/A

Using SRDF/A to Create Oracle Database Backups

A restartable point-in-time database image is *not* a valid Oracle backup. The database image produced by SRDF/A is restartable, but may not be used as part of an Oracle backup plan.

A valid Oracle online backup can *only* be made in one of two ways:

- ♦ The tablespace(s) are in hot backup mode when their BCV splits are performed. This is the EMC best practice.
- or
- ♦ The restartable point-in-time image is backed up using Oracle's Recovery Manager (RMAN). Only the RMAN backup is valid, the point-in-time image on the BCV is not.

The following are the EMC best practice steps undertaken from the host connected to the source Symmetrix system to create a valid Oracle backup on the remote BCV:

1. Establish a TimeFinder BCV to the target R2 devices.
2. Wait until the BCV is fully synchronized.
3. **Place the Oracle tablespace(s) in hot backup mode.**

4. Use the `symrdf checkpoint` command to ensure that the last write issued by the application is committed to the R2.
5. Split the BCVs from the R2 using the command `symmir -rdf -consistent split`.
6. Take the tablespace(s) out of hot backup mode.
7. Force the database to archive the current online redo log.
8. Make a copy of the Control File in the archive log file system.
9. Manage the log files by either:
 - a. Copying the log files over the network.
 - b. Creating a remote BCV of the log file by:
 - Using the `symrdf checkpoint` to verify that the log was written to the target.
 - Consistently splitting the BCV from the log file disk.

Using SRDF/A to Create Database Backups with SqlServer, DB2/UDB, Sybase, or Informix

You cannot currently make valid database backups off BCVs on the R2 side using SRDF/A with these databases. These databases require a freeze of database write I/O at the DBMS level to create valid split mirror backups. The length of time that the I/O would need to be frozen in order to ensure that this frozen state has been committed to the remote BCVs is too long to be supportable.

Conclusion

SRDF/A is a high-performance, extended-distance asynchronous solution built on the solid foundation of SRDF, and inherits all of the advantages of SRDF. It is host and operating system independent, and replicates all of the enterprise's data without consuming host cycles. It can withstand a wide variety of failures, and has been proven by many years of market experience.

For an existing SRDF customer, SRDF/A is especially easy to use because it requires the user to learn very few new options to the existing command set. Further detail on the use of the command and its options can be found in the documents referenced in the *Related Documentation* section on page 2.

Copyright © 2000-2004 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, Symmetrix, Celerra, CLARiiON, CLARAlert, Documentum, HighRoad, LEGATO, Navisphere, PowerPath, ResourcePak, SnapView/IP, SRDF, TimeFinder, VisualSAN, and where information lives are registered trademarks and EMC Automated Networked Storage, EMC ControlCenter, EMC Developers Program, EMC OnCourse, EMC Proven, EMC Snap, Access Logix, AutoAdvice, Automated Resource Manager, AutoSwap, AVALONidm, C-Clip, Celerra Replicator, Centera, CentraStar, CLARevent, Connectrix, CopyCross, CopyPoint, DatabaseXtender, Direct Matrix, Direct Matrix Architecture, EDM, E-Lab, Engenuity, FarPoint, FLARE, GeoSpan, InfoMover, MirrorView, NetWin, OnAlert, OpenScale, Powerlink, PowerVolume, RepliCare, SafeLine, SAN Architect, SAN Copy, SAN Manager, SDMS, SnapSure, SnapView, StorageScope, SupportMate, SymmAPI, SymmEnabler, Symmetrix DMX, Universal Data Tone, and VisualSRM are trademarks of EMC Corporation. All other trademarks used herein are the property of their respective owners.