



Global Distributed Storage Engineering

SRDF/A for Long Distance Replication

GDSE-05002-SG

Solutions Guide

Version 2.0

VERSION CONTROL

Version and Date of Issue	Author	Comment
V1.0.8 15th June 2005	Chris Clayton	First public draft
V1.1 11th July 2005	Chris Clayton	Additional operational procedural information and feedback from Scott Higgins included. Appendix B added with description of operational scripts
V1.2 4 th August 2005	Chris Clayton	Incorporated detailed Peer Review feedback from Stephen Hopkins. Other minor changes.
V 1.3 7 th September 2005	Chris Clayton	Updated recommendations on number of RDF groups per RA.
V1.4 24 th February 2006	Chris Clayton	Update following feedback from EMC.
V2.0 draft 11 th December 2006	Chris Clayton	Annual update.
V2.0 24 th January 2004	Chris Clayton	Included Peer review changes requested by Ludwig Derodel. Also Peer review feedback on Appendix C by Stephen Hopkins and Colin Bradley.

CONTENTS

Legal Notice	11
Contact	11
Latest Versions	11
Confidentiality	11
1 DOCUMENT SCOPE	12
1.1 Out of scope	13
1.2 Intended Audience	13
1.3 Changes in Version 2.0	13
1.4 Changes in Version 1.4	14
2 INTRODUCTION TO SRDF/A	16
2.1 Overview of SRDF/A	16
2.1.1 Product Highlights	16
2.1.2 How SRDF/A works	17
2.1.3 Role of SRDF/A in Disaster Recovery	18
2.2 SRDF Terminology	18
2.2.1 Consistency	19
3 APPLICABLE ENVIRONMENTS	21
3.1 Recovery Point Objective	21
3.2 Example SRDF/A configurations	21
3.2.1 2-site SRDF/A configuration	22
3.2.2 Concurrent (3-site) SRDF configuration	23
3.2.3 SRDF/A over long latency links	24
4 PREPARATION FOR AN SRDF/A IMPLEMENTATION	25
4.1 Introduction	25
4.2 Collection of Basic Application Requirements	26
4.2.1 Determining the appropriate SRDF configuration	26
4.2.2 Scalability limitations	30
4.2.3 Determining the number of RDF groups that are required	30
4.3 EMC SRDF/A Solution Qualification	32
4.3.1 Introduction	32
4.3.2 Overview of Solutions Qualification Process	32

4.4	Background information on Network Bandwidth Sizing	34
4.4.1	Comment on network quality	35
4.5	Network Hardware	36
4.5.1	Overview of Network equipment configuration	36
4.6	Management server requirements	38
4.6.1	Supported platforms	38
4.6.2	Source site	38
4.6.3	Target sites	38
4.7	Storage Hardware Considerations	39
4.7.1	Cache requirements	39
4.7.2	Additional RA requirements	39
4.7.3	Additional note on RDF groups per RA limit	40
4.8	DMX bin file configuration	43
4.8.1	Microcode levels	43
4.8.2	Dynamic RDF bin file requirements	43
4.8.3	BCV requirements at target site	43
4.8.4	BCV usage constraints at source site	45
4.8.5	Additional Gatekeeper requirements	45
4.8.6	Hot Spares and Symoptimiser	45
4.9	Storage Software Considerations	46
4.9.1	SRDF/A licensing	46
4.9.2	Solutions Enabler 6.x	46
4.9.3	InfoSec implications	46
4.10	Other planning considerations	47
4.10.1	Dynamic RDF	47
4.10.2	Performance Impact Considerations	47
4.11	Large Network Latency Implementations	48
4.11.1	Reduced throughput and stability with packet loss at 250ms latency	48
4.11.2	Effective bandwidth as a function of latency	48
4.11.3	Cycle time increases with latency for MSC replication	48
4.11.4	Long runtime for control/query commands	48
5	GENERIC SRDF/A CONFIGURATION	49
5.1	Management Server Configuration	50
5.1.1	Solutions Enabler 6	50
5.1.2	Software Licensing	50
5.1.3	/etc/system	51
5.1.4	/var/symapi/config/options	51
5.1.5	/var/symapi/config/daemon_options	51
5.1.6	Gatekeeper requirements	52
5.1.7	symacl configuration requirements	52
5.2	Network Equipment Configuration	53
5.2.1	McData IPS configuration	53
5.2.2	Cisco Switch configuration	53
5.3	Dynamic RDF configuration	54
5.3.1	Introduction	54

5.3.2	Dynamic RDF requirements	54
5.3.3	Mapping RDF groups to RAs	55
5.3.4	Creating dynamic RDF groups	55
5.3.5	Creating dynamic RDF device pairs	57
5.3.6	Checking the status of dynamic RDF device pairs	58
5.4	Target site Point-In-Time image for SRDF/A resynchronisation	59
5.4.1	Overview of requirement	59
5.4.2	Best Practice Summary	59
5.4.3	Mapping PIT devices to standards	59
5.4.4	Adding BCVs to device groups	59
5.5	Device group definitions	61
5.6	Enabling SRDF/A replication	62
5.7	Worked Example of a 2-site SRDF/A configuration	63
5.7.1	Link Limbo timer setting	65
5.7.2	Cycle time	65
5.7.3	Priority	65
5.7.4	SRDF/A Cache Utilization	66
5.8	SRDF/A Replication Monitoring	67
6	2-SITE MULTI-SESSION CONSISTENCY	68
6.1	Overview of MSC	68
6.2	High-availability of MSC replication	69
6.2.1	Overview	69
6.3	Configuring MSC support on management servers	70
6.3.1	Gatekeeper requirements	70
6.3.2	High-availability of RDF daemon	70
6.4	Use of Composite groups	71
6.4.1	Scalability restrictions	71
6.4.2	Creating the Composite Group	71
6.4.3	Synchronizing Composite Group definitions across management server	72
6.4.4	Enabling MSC replication	72
6.5	Cycle time under MSC	74
7	CONCURRENT 3-SITE SRDF/A CONFIGURATION	75
7.1	General comments	75
7.2	Licensing	75
7.3	Device group definitions	75
7.3.1	Configuration 2: 3 site SRDF/A with mirrored R2s. No subset failover	78
7.3.2	Configuration 3: 3 site SRDF/A with mirrored R2s. With subset failover	79
7.3.3	Configuration 4: 3-site SRDF/A with unmirrored Sync R2 + COB BCV. No subset failover.	80

7.3.4	Configuration 5: 3-site SRDF/A with unmirrored Sync R2 + COB BCV. With subset failover.	81
7.4	Worked example of a non-Star 3-site RDF group configuration	82
7.5	Multi-Session Consistency with Concurrent SRDF	86
8	OPERATIONAL PROCEDURES	87
8.1	Storage provisioning with multi-session SRDF/A	87
8.1.1	Overview of issues	87
8.1.2	Creating a dynamic RDF group	88
8.1.3	Setting RDF group parameters	88
8.1.4	Adding devices to a dynamic RDF group – 2-site implementation	88
8.1.5	Adding devices to a dynamic RDF group – Non-Star 3-site implementation	90
8.1.6	Removing devices from a dynamic RDF group – 2 site implementation	92
8.1.7	Removing devices from a dynamic RDF group – non-Star 3-site concurrent implementation	93
8.1.8	Removing a dynamic RDF group	94
8.2	Starting and stopping SRDF/A	95
8.2.1	Starting SRDF/A	95
8.2.2	Suspending SRDF/A	95
8.2.3	Restarting SRDF/A after controlled suspension	96
8.2.4	Restarting SRDF/A after uncontrolled suspension	96
8.2.5	Processing multiple device groups	96
8.3	Restarting SRDF/A after extended network outage	98
8.3.1	Introduction	98
8.3.2	Recovering after loss of links	98
8.4	Single Application Failover/Failback – 2-site SRDF/A configuration	100
8.5	Single Application Failover/Failback – non-Star 3-site configuration	101
8.5.1	Concurrent SRDF - Synchronous leg failover/failback	101
8.5.2	Concurrent SRDF - Asynchronous leg failover/failback	103
8.6	Subset failover/failback on Synchronous leg – non-Star 3-site configuration	105
8.6.1	Introduction	105
8.6.2	Executing subset failover/failback	108
8.7	SRDF Personality Swapping	111
8.7.1	2-site process (non-MSD)	111
8.7.2	2-site process (MSD)	112
8.7.3	3-site process	112
8.8	2-Site Multi-Session Consistency Operations	113
8.8.1	Independent RDF group failover/failback	113
8.8.2	Composite group failover/failback	114
8.9	Changing replication mode	115
8.9.1	Changing to Async mode from Sync or Adaptive copy	115
8.9.2	Changing from Async to Adaptive copy mode	115
8.9.3	Changing from Async to Sync mode	115
8.10	Interactive monitoring of SRDF/A configuration and replication	116

8.10.1	symstat	116
8.10.2	symrdf list –concurrent	119
8.10.3	symrdf –g <device group> query –rdfa –rdfg all	120
8.10.4	symcfg list –rdfg all	122
8.10.5	symcfg list –ra all –switched	123
8.10.6	symevent list	123
9	OTHER OPERATIONAL ISSUES	124
9.1	Use of BCVs at concurrent SRDF/A source sites	124
9.2	Using the SRDF/A PIT BCVs for COB testing	124
9.3	SRDF/A over long latency links	125
9.4	Initial Master Program Load (IMPL) issues	126
9.5	Troubleshooting network issues	126
9.5.1	srdfa_health script	126
9.5.2	symevent	127
9.5.3	IPS troubleshooting	127
10	APPENDIX A – BASIC REQUIREMENTS QUESTIONNAIRE	128
10.1	Definitions	128
10.2	Questions	129
10.3	Summary of Basic Requirements Questionnaire results	132
11	APPENDIX B – OPERATIONAL SCRIPTS SUMMARY	133
11.1	Summary of scripts	133
11.2	Script mon_srdfa_inactive	136
11.2.1	Overview	136
11.2.2	Invocation and required arguments	136
11.2.3	Usage notes	136
11.3	Script mon_srdfa_cache	137
11.3.1	Overview	137
11.3.2	Invocation and required arguments	137
11.3.3	Usage notes	137
11.4	Script mon_srdfa_cycle	138
11.4.1	Overview	138
11.4.2	Invocation and required arguments	138
11.4.3	Usage notes	138
11.5	Script srdfa_health	139
11.5.1	Overview	139
11.5.2	Invocation and required arguments	139
11.5.3	Usage notes	139

11.6	Script add_PIT_BCV	140
11.6.1	Overview	140
11.6.2	Invocation and required arguments	140
11.6.3	Usage notes	140
11.7	Script rm_PIT_BCV	141
11.7.1	Overview	141
11.7.2	Invocation and required arguments	141
11.7.3	Usage notes	141
11.8	Script 2site_add_device	142
11.8.1	Overview	142
11.8.2	Invocation and required arguments	142
11.8.3	Usage notes	142
11.9	Script 2site_rm_device	144
11.9.1	Overview	144
11.9.2	Invocation and required arguments	144
11.9.3	Usage notes	144
11.10	Script 3site_add_device	146
11.10.1	Overview	146
11.10.2	Invocation and required arguments	146
11.10.3	Usage notes	146
11.11	Script 3site_rm_devices	148
11.11.1	Overview	148
11.11.2	Invocation and required arguments	148
11.11.3	Usage notes	148
11.12	Script post_IMPL_reset	150
11.12.1	Overview	150
11.12.2	Invocation and required arguments	150
11.12.3	Usage notes	150
11.13	Script resume_srdfa	151
11.13.1	Overview	151
11.13.2	Invocation and required arguments	151
11.13.3	Usage notes	151
11.14	Script 2site_async_failover	152
11.14.1	Overview	152
11.14.2	Invocation and required arguments	152
11.14.3	Usage notes	152
11.15	Script 2site_async_failback	153
11.15.1	Overview	153
11.15.2	Invocation and required arguments	153
11.15.3	Usage notes	153
11.16	Script 3site_sync_failover	154
11.16.1	Overview	154
11.16.2	Invocation and required arguments	154
11.16.3	Usage notes	154
11.17	Script 3site_sync_failback	155
11.17.1	Overview	155

11.17.2	Invocation and required arguments	155
11.17.3	Usage notes	155
11.18	Script 3site_async_failover	156
11.18.1	Overview	156
11.18.2	Invocation and required arguments	156
11.18.3	Usage notes	156
11.19	Script 3site_async_failback	157
11.19.1	Overview	157
11.19.2	Invocation and required arguments	157
11.19.3	Usage notes	157
11.20	Script 2site_async_swap	158
11.20.1	Overview	158
11.20.2	Invocation and required arguments	158
11.20.3	Usage notes	158
11.21	Script 2site_MSC_failover	159
11.21.1	Overview	159
11.21.2	Invocation and required arguments	159
11.21.3	Usage notes	159
11.22	Script 2site_MSC_failback	160
11.22.1	Overview	160
11.22.2	Invocation and required arguments	160
11.22.3	Usage notes	160
11.23	Script 2site_CG_failover	161
11.23.1	Overview	161
11.23.2	Invocation and required arguments	161
11.23.3	Usage notes	161
11.24	Script 2site_CG_failback	162
11.24.1	Overview	162
11.24.2	Invocation and required arguments	162
11.24.3	Usage notes	162
11.25	3site_subset_failover	163
11.25.1	Overview	163
11.25.2	Invocation and required arguments	164
11.25.3	Usage notes	164
11.26	3site_subset_failback	165
11.26.1	Overview	165
11.26.2	Invocation and required arguments	166
11.26.3	Usage notes	166
12	APPENDIX C - MIGRATING <u>EXISTING</u> SOURCE DMX FROM 2-SITE SRDF/S TO 2-SITE SRDF/A, USING CONCURRENT SRDF	167
12.1	Introduction	167
12.2	SAN environment planning preparation	168
12.2.1	General planning and preparation	168
12.2.2	Network infrastructure	168

12.2.3	Symmetrix changes (not in any order of priority)	168
12.2.4	SAN management server changes	170
12.3	Application migration planning	171
12.3.1	General comments	171
12.3.2	Existing BCVs on source Symmetrix	171
12.3.3	RDF groups	171
12.3.4	Device groups definitions	172
12.4	Application COB migration	174
12.4.1	Setting up Concurrent SRDF	174
12.4.2	Changes to operational procedures when temporarily running in concurrent SRDF mode	176
12.5	Post-migration steps	178
12.5.1	Cease SRDF/S replication	178
12.5.2	Modify source site device groups	179
12.5.3	Implement Multi-Session consistency (if required)	179
12.5.4	Update local procedures	179
12.5.5	Physical cleanup	180
13	APPENDIX D – SELECTION OF POINT-IN-TIME TECHNOLOGY FOR SRDF/A RESYNCHRONIZATION PROTECTION	181
13.1	Background information	181
13.1.1	TimeFinder Modes	181
13.1.2	Use of RAID-5 to protect RAID1+0 standards	181
13.1.3	Symoptimizer / hot spare issue	182
13.2	Best Practice Recommendation	183
13.2.1	SRDF/S – RAID1+0 R2s	183
13.2.2	2-site SRDF/A & 3-Site concurrent SRDF – RAID1+0 R2s	183
13.2.3	3-site SRDF/Star – RAID1+0 R2s	184
13.2.4	All SRDF configurations with RAID5 R2s	184
13.3	“FAQs”	185
13.3.1	Why is the use of RAID-5 BCVs not being recommended for RAID1+0 R2s?	185
13.3.2	Why is the use of symclone not being recommended?	185
13.3.3	Why is the use of larger disks not being recommended for PiT copies?	185
13.3.4	What are the downsides of sticking with “Classic” TimeFinder/Mirror rather than using clone emulation mode?	186
13.4	Mapping SRDF/A point-in-time copies to R2s	186

Legal Notice

Information contained herein is for Internal Use and may be used only for business purposes authorized by Citigroup.

CITI, CITIBANK, CITICORP, CITIGROUP and the Umbrella Device are trademarks and service marks of Citicorp and are used and registered throughout the world. Citibank, Citicorp, Citigroup and their subsidiaries also claim rights in certain other trademarks and service marks contained in these pages.

Contact

Please address any comments or questions regarding the information contained in this document via email to the “*GT Global Distributed Storage Engineering” distribution list or alternatively to gdse@citigroup.com.

Latest Versions

It is recommended that the latest version of the document be used from the GDSE website

<http://engineering.ti.citigroup.net/gdse/products/>

Newer versions may include expanded scope, or comments on issues discovered after deployment.

Confidentiality

No part of this document should be shared by EMC with any of its other customers. Any unauthorized review, use, disclosure or distribution is prohibited without express permission.

1 Document Scope

This Solutions Guide recommends Best Practice for using SRDF/A technology on EMC DMX within Citigroup. It is not a replacement for the Vendor documentation referenced below. The following EMC documentation should also be consulted when implementing SRDF/A.

Furthermore, this Solutions Guide must be read in conjunction with the GDSE document SRDF/A Release Notes, which covers (or references) in more detail product certification, known issues and test results (both functional and performance).

Document	Location
EMC documentation	
EMC Solutions Enabler Symmetrix SRDF Family CLI Product Guide Version 6	Available from EMC
EMC Engineering White Paper: Using SYMCLI to Perform Control Operations with SRDF Family Products	Available from EMC
EMC Solutions Enabler Symmetrix Access Control CLI Product Guide	Available from EMC
GDSE documentation	
SRDF/A Evaluation	http://engineering.ti.citigroup.net/gdse/products/platforms/symmetrix/stdsdocs/05002-e.pdf
SRDF/A Release Notes	http://engineering.ti.citigroup.net/gdse/products/platforms/symmetrix/releasenotes/05002-rn.pdf
SRDF/Star for Open Systems Solutions Guide	http://engineering.ti.citigroup.net/gdse/products/platforms/symmetrix/stdsdocs/06050-sg.pdf
McData IPS SAN Router Solutions Guide v2	http://engineering.ti.citigroup.net/gdse/products/platforms/mcdata/aips/stdsdocs/06030-sg.pdf
McData IPS SAN Router Release Notes v2	http://engineering.ti.citigroup.net/gdse/products/platforms/mcdata/aips/releasenotes/06030-rn.pdf
WAN Throughput Optimization for SRDF/A Solutions Guide	http://engineering.ti.citigroup.net/gdse/products/platforms/symmetrix/stdsdocs/06034-sg.pdf
TimeFinder Clone Solutions Guide	http://engineering.ti.citigroup.net/gdse/products/platforms/symmetrix/stdsdocs/05008-sg.pdf

TimeFinder Clone Release Notes	http://engineering.ti.citigroup.net/gdse/products/platforms/symmetrix/releasenotes/05008-rn.pdf
EMC Symmetrix DMX Best Practice	http://engineering.ti.citigroup.net/gdse/products/platforms/symmetrix/stdsdocs/03038-sg.pdf
EMC Solutions Enabler 6.3 Solutions Guide	http://engineering.ti.citigroup.net/gdse/products/platforms/symmetrix/stdsdocs/05016-sg.pdf

1.1 Out of scope

- Performance information. This can be found in the SRDF/A evaluation document.
- COB processes that are not directly related to SRDF/A-specific issues.
- SRDF/Star operations (see separate SRDF/Star Solutions Guide in references).
- Multi-Session Consistency in Concurrent (3-site) configurations.
- Consistency operations between multiple source Symmetrix
- Best Practice for McData IPS (see separate Solutions Guide in references).
- Tuning SRDF/A throughput with the McData IPS (see separate Solutions Guide in references).
- SRDF/A control from platforms other than Solaris (not supported by Citigroup)

1.2 Intended Audience

This detailed document is aimed at implementing an SRDF/A solution. It is not intended as an overview for end users, developers or Project Managers.

1.3 Changes in Version 2.0

Throughout document – References to SE6.0 changes to refer to more generic 6.x version, and users instructed to use latest GDSE certified version of SE 6.

Throughout document – References to 5671 changed to refer to more generic 5x71 version to accommodate DMX3.

Section 3.2.2 – Updated to state that multi-session Concurrent SRDF is now available, but only supported within Citigroup in an SRDF/Star configuration.

Section 3.2.3 – Updated to reference the new WAN throughput Optimization for SRDF/A Solutions Guide.

Section 4.1 – Emphasis that SRDF/A can currently only be deployed for existing applications with measurable I/O profiles due to the lack of a “Utility SRDF/A” model from EMC.

Section 4.2 – This section has been completely re-written to accommodate the certification of SRDF/Star.

Section 4.4 – Section updates to reflect existence of the new WAN Throughput Optimization for SRDF/A Solutions Guide.

Section 4.5 – This section has been completely re-written to accommodate the certification of fabric-attached 2640 IPS units, and the existence of WAN Throughput Optimization for SRDF/A Solutions Guide.

Section 4.7 – Emphasis on the criticality of installing “balanced” systems for SRDF/A.

Section 4.8.3 – Re-written for new SRDF PiT copy Best Practice.

Section 4.9.1 – Updated licensing information.

Section 4.11.2 – Section updated to reflect existence of WAN Throughput Optimization for SRDF/A Solutions Guide.

Section 5.1.2 – Updated licensing information

Section 5.2 – Updated to reflect existence of WAN Throughput Optimization for SRDF/A Solutions Guide and new IPS solutions guide.

Section 5.3 – Various corrections to syntax for creating dynamic RDF groups.

Section 5.4 – Updated to reflect new Best Practice for creating SRDF/A resync PiT copies, and to document how to create on non-contentious source / target pairings

Section 5.7 – Re-written to reflect discuss creation on non-contentious source / target pairings

Section 6.4.1 – Update on scalability restrictions

Section 7 – Various references to SRDF/Star added with direction on superceeding Best Practice

Section 7.4 – Re-written to reflect discuss creation on non-contentious source / target pairings and to accommodate existence of SRDF/Star.

Section 7.5 – Updated to state that multi-session Concurrent SRDF is now available, but only supported within Citigroup in an SRDF/Star configuration.

Section 8 – Various references throughout to SRDF/Star Solutions Guide.

Appendix C – New appendix with advice on migrating existing source DMX from 2-site SRDF/S to 2-site SRDF/A, using concurrent SRDF.

Appendix D – New appendix with background to Best Practice for technologies to be used for SRDF/A resync PiT copies

1.4 Changes in Version 1.4

Section 3.2.2 – Reinforcement that as Multi-Session Consistency (MSC) is not yet supported by EMC in a concurrent SRDF configuration, there will not be guaranteed consistency between different RDF groups on the synchronous leg.

Section 4.7.3 – Updated with latest recommendations and limits on the number of RDF groups per RA.

Section 4.8.3 – Use of 146GB disks at the Asynchronous site is no longer recommended when the production site is using smaller/faster disks. EMC have found issues with such configurations that can cause SRDF/A to fail as the R2 is unable to keep up with the R1, resulting in cache at the source site becoming exhausted.

Section 11.26 – Correction to calling sequence for template 3-site subset failover/failback scripts.

2 Introduction to SRDF/A

2.1 Overview of SRDF/A

2.1.1 Product Highlights

SRDF/Asynchronous (SRDF/A) is a remote mirroring solution for the Symmetrix DMX series. It uses a Delta Set architecture to maintain a consistent point-in-time target (R2) image with only a slight lag behind the source (R1) side, resulting in minimal data loss if the R1 is lost. SRDF/A allows one to configure an SRDF environment in which R1 devices transfer data to R2 devices in chunk cycles.

Some key features of the SRDF/A version that has now been certified are:

1) Concurrent SRDF/A and SRDF/S from a single source – This key feature provides the ability to replicate a group of devices in Synchronous mode to one target site and in asynchronous mode to a tertiary site at an extended distance. The same source device can be replicated synchronously using SRDF/S mode down one link and asynchronously using SRDF/A down another.

This capability provides a limited multi-site protection capability. In the event of a primary site failure, there would be no data loss since it is synchronously replicated to a local/regional recovery site. In the event of a regional disruption, data has been simultaneously replicated to a recovery site much further away, with minimal potential data loss.

2) Support for multiple SRDF/A Groups – Multiple (up to 64 depending on configuration) SRDF/A groups per DMX array.

Also supported (in non-concurrent configurations) is SRDF Multi-Session Consistency (SRDF/MS), which provides consistency protection for an asynchronous-mode consistency group by performing SRDF/A cycle-switching and cache recovery operations across all SRDF/A sessions in that group.

3) Availability – SRDF/A provides a consistent, restartable image at the target site at all times and enables fast and efficient incremental failback and re-synchronization.

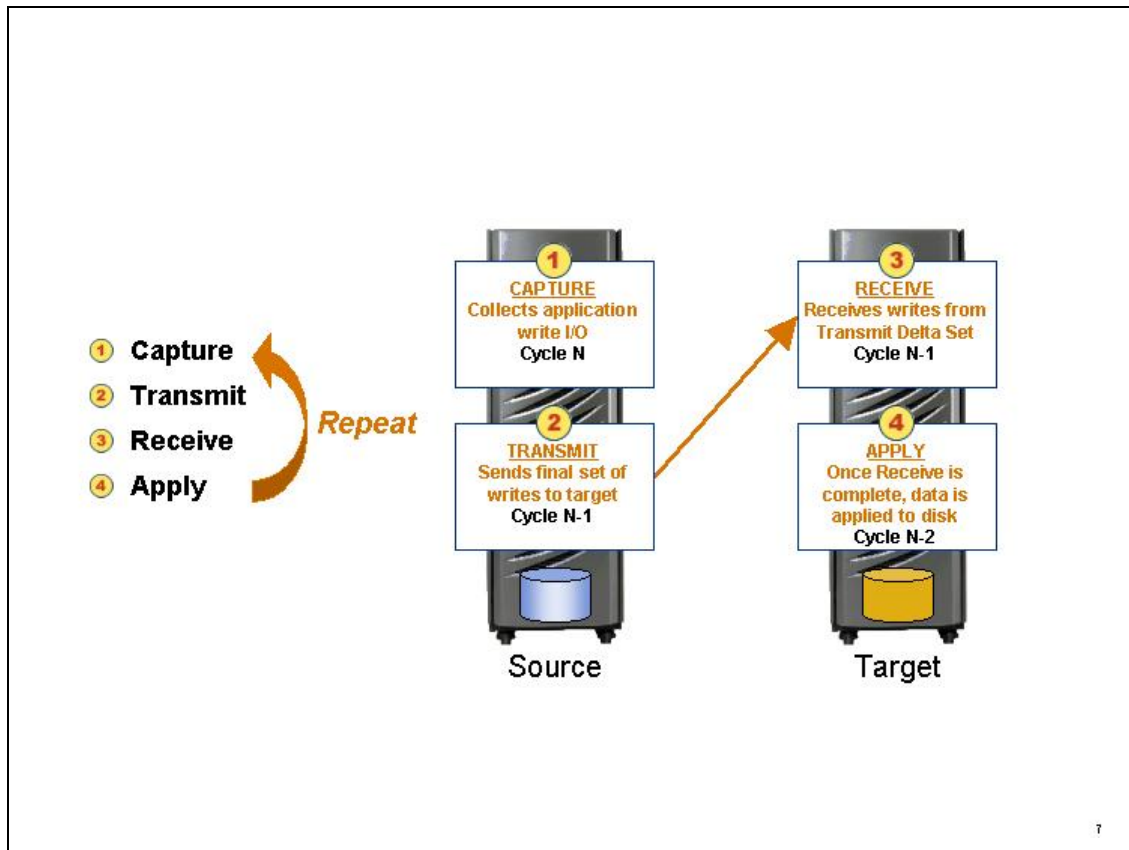
4) Performance – Generally negligible host application (dependent on configuration) impact regardless of distance between sites (if solution properly sized)

5) Bandwidth Efficiency – One difference between SRDF/A replication and other asynchronous methods of replication is that less data is transferred, and the data is handled fewer times. If the same data is updated multiple times in the same cycle (for example, the same track is written to ten times in a given time period), that data is sent across the RDF link only once. This is known as write-folding. This performance benefit can result in more efficient link utilization.

6) Familiarity – The SRDF/A product uses a toolset that users of synchronous SRDF will already be familiar with, and for which we already have developed security controls. This is a major benefit from an operational perspective.

2.1.2 How SRDF/A works

The following brief overview is intended to provide sufficient knowledge of the process to aid with the understanding of the results of this evaluation. It is adapted from EMC's White Paper on SRDF/A.



DMX systems implement asynchronous host write from the source to the target using predetermined timed Delta Sets. Each Delta set contains groups of I/Os for processing, which are managed for consistency by the Enginuity operating environment. Using Delta Sets, SRDF/A transfers sets of data, one Delta Set at a time, between the R1 and the R2. If the same data block is written to more than once within a Capture Delta Set on the R1 side, SRDF/A sends the update over the link only once.

Dependent write consistency is achieved through the processing of four ordered SRDF/A Delta Sets between the source (R1) and the target (R2), as shown in the above diagram. Dependent write consistency ensures that all writes to the R2 are processed in the sequential Delta Sets to maintain a consistent copy of data between R1 and R2.

When the first SRDF/A Capture Delta Set is active, it collects and new writes on the R1, overwriting an previously written data blocks intended for the data transfer over the link. The Delta Set is active for a predetermined minimum amount of time, which can be configured on the DMX (default is 30 seconds). After this minimum time has been reached, the Capture Delta Set data become the Transmit Delta Set and begins transferring the data over the link into the R2. A new Capture Delta Set begins collecting the new writes in global memory.

On the R2 side, the Receive Delta Set collects all the data from the Transmit Delta Set. Once the Delta Set has been received in its entirety, it is promoted to an Apply Delta Set and committed to disk (R2). This ensures a consistent, re-startable R2 copy with each application of an Apply Delta Set.

One Delta Set is dependent upon the other for achieving write consistency. No Delta Set can begin destaging to disk until the prior one has completed. All data is transferred at the block level.

2.1.3 Role of SRDF/A in Disaster Recovery

It must be stressed that SRDF/A will only give a business an I/O consistent copy of their data set at the DR site for a particular server, application or set of applications (depending at what level consistency is enforced).

In the case of a production failure that only impacts a server and not the storage infrastructure (e.g. a server hardware failure), there will be no data loss.

In the case where the storage infrastructure or indeed the whole production site fails, the DR copy will be from a point-in-time somewhere between 30-60 seconds (for the default cycle time) prior to the production failure and data loss will occur.

Note that in this case:

- the storage infrastructure will not be able to define exactly how much data was lost
- the storage infrastructure does NOT provide a mechanism for dealing with the data loss. This is the responsibility of the application.
- it is the application owners that needed to deal with issues relating to recovery using different point-in-time data set for applications that were not bound together from a consistency perspective.

2.2 SRDF Terminology

Symmetrix – Product name for EMC's High-end Storage Arrays.

DMX – Product name for latest generation of Symmetrix

Enginuity – Operating system run on Symmetrix

SRDF – The family name for EMC's remote storage replication solutions for Symmetrix

SRDF/S - High-performance, host-independent, **real-time synchronous** remote replication from one Symmetrix to one or more Symmetrix systems.

SRDF/A - High-performance **extended distance** asynchronous replication using a Delta Set architecture for reduced bandwidth requirements and no host performance impact.

Dynamic RDF – Version of SRDF that the end user can configure and manage

Concurrent SRDF – Simultaneous SRDF replication from a single R1 device to multiple R2 devices.

Delta Set - Data captured for a limited period (default of 30 seconds).

Write-folding - If the same data is updated multiple times in the same Delta Set, that data is sent across the RDF link only once.

R1 - Source device in an SRDF replication set up

R2 - Target device in an SRDF replication set up

Device Group – A collection of storage devices that are managed together as a group, for example, for failover purposes.

Composite Group – A grouping of device groups. Typically used for consistency operations.

Multi-Session Consistency (MSC) - the ability to provide SRDF/A replication of multiple SRDF groups with I/O consistency across all the groups

Business Continuance Volume (BCV) - A point-in-time copy of a data volume

Clone – EMC's new replacement technology for BCVs. Clones have less restrictions than BCVs when used with SRDF.

Latency (round trip) - The amount of time it takes a packet to travel from source to destination and back

Subset failover – failing over a subset of disks within a synchronous RDF group in a concurrent SRDF configuration

2.2.1 Consistency

It is necessary to clarify what is meant by consistency in this document as the word has differing meanings depending on its context.

One of the advertised benefits of SRDF/A is that it maintains an I/O consistent image at the target (R2) site at all times. In practical terms, this means that a database can always be restarted from this image.

Device group consistency

Device group consistency refers to devices acting in unison to maintain the integrity of a database distributed across multiple LUNs.

Temporarily disabling consistency for a device group is required for certain SRDF control operations. Disabling consistency protection does not make the R2 data inconsistent. This action simply causes the DMX to tolerate disk failures.

If consistency is enabled for a device group and both mirrors of a protected Symmetrix device fail (a very unlikely event), then replication from the other LUNs, which are part of the same group will be stopped to ensure IO consistency of the dataset at the target site.

If consistency is disabled, and both mirrors fail, then there will be loss of consistency (in both synchronous and asynchronous modes), as other LUNs that are still functioning will continue to replicate.

Hence, the default consistency should be enabled to provide an extra level of protection in the event of a double disk failure. However, there is very little risk involved with temporarily disabling consistency for short periods to issue control instructions.

Multi-Session Consistency

Multi-Session Consistency offers the ability to provide SRDF/A replication of multiple RDF groups with I/O consistency across all the groups, whilst still retaining the ability to failover individual RDF groups to deal with, for example, application server failures, without impacting the replication of other applications.

Consistency here means that the devices act in unison to maintain the integrity of a database distributed across multiple RDF groups within a single Symmetrix, or indeed across multiple Symmetrix. If a source R1 device in the consistency group cannot propagate data to its corresponding target R2 device, SRDF/A MSC suspends data propagation from all R1 devices in the consistency group, halting all data flow to the R2 targets.

3 Applicable Environments

This section describes the main scenarios in which SRDF/A is a suitable replication product. It also describes the key related concept of Recovery Point Objective..

3.1 Recovery Point Objective

Recovery Point Objective (RPO) is the point in time to which data must be restored to successfully resume processing, often thought of as time between the point in time when last backup was made and when the outage occurred.

Synchronous SRDF can offer an RPO of zero. However, synchronous replication is not suitable for long-distance replication (typically defined as > ~80km) as the network latency between the source and target sites can result in unacceptable IO performance on the source.

Asynchronous SRDF does not require each IO to be acknowledged by the target Symmetrix, before the next can be sent and hence does not suffer in the same way from network latency. However, as the process is asynchronous, there is the potential for data loss and hence a non-zero RPO.

During the Citigroup evaluation of SRDF/A, RPOs of between 60 and 90 seconds were achievable for the different test configurations during normal operations. These configurations included simulation of all the different network latencies that are expected to exist between source and target sites in the different regions.

In production, achieving this RPO would be dependent of there being sufficient effective bandwidth for the actual workload. The underlying IP networks over which replication runs would also need to be sufficiently stable and of high enough quality to avoid SRDF/A dropping out due to connectivity issues. EMC need to be formally engaged to determine bandwidth requirements for specific implementations of SRDF/A.

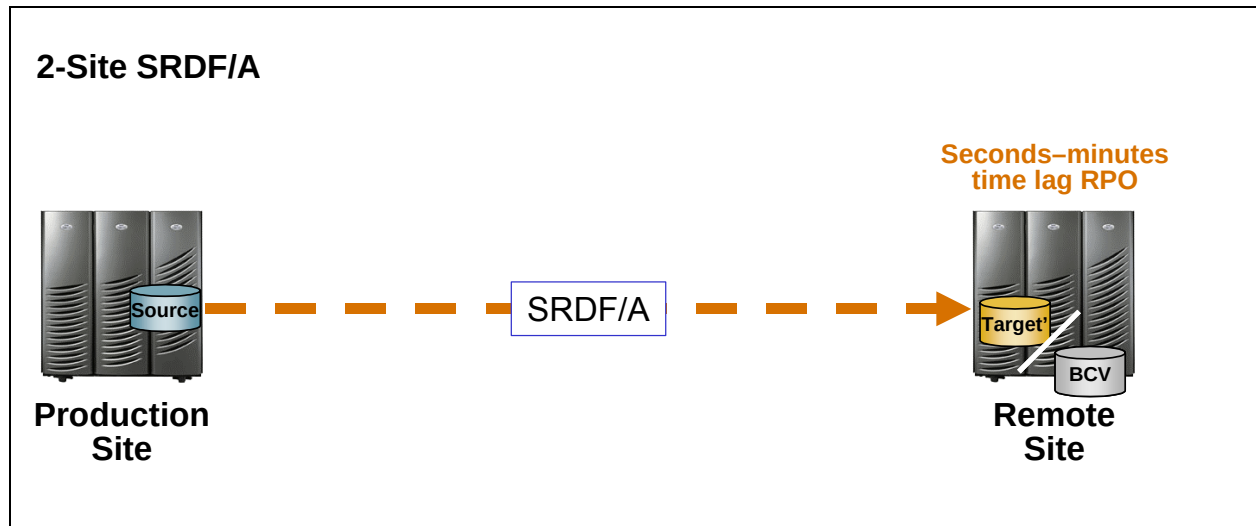
3.2 Example SRDF/A configurations

Citigroup is supporting use of SRDF/A under the following circumstances:

- **2-site SRDF/A configuration:** Replication between two sites separated by a distance too great for the use of synchronous replication i.e. for out of region disaster recovery. Note that this will produce a non-zero RPO in the event of a production site failure.
- **Concurrent (3-site) SRDF configuration:** In this scenario, synchronous SRDF will be used to a local COB site and asynchronous SRDF to a remote site. The local site will provide zero-RPO COB to deal with service issues at the production sites. The remote site would provide protection against a regional disaster that rendered both the production and local COB sites inoperative.

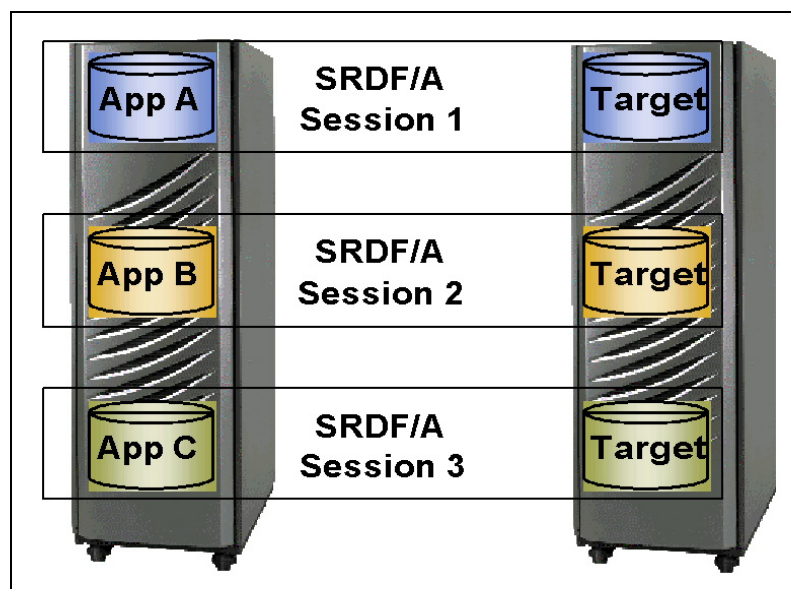
3.2.1 2-site SRDF/A configuration

The following diagram illustrates the two-site SRDF/A solution.



Data is replicated asynchronously from Production site to the Remote site with negligible IO response impact, despite the long distance between sites. An RPO measured in minutes or seconds can be achieved in the event of the loss of the production site. It is good practice to have BCVs available at the remote site to allow point-in-time copies to be made during resynchronizations.

Multiple (up to 64) SRDF/A replication sessions can be configured, each of which can be controlled completely independently of each other. This allows individual servers/applications to be failed over in the event of a server/application failure, without impacting the replication of other healthy server/applications. This is referred to as **multi-session SRDF/A**.

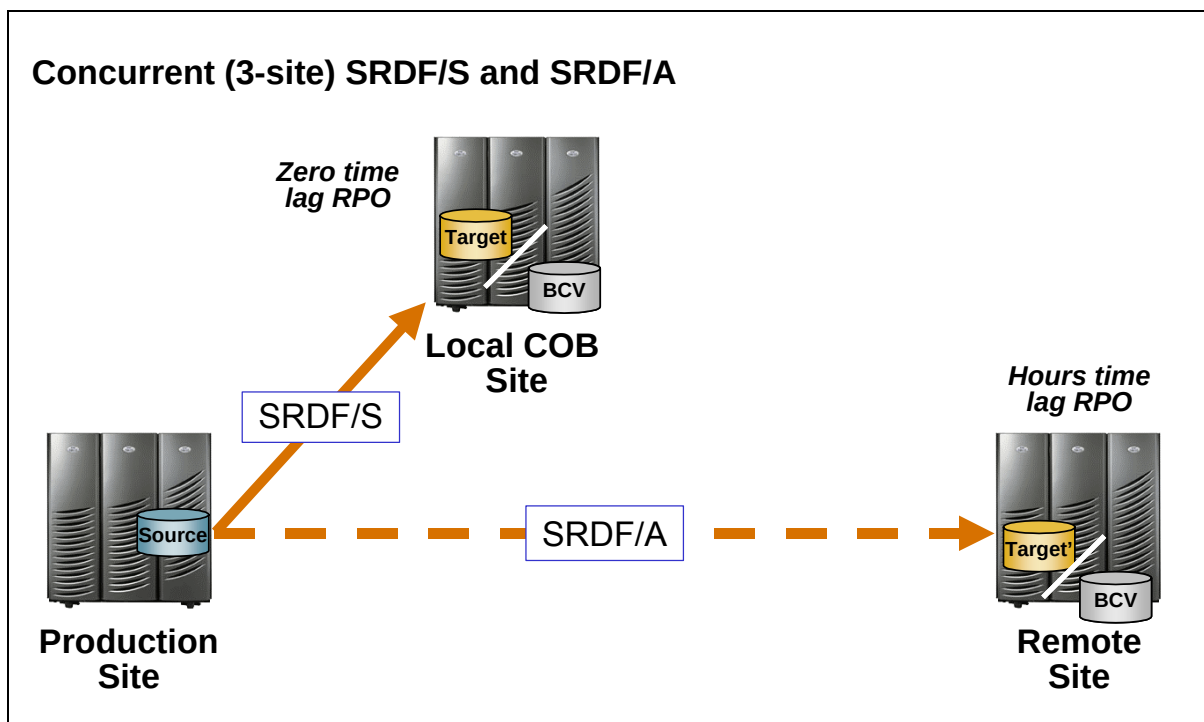


In the event of server/application failure, where the rest of the storage and network infrastructure is unaffected, all the application data written at the production site prior to the failure will be replicated to the remote site. Hence, it is possible to achieve an RPO of zero in the event of server/application failure (only) with a multi-session 2-site SRDF/A.

It is also possible to bind multiple SRDF/A replication groups (i.e. multiple applications) together into a consistency group. This ensures that should the replication of one application fail (due to storage hardware or network failure), then replication of the data of the other applications within the same consistency group will cease. This ensures that the data at the target site for all the applications protected by the consistency group is from the same point in time. This is known as **Multi-Session Consistency**.

3.2.2 Concurrent (3-site) SRDF configuration

The following diagram illustrates the Concurrent (or 3-site) SRDF solution.



Data is replicated synchronously from Production site to Local COB site. An RPO on zero can be achieved in the event of the loss of the production site.

Simultaneously, the same data is replicated asynchronously from Production site to the Remote site with negligible I/O response impact from the perspective of the server, despite the long distance between sites. Failover to the remote site would only occur if both the Production and Local COB site were rendered inoperative (regional disaster). An RPO measured in minutes or seconds can be achieved in the event of a regional disaster.

Multiple (up to 31) SRDF/A replication groups can be configured, each of which can be controlled completely independently of each other. Each SRDF/A replication group would also have to have a separate SRDF/S replication group associated with it. This is necessary to ensure that individual applications can be failed over to the Local COB site (which will halt

replication to the Remote site for those applications) without impacting the replication to the Remote site of other applications.

Note that Multi-Session Consistency (MSC) is not supported by Citigroup in a concurrent SRDF configuration, except in an SRDF/Star configuration. **Hence, if a concurrent SRDF configuration is deployed without SRDF/Star, there will not be guaranteed consistency between different RDF groups on the synchronous leg.**

In summary:

1. Multi-session Concurrent SRDF (with one SRDF/S leg and a second SRDF/A leg) is supported
2. Concurrent SRDF implementations should use two separate replication groups for each application, one for each leg. A maximum of 31 separate replication groups can be set up per leg.
3. Multi-Session Consistency is available for concurrent SRDF via SRDF/Star.

3.2.3 SRDF/A over long latency links

In principle, SRDF/A can run over an unlimited distance. In practice, there are in fact some practical limitations that need to be taken into account before deciding to go ahead with an implementation.

1. Large latencies can dramatically reduce the effective throughput on a network link. For example, the throughput on a nominal 100MB/sec network link with a 4K writes IO workload (typical for SRDF/A) has been measured to reduce dramatically as latency increases.

GDSE has developed Best Practice to optimise WAN throughput for SRDF/A replication using the McData IPS product. See references section for further information.

2. SRDF/A becomes sensitive to packet loss with very high latencies (e.g. 250ms). The consequence are:
 - a. an increased probability that SRDF/A will abort (i.e. replication will cease) when IP packet loss occurs between the source and target sites.
 - b. a significant reduction in throughput when packet loss occurs on the IP network between the source and target sites.

GDSE has developed Best Practice to minimize this issue when using the McData IPS product. See references section for further information.

3. Cycle time increases with latency for replication sessions under Multi-Session Consistency control. This becomes significant over 40ms.
4. Much longer runtime for SRDF control and query operations. This becomes significant for latencies over 40ms.

4 Preparation for an SRDF/A implementation

4.1 Introduction

An SRDF/A implementation requires considerable planning. In this section, the main planning considerations and pre-requisite work is discussed. EMC needed to be closely involved in the planning, as it is necessary for them to certify each and every implementation.

An SRDF/A implementation is very application sensitive and requires detailed analysis of the I/O profile of the target applications prior to deployment to ensure that the correct network bandwidth and storage hardware is made available for the implementation.

EMC are currently unable to define a utility model that allows one to pre-build an “SRDF/A-enabled” tier of storage onto which applications with unmeasured I/O profiles could then be deployed. Citigroup Storage Engineering are attempting to work with EMC to develop such a “Utility SRDF/A” model, which would allow new applications to be deployed onto a pre-built SRDF/A storage tier. However, at the present time SRDF/A can only be deployed for identified existing applications with measurable I/O profiles.

The main steps in preparing for an SRDF/A implementation for existing applications are as follows. Additional steps required for an MSC or concurrent SRDF implementation are covered in later sections:

1. Collect basic requirements information for target applications to determine SRDF/A configuration and capacity requirements, licensing requirements, etc.
2. Engage with EMC to obtain Solution Qualification for the proposed solution. This process can take up to 2 months and should be started immediately an SRDF/A solution is proposed.
3. Order additional network links or bandwidth (if necessary), other network infrastructure and other storage licenses and hardware consistent with the Solution Qualification.
4. Install DMX storage, SAN management servers, McData IPS units. Connect IPS units to network.
5. Configure DMX ready for SRDF/A usage with 5x71 microcode. Install latest certified version of Solutions Enabler 6.x on management servers. Configure McData IPS units.
6. Map RDF groups to RAs. Map applications to RDF groups. Map applications to devices and hence devices to RDF groups.
7. Create required RDF groups and populate with application devices.

This section describes how to plan an SRDF/A solution from scratch. Some advice on how to convert an existing 2-site SRDF/A solution into a 2-Site SRDF/A solution can be found in Appendix C, for the scenario where the production site remains fixed but the COB/DR site swings to a new highly remote location.

4.2 Collection of Basic Application Requirements

Before any design work can be undertaken for a specific SRDF/A implementation, it is vital to collect some basic information on the requirements of the target applications to understand the scope of the project ready for initial discussions with EMC.

A suggested list of initial questions, and the reasoning for asking each can be found in Appendix A.

EMC will also require much more detailed information as part of the Solutions Qualification process.

One of the most crucial decisions that must be made is which applications require point-in-time consistency with each other at the target site. This critical decision determines:

- How many RDF groups will be needed
- If multi-session consistency will be needed

Hence, these decisions must be made very early in the planning process.

Enforcing consistency between applications at the storage subsystem level does have negative points and hence it is essential that is undertaken only where such consistency enforcement is genuinely required and where the application owners have been informed of the “binding” together of these applications, and the negative consequences (see Section 6).

4.2.1 Determining the appropriate SRDF configuration

The answers from the above questionnaire can be used to determine basic design parameters for the specific SRDF/A implementation.

The decision on whether to go for a two or a three site solution depends on consideration of the detailed COB/DR requirements for the application. There are two strategic directions for long-distance SRDF within Citigroup for Open Systems:

- 2-site SRDF/A
- 3-site SRDF/Star

3-Site Concurrent SRDF (without Star) is also supported, although offers little advantage over 2-site SRDF/A in the context of the new Data Centre Strategy to use long-distance sites as the primary COB/DR sites, rather than local sites. 3-Site Concurrent SRDF was certified before SRDF/Star was available, and primarily for when out-of-region recovery is needed but the primary COB/DR site is still in region (tens of kms from production).

Concurrent SRDF has most of the disadvantages of SRDF/Star (higher cost, I/O response, complexity but with none of the advantages (potential for zero data loss at long-distance site and on-going COB after loss of production site), and hence is not the recommended solution going forward in the context of the new Data Centre Strategy.

	2-site SRDF/A	3-site Concurrent SRDF	3-site SRDF/Star
Certified by Global Engineering?	Yes (Strategic solution)	Yes (Non-strategic)	Yes (Strategic, but for selected applications only)
Typical separation of production and COB sites	Thousands of km	Thousands of km	Thousands of km
Protects against total loss of all data availability after regional disaster?	Yes	Yes	Yes
Typical amount of data loss after server failure	None	None	None
Can support individual data server level failover (where > 1 data server per application)?	In practice, No	In practice, No	No
Can offer application level failover	Yes	Yes	Yes
Typical amount of data loss after production site failure	30-60 sec	30-60 sec	None
Guarantees no data loss in a rolling disaster scenario?	No	No	No
Typical amount of data loss at remote site after wide-scale disaster at production location	30-60 sec	30-60 sec	30-60 sec
Number of sites requiring storage	2	3	3
Ongoing replication to a remote site after loss of original production site	No	No	Yes
Application I/O performance not limited by synchronous replication	Yes	No	No

4.2.1.1 Similarities between SRDF/A and SRDF/Star

- Both offer zero data loss after a server/application level failure
- ***Neither can realistically offer server level failover*** (i.e. single data server within an application comprised of multiple data servers with SRDF replication)
- Both can offer application level failover
- Neither guarantees zero data loss in a Rolling Disaster scenario
- Neither guarantees zero data loss in a Regional disaster scenario
- Both protect against total data loss in the event of a regional disaster
- Long distance network costs could be similar if engineered carefully

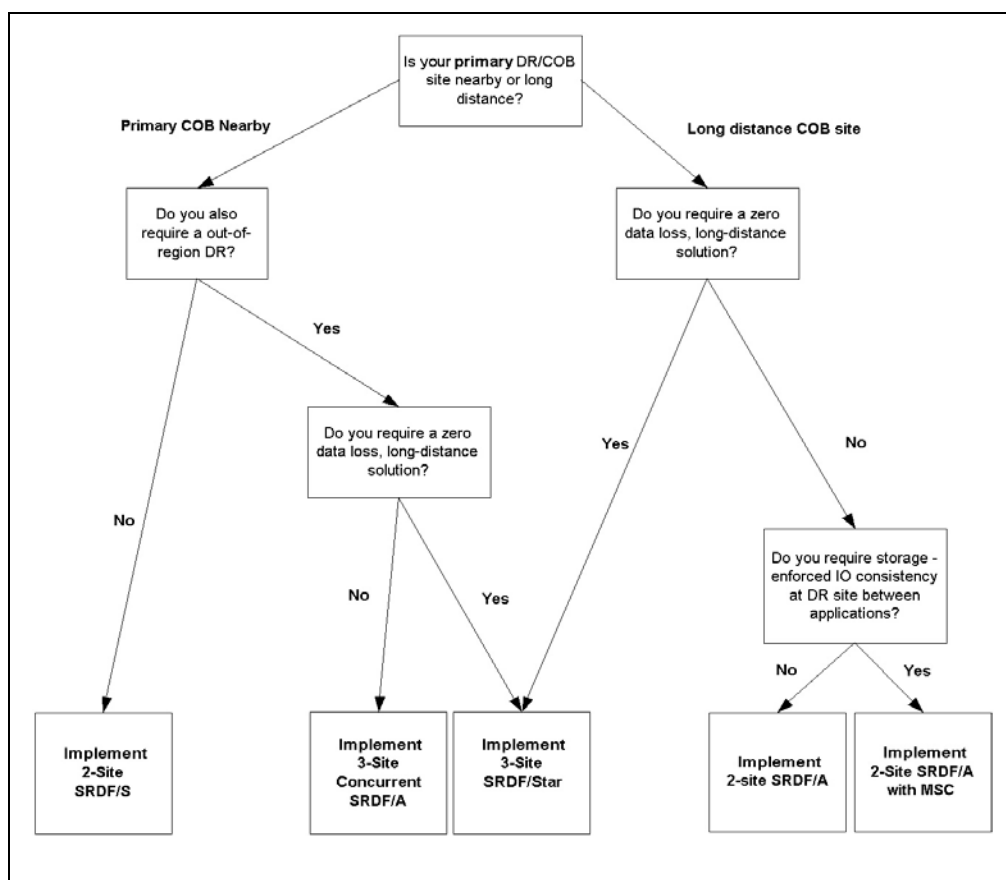
4.2.1.2 Differences between 2-site SRDF/A and 3-site SRDF/Star

- Only STAR provides zero data loss (up the start of a rolling disaster) in the event of loss of the production site
- Only STAR offers ongoing data replication after the loss of Site A
- 2-site SRDF/A is a higher I/O performance solution than SRDF/Star as it does not have a “slow” synchronous replication component.
- STAR requires a third bunker site
- STAR has approximately 50% extra storage costs.
- STAR may have higher networking cost compared with 2-site SRDF/A, depending on how the network component is engineered.
- Failovers with STAR is a much more complex and slower process than with 2-site SRDF/A
- STAR is much less scalable and flexible than 2-site SRDF/A

4.2.1.3 Selecting which SRDF solution is suitable for an application

The following flow diagram attempts to summarize the simplified process for determining which type of configuration needs to be used to meet a protection requirement. The primary decision points are:

- The location of the primary COB /DR site – is it nearby or remote?
- If local, is there also a requirement for an additional long distance DR site to cover for regional disasters
- If a zero data loss solution is needed for a long distance implementation



Other factors such as those covered earlier in Section 4.2.1 will also have a bearing on the decision.

SRDF/Star is outside of the scope of the present document and is covered in a separate Solutions Guide (see References section). Hence, it is not discussed in detail any further here. The focus of this document is on 2-Site SRDF/A, 2-Site SRDF/MS and 3-site (non-Star) Concurrent SRDF.

4.2.2 Scalability limitations

There are limitations on how many independent applications a single DMX can support.

Here we define independent as the ability to failover a server/application over to the COB/DR site without affecting replication of other servers/applications, or have that server/application's replication impacted by the failover of other servers/applications

For 2-site standard multi-session SRDF/A, the limit is 63. This restriction is due to the number of dynamic RDF groups that a DMX can support.

Mixing 2-site standard multi-session SRDF/A and 2-site MSC SRDF/A with Multi-session Consistency (MSC) control is supported.

For 3-site Concurrent SRDF implementations, the limit on the number of independent applications is 31. This is half the limit of the 2-site configuration, as each application needs two RDF groups; one for the sync leg and one for the async leg.

Limitis for SRDF/Star are discussed in the SRDF/Star Solutions Guide.

2-site and 3-site Concurrent SRDF can, in principle, both be running from the same source DMX.

4.2.3 Determining the number of RDF groups that are required

4.2.3.1 2-site SRDF/A

The key point to remember when determining how many RDF groups will be needed is that with 2-site SRDF/A, failover is an all-or-nothing process. It is not possible to failover a subset of the disks within the SRDF/A RDF group, for example, for a single server. All servers associated with the RDF group will need to be failed over together.

- For a 2-site solution, each application will typically have its own RDF group. This permits failover/failback of that application independently of any other applications.
- For a 2-site solution where certain applications require storage-infrastructure-enforced data consistency between them, but also require the ability to independently failover applications, then each application should have its own RDF group, but then replication of these groups must be under Multi-Session Consistency (MSC) replication control. Note that MSC requires the purchase of SRDF/Consistency Group Licenses.
- In certain circumstances, it may be necessary to failover a subset of disks that are part of an application. For example, an application may have many servers associated with it and it may not be acceptable to failover ALL servers associated with the application when a single server fails. In this case, subsets of the application servers will need to be put into different groups. If the data hosted by servers in different RDF groups needs to be consistent, then MSC will need to be used to enforce consistency across these groups. Note that MSC requires the purchase of SRDF/Consistency Group Licenses.

4.2.3.2 3-site Concurrent SRDF

3-site Concurrent SRDF configuration is much more complex as there will be interaction between the Synchronous and Asynchronous replication legs.

- For 3-site Concurrent SRDF, each application will typically have two RDF groups of its own. This permits failover/failback of that application down either the Synchronous or Asynchronous leg independently of any other applications. As with 2-site SRDF/A failover/failback down the SRDF/A leg is all-or-nothing for all devices within that leg.

3-site Concurrent SRDF is discussed in more detail in Section 7.

4.3 EMC SRDF/A Solution Qualification

4.3.1 Introduction

Every SRDF/A implementation requires qualification by EMC. The local EMC account team should be contacted as soon as an SRDF/A implementation is proposed, and instructed to request a Solutions Qualifier (SQ) for Business Continuance.

This process will involve data collection for the target applications and subsequent analysis of the data. Hence, all applications in scope will need to be confirmed before this work can take place. This analysis must be completed **before any orders are placed**, as it will be used to determine network bandwidth requirements, DXM hardware configuration and other hardware requirements.

The whole process can take up to 2 months, although with focussed effort, the duration could be reduced to 3-4 weeks. The local account team will be responsible for the completion of the SQ, but will need significant input from the local Citigroup staff. The SQ form is very detailed (hundreds of questions) and also requires the collection of current performance data.

At this time, Citigroup is only certifying SRDF/A for use with existing application with I/O profiles that can be measured. Citigroup is not certifying the installation of SRDF/A infrastructure without clearly defined target applications and hence sizing requirements.

4.3.2 Overview of Solutions Qualification Process

EMC have developed tools to model any proposed SRDF/A deployment. The customer supplies information including (but not limited to):

- Network equipment to be used (e.g. McData IPS 2640)
- Maximum bandwidth per link
- Number of links
- Estimated compress for data (or use default 2:1)
- Distance between sites
- Estimate of network protocol overhead (Default from EMC lab testing for IP is 20%)
- DMX details (microcode levels, hardware type, number of RAs, connection type to DMX e.g. FC)
- Bandwidth available for this DMX (e.g. 10%)
- Symmetrix Devices used by applications in scope for SRDF/A replication.

EMC also require performance data from the DMX to be replicated. It is recommended that all Citigroup sites request the collection of STP data using latest version of STP.

STP data must be 10 min cycles run for 3-5 days **as a minimum**. The data must cover periods of peak usage (e.g. month end). If the target application is not on Symmetrix today, then host performance data (e.g. iostat) will need to be collected.

EMC will provide two reports (per DMX). The initial report will indicate cycle time that can be achieved, bandwidth requirements and cache requirements.

The second, longer performance report will profile all IOs for the target application. It will include, for example, an analysis of periods when the desired cycle time (and hence RPO) cannot be achieved with the proposed bandwidth due to peaks in traffic.

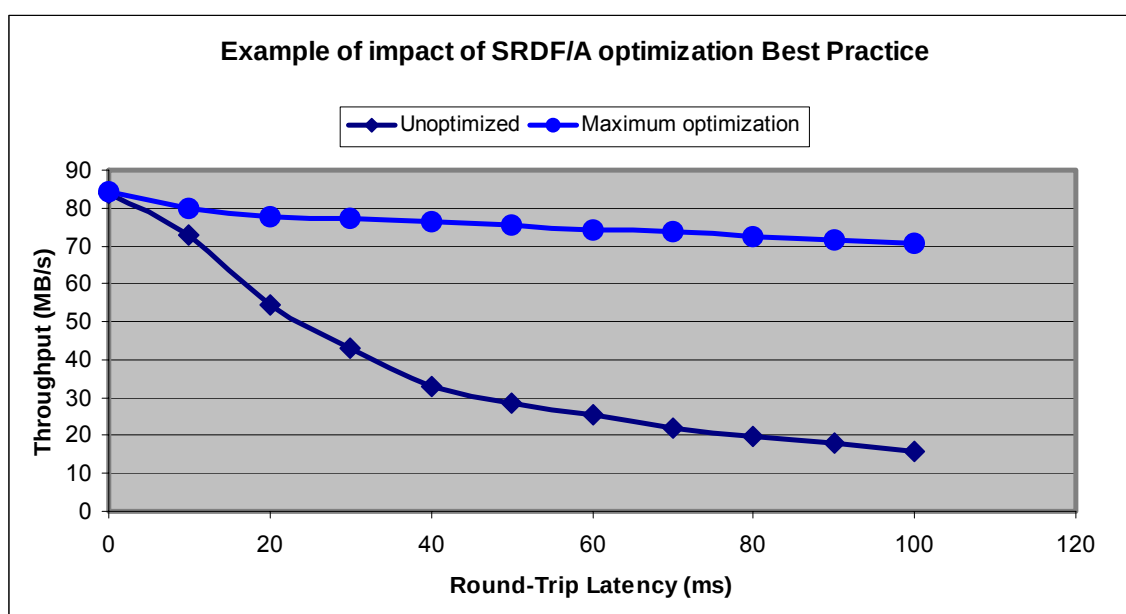
If the DMX you want to SRDF/A replicate is not yet in production and is in fact the target for a migration process from multiple older Symmetrix, then EMC can collect data from the Symmetrix units and use symmerge to estimate what the workload would look like on the new DMX.

4.4 Background information on Network Bandwidth Sizing

The sizing of the network links in an SRDF/A implementation is absolutely critical. The main inputs to the sizing of the network bandwidth needed for SRDF/A are:

- **1) Write IO rate.** This is the main driver; the more data you write during each fixed-length cycle, the more data needs to be transferred across during the next fixed-length cycle and hence more bandwidth is required. At this time, it be necessary for EMC to determine your bandwidth requirements for each implementation following an analysis of the I/O write profile of the applications to be replicated with SRDF/A.
- **2) Latency.** High latency dramatically reduces the effective bandwidth of a network link. This in turn will dictate the maximum write I/O load that can be supported. The type of device used for FC-to-IP conversion will also have an input on the bandwidth efficiency.

The following graphs are taken from the GDSE document “WAN Throughput Optimization for SRDF/A Solutions Guide” illustrate the point clearly.



Citigroup had adopted the McData solution for SRDF/A and developed Best Practice of combating latency. The lower line shows the unoptimized throughput, and the upper line optimized throughput achieved by following the Best Practice in the Solutions Guide.

- **3) Write-folding.** Greater write folding reduces the amount of RDF traffic generated for a given I/O write rate. Typical write-folding write traffic reductions are 20% for 30 second cycles. This might rise to 25-30% for 60 second cycles.

It must be stressed that changing the cycle time has only a minimal impact of bandwidth requirements. If the cycle time is doubled, the amount of data that needs to be carried across during the next cycle is doubled (less any further write-folding benefits). Hence, the only benefit (in terms of bandwidth utilization) of changing the cycle time is potentially greater write-folding for longer cycle times.

Furthermore, it must be stressed that insufficient bandwidth does not simply lead to longer and longer cycle times. With SRDF/A there is also the issue of cache utilization. If the IO write rate exceeds the bandwidth available for an extended period, cache utilization will grow and grow until cache is exhausted and SRDF/A aborts.

Lastly, the amount of time that a business is prepared to have their desired RPO following an outage may dictate bandwidth requirement. To be specific, after a lengthy outage, outstanding data on the R1 side will need to be copied to the R2 side. During this time the R2 will be inconsistent. Whilst a BCV copy of the R2 would be taken prior to the re-sync, this would be from a fixed point-in-time (and hence aging) and thus the DR site would be drifting further and further behind production. Greater bandwidth would speed up the re-sync process.

Hence, sizing the network bandwidth required for an SRDF/A installation is a complex and critical task.

4.4.1 Comment on network quality

Extensive testing has shown that SRDF/A is very stable when run over a robust underlying network. However, SRDF/A will abort if their network is not of sufficient quality. Testing also found that SRDF/A becomes much more susceptible to imperfect network issues such as packet drop as latency increases.

Hence, it is absolutely essential that SRDF/A be deployed over a network with:

- Dedicated bandwidth.
- Low packet drop (%). This will usually be 0% unless the link is heavily (95%) utilised.
- No Packet re-ordering. It is essential to ensure that load sharing does not occur over links with different latencies.
- Duplicate packets. Usually only occurs if there is a hardware issue.

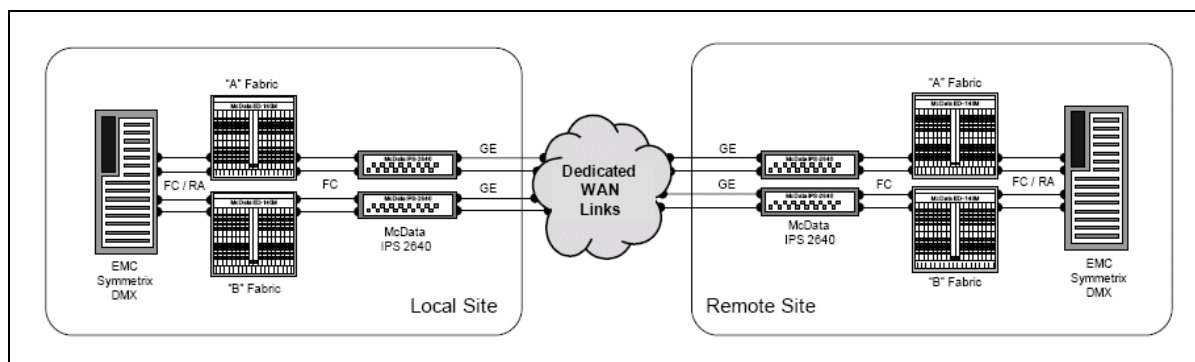
If the network is not robust, replication will abort by design, as the two DMX will not be able to communicate with each other within the operational parameters of SRDF/A.

The GDSE document “**WAN Throughput Optimization for SRDF/A Solutions Guide**” also gives advice on minimizing and combating packet drop.

4.5 Network Hardware

4.5.1 Overview of Network equipment configuration

The following diagram shows an example 2-Site SRDF/A network configuration.



McData IPS SAN routers should be used for the FC to IP conversion. The above example shows the DMX units connected to two fabrics, which in turn are connected to the McData IPS units. The DMX can also be directly connected to the IPS units. Guidance on how to plan an IPS implementation for SRDF/A usage can be found in the GDSE document **“McData IPS SAN Router Solutions Guide”** (see references in Section 1) and is not discussed further here.

However, it must be noted that this is a very significant piece of work and the above document should be reviewed as soon as an SRDF/A implementation is considered.

The above example shows a source DMX to the left. 4 RAs are logically connected to 2 McData IPS 2640 SAN router units via SAN fabrics. The IPS units convert SRDF traffic running over Fibre-Channel into IP traffic. These in turn are connected to a (redundant) pair of Cisco switches. The SRDF IP traffic flows across the corporate network to the remote site, where it is converted back into FC and into the target DMX.

The above example is for a 2-site SRDF/A configuration. Note, however, that an identical configuration would be needed for the SRDF/A leg of a 3-Site concurrent SRDF implementation. The number of IPS units required would depend on the number of RAs to be used on the SRDF/A leg, which in turn depends on the number of RDF groups to be supported. Determination of the number of RAs required is discussed in a later sub-section. Advice on sizing the IPS infrastructure is also covered the **“McData IPS SAN Router Solutions Guide”** referenced above.

Each member of pair of RAs supporting a given set of RDF groups should be connected to a different IPS unit. For example, if RDF groups 3 through 8 were using RAs 8D and 9D, then 8D should be connected to one IPS and 9D to a different unit. Best practice on mapping RDF groups to RAs and RAs to IPS ports is given in the document **“WAN Throughput Optimization for SRDF/A Solutions Guide”**.

On the IP side, the network is broken up into separate VLANs (one on each Cisco switch) and each IPS switch has a connection into each Cisco switch and hence each VLAN. Hence, the entire configuration contains no single points of failure.

The links via the Corporate IP network that are used for SRDF/A replication:

1. Must be provided by your Networks team with Dedicated Bandwidth

2. Must be redundant (primary/secondary configuration)
3. Must set up so that under normal running, the links have a single fixed latency i.e. there is no load balancing across two different routes with differing latencies.

The data transfer must be via the Citigroup network, else it would need to be encrypted.

4.6 Management server requirements

4.6.1 Supported platforms

Control of SRDF/A via Solutions Enabler 6.x is only supported on Solaris. The management servers will be used to:

1. Configure and manage the SRDF/A environment
2. For controlling replication if Multi-Session Consistency (MSC) is implemented.

Management servers will be needed at the source site and both target sites (for failover)

4.6.2 Source site

There should be a minimum of 2 management servers at the source site. Only one is actually needed for SRDF control, but a second should be provided for redundancy.

If MSC is implemented, the management server service continuously controls replication and hence must be highly available. Implementation of high availability of the replication service is described in detail later in this document.

4.6.3 Target sites

There should be a minimum of 1 management server at each of the target sites to execute failovers should there be a failure at the production site.

4.7 Storage Hardware Considerations

One fundamental rule for SRDF/A deployments is that the systems should be “**balanced**” e.g. have the same disk type and RAID type at both ends. EMC have found that failure to do so can result in SRDF/A failing. A full description of the many issues with unbalanced systems is beyond the scope of this document, but the bottom line is that if the remote Symmetrix is not at least as fast as the source symmetrix, cache can fill up and SRDF/A will drop as the average write input exceeds the average write output to the remote target disks.

A full description of the issues with unbalanced systems can be found in the EMC document “**SRDF/A and SRDF/A Multi-Session Consistency – Unix and Windows Solutions Guide**”.

A balanced configuration is such that the design enables the ongoing de-staging of data to remote target disk at least as fast as it was accrued at the source. This is most easily achieved by having identical hardware and configuration at each end, and not using larger, slower disks with a different RAID configuration at the remote end to save money. For example, if you sue 73GB RAID1+0 at the production end, you should not deploy 146GB RAID-5 at the remote end.

Summary: EMC and GDSE highly recommend that you balance SRDF/A configurations. This means you have the same protection scheme, same number of drives, same speed, and same number of DAs on both the source volume and target volume. Failure to do so may result in SRDF/A dropping under heavy load, as it has at existing EMC customer sites where unbalanced configurations have been deployed.

4.7.1 Cache requirements

Additional cache is required on both the source and target DMX in an SRDF/A implementation to store the delta sets. EMC will determine the amount of additional cache that is needed as part of Solutions Qualification process described earlier as this is I/O workload dependent.

4.7.2 Additional RA requirements

EMC recommend that no more that 8 RDF groups are configured per RA port (as of Q4 2006 greater numbers can only be supported via RPQ). Hence, in a multi-session configuration SRDF/A implementation (both 2 or 3 site) additional Fibre Channel directors will usually need to be purchased compared with a traditional 2-site SRDF/S where all devices are usually in the same RDF group.

GDSE Symmetrix DMX Best Practice recommends that only port 0 be used on Fibre channel Directors. Hence, each additional Director is capable of providing 4 RA ports. Each RDF group should be spread across 2 ports (each on a separate director) for redundancy.

Hence, the formulae for determining the minimum number ports are required for SRDF RAs are as follows. It is assumed here that 6 Dynamic RDF groups will be spread across each pair of RA ports. This will be in addition to the underlying static RDF group that is put into the bin file by EMC (for a total maximum of 7 RDF groups per RA).

4.7.3 Additional note on RDF groups per RA limit

EMC SRDF performance expert (Dan Aharoni) has indicated that the impact of having multiple RDF groups on an RA is negligible for Synchronous SRDF. This point is being made for those planning concurrent SRDF implementations.

There is an impact for SRDF/A, and hence the recommendation for Concurrent SRDF of 8 RDF groups per RA today. EMC will extend this limit to 16 groups per RA via RPQ for 2-site (non-concurrent) SRDF/A. Check with your local implementation team on the current limits, as these may have changed since this document was written.

If you are planning to use 6 Application RDF groups per RA for SRDF/A before the patch is available, and are concerned about impact of the additional seventh underlying static RDF group (not used by any application) having a negative impact on performance, then it is acceptable to create separate static RDF group for each RA redundant pair and use that for SRDF/A replication. In this way, 6 application RDF groups can be hosted per RA with a total of only 6 RDF groups per RA.

The rest of this document, however, assumes that an underlying static group between two DMX is configured, and that application RDF groups are created dynamically in addition to that static group.

4.7.3.1 2-Site SRDF/A

Production Site

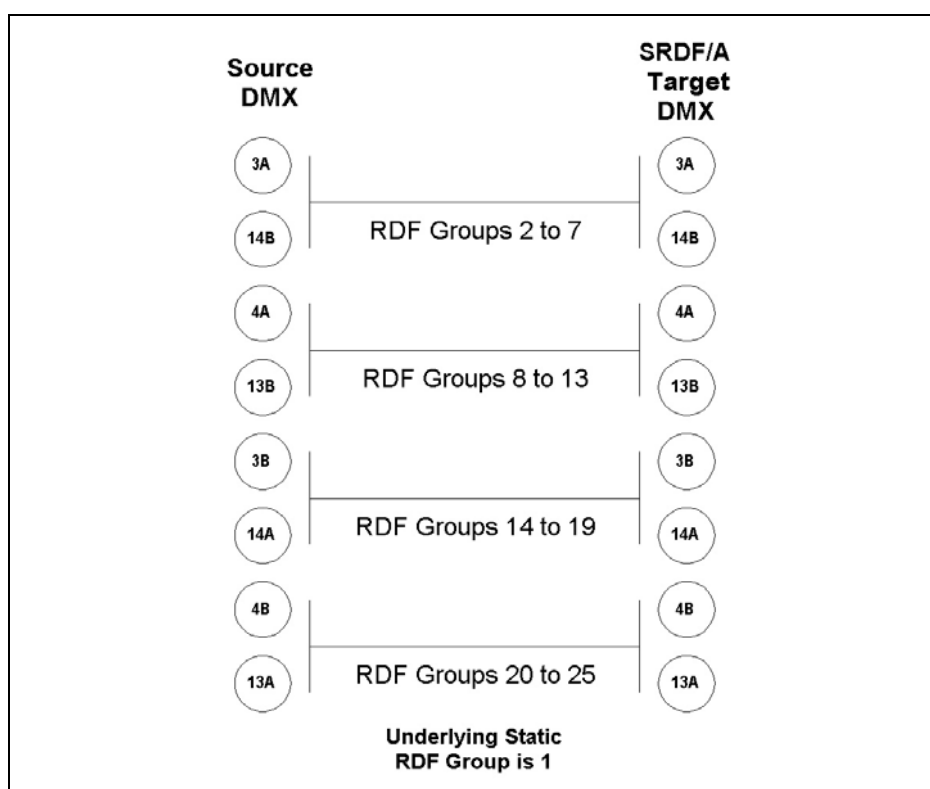
SRDF/A Source DMX RA ports needed = $\text{NumRDFG} / 6 * 2$

COB/DR site

SRDF/A Target DMX RA ports needed = $\text{NumRDFG} / 6 * 2$

where NumRDFG = Maximum number of RDF groups that will be implemented

For example, if you need to support 24 RDF groups, you will need 8 ports dedicated to SRDF/A on each of the source and target DMX. Each of the 4 pairs at each site would support 6 RDF groups.



The ports shown in the above diagram (e.g. 3A) are arbitrary and may differ in your implementation. The actual ports to use in your implementation should be agreed with EMC.

4.7.3.2 3-Site Concurrent SRDF

In this configuration, each RDF group for SRDF/A needs a counterpart for the synchronous leg too. Hence:

Production Site

SRDF/A Source RA DMX ports needed = $\text{NumRDFG} / 6 * 2$

SRDF/S Source RA DMX ports needed = $\text{NumRDFG} / 6 * 2$

Remote DR site

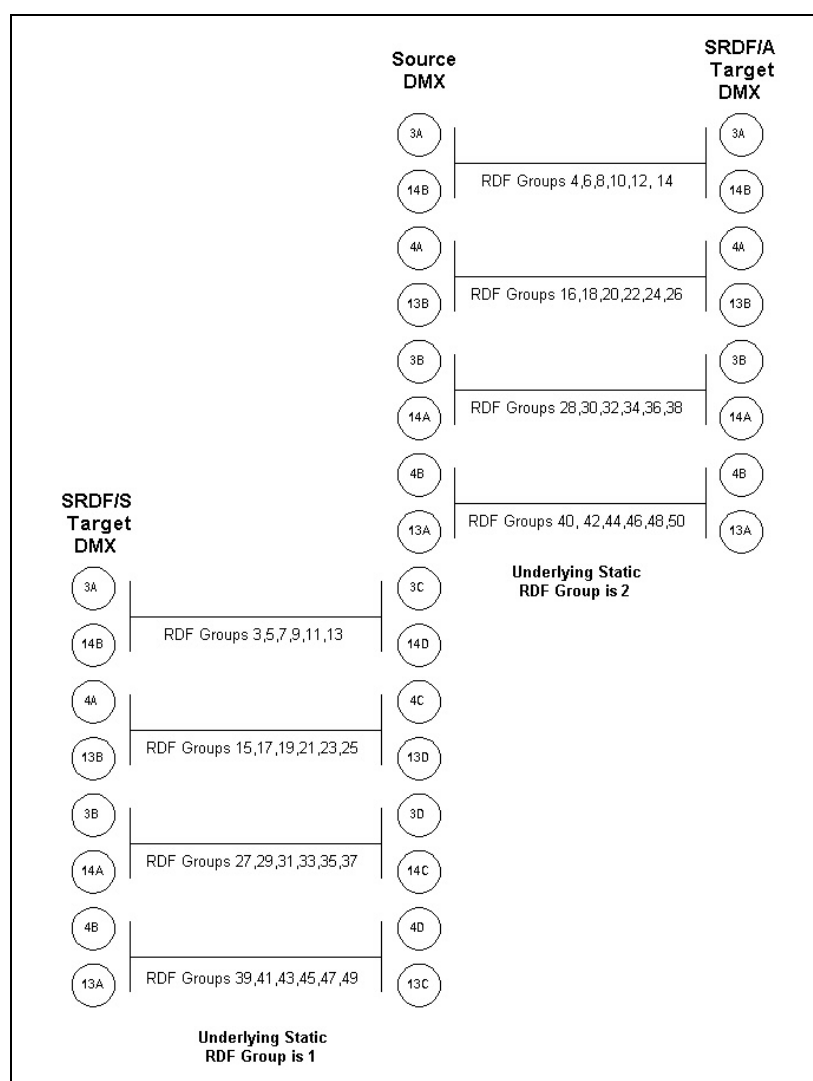
SRDF/A Target RA DMX ports needed = $\text{NumRDFG} / 6 * 2$

Local COB site

SRDF/S Target RA DMX ports needed = $\text{NumRDFG} / 6 * 2$

For example, if you want to support 24 Concurrent SRDF replication sessions you will need:

- 16 ports for SRDF on the source DMX (4 pairs for SRDF/S and 4 pairs for SRDF/A)
- 8 ports for SRDF/A on the Remote DR site DMX
- 8 ports for SRDF/S on the local COB site DMX



The previous example for the 3-site solutions illustrates several points:

- There must be a static RDF group between each pair of DMX. The RDF group must be configured across corresponding ports on the source and Target DMX.
- The RAs do not need to match between DMX. For example, port 13C on the source DMX is matched with port 3A on the SRDF/S Target DMX.

The actual number of additional boards that will need to purchase will depend on how many ports on the default configuration have been nominated for host FAs.

It is recommended that odd RDF groups be used for Sync legs and the following even number for the corresponding Async leg. This will help avoid manipulating the wrong leg in a concurrent SRDF environment. For example, for an application with RDF group 5 for the SRDF/S leg, the corresponding SRDF/A leg should use RDF group 6.

4.8 DMX bin file configuration

There are a number of bin file configuration requirements that are specific to an SRDF/A configuration and which go beyond and are in addition to the recommendations for bin file layout made in the GDSE document “**EMC Symmetrix DMX Best Practices**”, which should also be followed.

4.8.1 Microcode levels

All DMX1/2 must be loaded with 5671.5459 microcode or later. See Release Notes for further details of preferred versions, any additional patches/ePacks required and any other known issues. DMX3 users should use latest GDSE certified version of 5771 microcode.

4.8.2 Dynamic RDF bin file requirements

The DMX needs to be configured for Dynamic and Concurrent RDF operations. The following flags must be enabled:

```
Switched RDF Configuration State      : Enabled
Concurrent RDF Configuration State    : Enabled
Dynamic RDF Configuration State       : Enabled
Concurrent Dynamic RDF Configuration : Enabled
```

Also, any Director ports to be used for SRDF need to be set for **switched operations**. RA port flags need to be set to FF, which allows multiple RDF groups to share the same port.

All data volumes must be enabled for Dynamic RDF. The VCMDB, gatekeepers and any other volumes that will never be used with SRDF do not need to be enabled for Dynamic RDF.

R1 and their corresponding R2 volumes should be configured with the same volume number. Although this is not strictly necessary, it will significantly simplify and reduce error when configuring a dynamic RDF setup.

Finally, it is necessary to create a static RDF group between corresponding ports on a pair of DMX. This is a pre-requisite for creating dynamic RDF groups. For example, in a concurrent SRDF setup where DMX A is the source and B & C are the targets, the following static groups will need to be defined by EMC:

- RDFG 1 between all ports on DMX A and each of their corresponding ports on DMX B
- RDFG 2 between all ports on DMX A and each of their corresponding ports on DMX C

4.8.3 BCV requirements at target site

A BCV device needs to be configured in the bin file for every R2 device on the SRDF/A target DMX. These BCVs will be used to take a point-in-time (PiT) copy of the R2 volumes prior to resynchronisation of the SRDF pairs after SRDF/A has been suspended. This is required to ensure that there is an I/O consistent copy of production data at all time at the target site. They can also be used, if required, for COB testing.

In the case where BCVs are needed at the SRDF/A target site for both purposes, it will be recommended to:

- a) Mirror the SRDF/A R2 device

- b) Share a single BCV for each R2 volume for both the COB and for the SRDF/A PIT copy requirement.

4.8.3.1 Best Practice for BCV type and TimeFinder mode usage

RAID-1+0 R2 standards

RAID-1+0 R2 standards should be paired with unprotected RAID-0 BCVs. RAID-5 BCVs should not be used (due to spindle split limits and other interoperability constraints – see details in Appendix D).

TimeFinder/Mirror (either “Classic” and Clone emulation modes) should be used for RAID1+0 (or RAID-0) standards. BCVs should be built from the same physical disk types as standards when Classic TimeFinder/Mirror is used.

- **In 2-Site SRDF/A and 3-Site Concurrent SRDF deployments**, use one of the following:
 - o **Preferred option.** Use “Classic” TimeFinder/Mirror (with RAID-0 BCVs) on disks at least as fast as the R2 standard devices). “Classic” TimeFinder/Mirror is the more mature product, with more expertise currently available in both Citigroup and EMC. Can be used with DMX2 or DMX3.
 - o **Alternate option.** Use TimeFinder/Mirror Clone Emulation (again, with RAID-0 BCV). This option can be used when a mirror position must not be used by TimeFinder to permit, for example, use of dynamic hot sparing at the R2 end. Can be used with DMX2 or DMX3.

The long-term direction is to move away from using “Classic” TimeFinder/Mirror.

- **In 3-Site SRDF/Star deployments:**
 - o Use of TimeFinder/Mirror Clone emulation mode is preferred in this configuration as switch operations require a free mirror position, and will fail if a BCV is established at the R2 end (see SRDF/Star Solutions Guide for further details).
 - o “Classic” TimeFinder/Mirror could be used, but any BCV copies **MUST** be split off before an SRDF/Star switch operation can take place. This applies to both SRDF/A resync BCVs AND user BCVs. Use of “Classic” TimeFinder/Mirror by users should not be permitted in an SRDF/Star environment.

RAID-5 R2 standards

For all SRDF configurations, RAID-5 R2 standards should be protected with RAID-5 BCVs, with identical configuration. TimeFinder/Mirror should be used (clone emulation will automatically be invoked). Can be used with DMX3 only.

4.8.3.2 BCV configurations

The following configuration ensures that COB site disk performance is identical to production performance i.e. is a balanced system. This configuration results in the Source DMX (and also

the Sync Target DMX for 3-site SRDF) being only 66% full. The example shows 146GB disks. It applies to all disk sizes, but the key point is that the source and SRDF/A target sites should have matching disk types.

	2 or 3-site Configuration 146GB throughout*	2 or 3-site Configuration 146GB throughout*
	Where Sync R2 protection is via a BCV	Where Sync R2 is mirrored standard (M1 + M2)
Source R1 devices	Mirrored on 146GB disks	Mirrored on 146GB disks
Sync Target R2 devices (3-site only)	R2 (unmirrored) on 146GB disks with unmirrored BCV on 146GB disks	Mirrored on 146GB disks
SRDF/A Target R2 devices	R2 (mirrored) on 146GB disks	R2 (mirrored) on 146GB disks
SRDF/A Target SRDF/A PIT BCV	Unmirrored BCV on 146GB disks	Unmirrored BCV on 146GB disks

* Note that the 146GB disks in the table above should be replaced with 73GB disks throughout if 73GB are used in Production.

Disks should not be used at the async target site that are larger than those at the production site. The larger disks at the target sites will not have the same performance characteristics, and this is known to cause issues with SRDF/A due to the R2 disks having lower performance than the R1 disks and being unable to keep up.

4.8.4 BCV usage constraints at source site

It is not possible to use a TimeFinder/Mirror BCV with a mirrored volume that is the source of concurrent SRDF replication. If BCV-type functionality is required, it will be necessary to use either TimeFinder/Mirror Clone emulation or TimeFinder/Clone technology to create PIT copies at the source site. Clones have already been certified for use in this circumstance by GDSE. The relevant documents are listed in the Document Scope section of this document.

4.8.5 Additional Gatekeeper requirements

A minimum of 3 Gatekeepers is required on each management server for SRDF configuration, control and reporting.

If Multi-Session Consistency (MSC) is to be used, then one additional gatekeeper is needed per Composite Group on the source site management server. It will be necessary to put additional Gatekeepers into your DMX configuration if implementing MSC.

4.8.6 Hot Spares and Symptomiser

With **Concurrent SRDF** it is not possible to use Hot Spares on the source DMX, as all four mirror positions are used up for local mirroring and two SRDF relationships. Similarly, as Symptomiser needs a spare mirror position, it is also not compatible with concurrent SRDF.

4.9 Storage Software Considerations

4.9.1 SRDF/A licensing

Any DMX to be used in an SRDF/A configuration should be purchased with the High Availability (HA) bundle pricing. This includes one SRDF license. The licenses can be used for either running the DMX in SRDF/A (Option A) or SRDF/S with SRDF Adaptive Copy (Option B) but not both.

Note that Adaptive Copy is covered by either an SRDF or an SRDF/DM license key and it is not possible to change to this mode if you only have an SRDF/A license key; an SRDF license is needed to set mode to sync and adaptive copy mode. An SRDF/A license is needed to set mode to asynchronous mode.

EMC intend to change this functionality so that the SRDF/A license will also permit changing to adaptive copy mode. As a workaround until that time, EMC have agreed to also supply an SRDF/DM key when an SRDF/A license is purchased. This will permit changing mode to adaptive copy, but not to synchronous mode.

Hence, for a 2-site SRDF/A solution, no additional SRDF licenses are required for either the source or target DMX over and above those supplied with the HA bundle, but both SRDF/A and SRDF/DM keys will be needed.

For a 3-site solution, it is necessary to purchase an additional SRDF license for the source DMX (only).

The SRDF/A target DMX will also need a TimeFinder / Consistency Group license to permit consistent PIT copies to be made prior to SRDF resynchronisations *if* splits are made when SRDF is still running, for example for COB testing purposes.

If the use of Multi-Session Consistency (MSC) is required, an SRDF Consistency Group license will be needed on each DMX (i.e. both source and target).

4.9.2 Solutions Enabler 6.x

SRDF/A requires Solutions Enabler 6.0.3 as a minimum version on the management servers.

4.9.3 InfoSec implications

SRDF/A does not present any new authentication or access issues over and above those already present in the existing Production SAN infrastructure.

SRDF/A is simply a different mode of SRDF operation. It uses the same command interface (Solutions Enabler) as synchronous SRDF and the same SAN management servers, and hence should be covered by the existing Symmetrix Product ICQ and Standard Build.

Hosts which do not have symacl admin rights and which do not have the symacl ACE RDF cannot control MSC replication.

4.10 Other planning considerations

4.10.1 Dynamic RDF

Multi-session SRDF/A requires the use of Dynamic RDF by the SAN BAU team for certain operations. Today, dynamic SRDF is not always used at all by BAU teams, or where it is used, all configuration work is done prior to a Symmetrix going into production, and no subsequent changes are made. This represents a significant new level of complexity to the storage provisioning process.

To be specific:

- 1) BAU teams must create separate RDF groups for individual applications or groups of applications. Subsequently it may be necessary to modify or destroy those groups when an application is retired or migrated.
- 2) BAU must add new devices for a server or application into the appropriate RDF group. If storage is later reallocated, it will be necessary to remove that storage from the application's RDF group and move it either back to a pool of non-SRDF enabled disks, or to another RDF group. Note that every time a device is added to an RDF group, it is necessary to re-sync the whole device (full establish).

Procedures for managing Dynamic RDF and its impact on the storage provisioning process are documented later in this guide. However, it may be necessary to enhance SAN BAU resources to accommodate this extra workload.

4.10.2 Performance Impact Considerations

SRDF/A does not impact from write IO response in the same way that synchronous SRDF does as it does not have to wait for the remote site to acknowledge each IO before waiting for the next. However, there are some characteristics that anyone deploying SRDF/A should be aware of.

- SRDF/A was found to have no noticeable impact on write response in 3-site, concurrent configurations as the SRDF/S leg response time dominates
- SRDF/A was found to cause a noticeable increase in write-response compared with no SRDF replication. This was very modest for large IOs, but large (50%) for very small IOs.
- SRDF/A read response for very small IOs (only) is marginally worse than with no SRDF replication.

4.11 Large Network Latency Implementations

A number of performance and stability issues were found relating to replication over very long latency links (e.g. 250ms). These are summarized below for planning purposes. For more detail see GDSE document “**SRDF/A Evaluation**” http://gdse.ny.ssmb.com/GDSE_docs/GDSE-05002-E.pdf, Section 6.11.

4.11.1 Reduced throughput and stability with packet loss at 250ms latency

Testing has shown that simulated packet loss:

- reduces effective replication throughput as a function of latency
- causes SRDF/A to drop out, with greater sensitivity to packet loss as a function of latency i.e. the threshold for SRDF/A to abort for a given packet loss rate decreases as latency increases.

4.11.2 Effective bandwidth as a function of latency

Even in an environment with a “perfect” network link (no jitter, no packet loss, no packet re-ordering etc), there is a reduction in throughput as a function of latency, as described earlier in this document. Effective bandwidths may be just 10% or less of the maximum potential bandwidth of the link if not optimized (see Section 4.4 for an example).

The GDSE document “**WAN Throughput Optimization for SRDF/A Solutions Guide**” gives advice on minimizing the impact of latency on throughput.

4.11.3 Cycle time increases with latency for MSC replication

For RDF groups under MSC control, cycle time will increase beyond the pre-set value in proportion to latency on the network and the number of sessions under MSC control.

Cycle times as high as 90 seconds were observed in testing for even zero IO load due to the overheads of running MSC over long latency links (tested up to 250ms).

For further details see GDSE document “**SRDF/A Evaluation**” http://gdse.ny.ssmb.com/GDSE_docs/GDSE-05002-E.pdf, Section 6.10.1.

4.11.4 Long runtime for control/query commands

High latency also has a negative impact on the runtime of many control and query operations. This is because many of these commands are written in such a way that they communicate multiple times with the remote DMX for each invocation.

The practical impact is that some commands may take 10 to 20 times longer to run over, say, a 250ms round trip latency link compared with a low latency link. For further details see GDSE document “**SRDF/A Evaluation**” http://gdse.ny.ssmb.com/GDSE_docs/GDSE-05002-E.pdf, Section 6.11.5.

5 Generic SRDF/A configuration

In this section, configuration steps generic to all SRDF/A implementations are described. It is assumed that all steps of the SRDF/A preparation phase documented in the previous section have been completed.

Additional steps specific to 2-site Multi-Session consistency and 3-site Concurrent SRDF deployments are covered in Sections 6 and 7 respectively, and if relevant to your configuration, those sections must be read and understood before any configuration work takes place.

The main generic SRDF/A configuration steps are:

1. Management Server Configuration
2. Network Equipment Configuration
3. Dynamic RDF configuration
4. Target site Point-In-Time image for SRDF/A resynchronisation
5. Device group definitions
6. Enabling SRDF/A replication
7. Tuning SRDF/A settings
8. Replication monitoring

This section describes how to build a 2-site SRDF/A solution from scratch. Some advice on how to convert an existing 2-site SRDF/A solution into a 2-Site SRDF/A solution can be found in Appendix C, for the scenario where the production site remains fixed but the COB/DR site swings to a new highly remote location.

5.1 Management Server Configuration

5.1.1 Solutions Enabler 6

Latest GDSE certified version of Solutions Enabler 6 should be installed on all SAN management servers to be used for SRDF/A management. For further details of this process see the Solutions Enabler 6.x Solutions Guide and relevant EMC documentation.

GNS should be used for replication of device and composite group information in a 2-site configuration. Unfortunately, GNS remote mirror replication does not work for device or composite groups containing devices using concurrent SRDF. GNS can, however, be used in all configurations to ensure changes to device and composite definitions are automatically visible on other local SAN management servers.

5.1.2 Software Licensing

The following additional licenses should be registered using symlmf for an SRDF/A implementation (if not already registered):

The license keys should be obtained from your local EMC account team.

Functionality	License available via	Required feature (as it appears in symapi_licenses.dat file)
Base Solutions Enabler functionality	Both HA bundles	Feature: BASE / Symmetrix
SRDF/S & Adaptive Copy	HA bundle (Option A)	Feature: SRDF / Symmetrix
SRDF/A & Adaptive copy	HA bundle (Option B)	Feature: SRDFA / Symmetrix Feature: SRDF/DM
TimeFinder/Mirror	Both HA bundles	Feature: TimeFinder / Symmetrix
TimeFinder/Clone	Both HA bundles	Feature: TimeFinder / Symmetrix
Optional: TimeFinder/Consistency Groups	Incremental software license purchase	Feature: TF/CG / Symmetrix
SRDF/ Consistency Groups	Incremental software license purchase	Feature: SRDF/CG / Symmetrix
Configuration Manager	Both HA bundles	Feature: ConfigChange / Symmetrix

The TimeFinder/Consistency Groups license is needed to create an IO consistency Point-in-time copy of the R2 prior to resynchronisation. It is only needed for the SRDF/A target DMX and only if copies are made when SRDF is still running. The Best Practice for SRDF/A resyncs is to make the copy when SRDF is down, and hence is optional. If, however, a PiT copy is ever needed for COB testing and the application is running when the copy is made, then this license would be needed to guarantee consistency of the copy.

The SRDF/ Consistency Groups license is only needed if implementing MSC.

Note that use of the consistency for devices within a device group (symdmg enable) does not need any additional licenses over and above the Base Solutions Enabler License.

5.1.3 /etc/system

The following are standard entries for /etc/system for Solutions Enabler 6.x on Solaris 8. However, they are re-iterated here as testing has shown that SRDF/A control does not work correctly if they are not set.

Standard Solutions Enabler requirements for Solaris 8:

```
set semsys:seminfo_semmni=600
set semsys:seminfo_semmns=600
set semsys:seminfo_semume=600
set semsys:seminfo_semmnu=600

set shmsys:shminfo_shmmax=1048576
set shmsys:shminfo_shmmni=600
```

5.1.4 /var/symapi/config/options

The following options should be configured in the symapi options file.

```
SYMAPI_USE_RDFD                = ENABLE
SYMAPI_ALLOW_RDF_SYMFORCE      = TRUE
SYMAPI_WAIT_ON_LOCKED_GK       = ENABLE
```

SYMAPI_USE_RDFD controls if the RDF Daemon is used. This will need to be enabled for MSC control.

SYMAPI_ALLOW_RDF_SYMFORCE is FALSE by default, as it is dangerous. However, this needs to be set to TRUE in an MSC environment, as symforce is needed with Composite Group failovers following an MSC-triggered suspension of SRDF/A.

SYMAPI_WAIT_ON_LOCKED_GK should be enabled to avoid command failure when due to Gatekeepers not being available.

Also, if GNS is to be used for mirroring of device and composite group definitions (recommended), you will need the following additional entries:

```
SYMAPI_USE_GNS                  = ENABLE
```

5.1.5 /var/symapi/config/daemon_options

The following options should be configured in the symapi daemon_options file. It will be necessary to first copy the template file and then edit.

```
# cp /var/symapi/config/README.daemon_options /var/symapi/config/ daemon_options
```

The logfile_type entries are not essential but do help with debugging issues with the daemons.

```
storrdfd:logfile_type=dated
storapid:logfile_type=dated
storgnsd:logfile_type=dated
storwatchd:logfile_type=dated
```

Also, if GNS is to be used for remote mirroring of device and composite group definitions, you will need the following additional entries:

```
storgnsd:autorestart=enable
storgnsd:gns_remote_mirror=enable
```

Entries of the form shown in the following example control RDF daemon usage.

```
# Set max gatekeepers on 0822 to 38
# Also set dedicated gatekeepers for RDFD to 10

storapid:max_gatekeeper_0822=38
storrdfd:rdfd_num_dedicated_gks=10
storrdfd:autorestart=enable
```

Optional Behaviour Parameter	Description
storapid:max_gatekeeper_<SID>	Controls the total number of Gatekeepers that are reserved for usage with the Symmetrix with the specified SID.
storrdfd:rdfd_num_dedicated_gks	Reserves the specified number of gatekeeper devices exclusively for use by the RDF daemon. Can be set from 0 – 20. Default is 2.
storrdfd:autorestart	If set to enable, makes use of the watchdog mechanism to automatically restart the daemon if it crashes.

Hence, in the example given, 38 GKs are reserved for use with Symmetrix 0822 and 20 are reserved for the RDF daemon.

5.1.6 Gatekeeper requirements

A minimum of 3 Gatekeepers is required on each management server for SRDF configuration, control and reporting.

If Multi-Session Consistency (MSC) is to be used, then one additional gatekeeper is needed per Composite Group on the source site management server.

5.1.7 symacl configuration requirements

Servers used for SRDF/A control should be owned by the SAN BAU team and should be members of the symacl AdminGrp on all the DMX involved in an SRDF configuration.

5.2 Network Equipment Configuration

5.2.1 McData IPS configuration

The IPS units should be configured for SRDF/A as per the detailed documents:

- **McData IPS SAN Router Solutions Guide V2**
- **WAN Throughput Optimization for SRDF/A Solutions Guide**

See references in Section 1 for the URLs.

5.2.2 Cisco Switch configuration

The network team should be requested to configure the Cisco Switch ports that connect to the IPS units with negotiation set to auto.

The active primary link(s) between the local and remote Cisco switches that connect to the local and remote IPS units must have the same latency between them. The secondary (redundant) route used when the primary route(s) fails can have a different latency. No load balancing across routes with different latencies must occur.

5.3 Dynamic RDF configuration

5.3.1 Introduction

Use of dynamic RDF allows the end user to easily create, manage and populate RDF groups. This is essential for SRDF/A where failover is at the RDF group level (unlike SRDF/S which is at the device level) and hence each application will usually have its own RDF group for its devices.

In this section, the steps required to create and populate RDF groups ready for use with SRDF/A are summarized.

5.3.2 Dynamic RDF requirements

- **Dynamic RDF configuration state of the Symmetrix must be enabled**
 - o EMC should set this at time of Symmetrix install. It can also be done from SYMCLI using symconfigure, although this latter approach is not covered in this document.

The following command can be used to check this setting.

```
# symcfg -sid <SID> list -v
```

Dynamic RDF Configuration State should be Enabled, as should Switched RDF, Concurrent Dynamic RDF & Concurrent RDF.

For example:

```
# symcfg -sid 0822 list -v
```

```
Symmetrix ID: 000187700822
Time Zone   : EDT
```

```
Product Model      : 2000P-M2
Symmetrix ID       : 000187700822
```

[Intermediate Lines Deleted]

```
Switched RDF Configuration State : Enabled
Concurrent RDF Configuration State : Enabled
Dynamic RDF Configuration State : Enabled
Concurrent Dynamic RDF Configuration : Enabled
RDF Data Mobility Configuration State: Disabled
Access Control Configuration State : Enabled
Device Masking (VCM) Config State : Enabled
```

- **Devices must be designated as RDF-capable**
 - o EMC should set this at time of Symmetrix install. It can also be done from SYMCLI using symmconfigure, although this latter approach is not covered in this document.

The following command can be used to verify the devices have been configured as dynamic RDF capable. Optionally one can also list which are currently configured as R1 and which as R2.

```
# symdev -sid <SID> list -dynamic [-R1] [-R2]
```

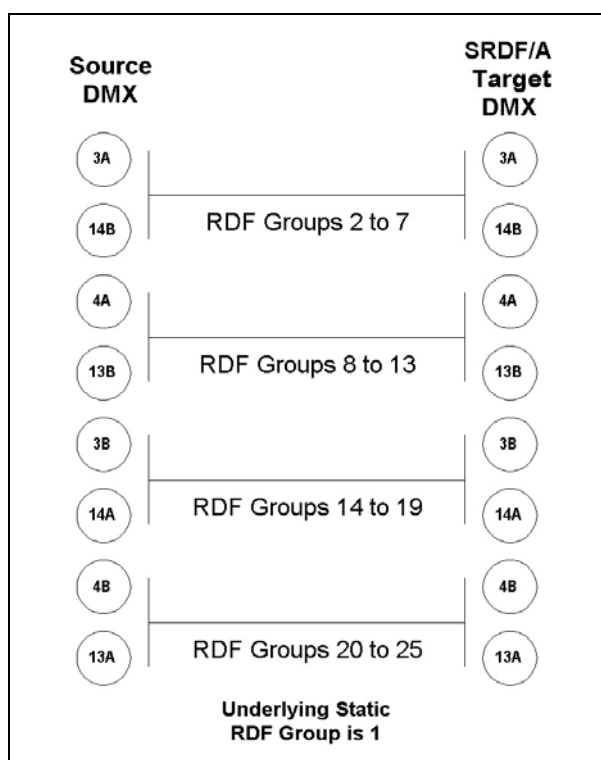
5.3.3 Mapping RDF groups to RAs

Each pair of RAs on a DMX can support 8 RDF groups (without an RPQ). You should carefully document:

- Which ports on each DMX correspond with each other. For example, port 3A on the source DMX may correspond to port 3A on the SRDF/A target.
- Which RDF groups are to run over each pair of ports.

This can either be documented using a diagram or a table as follows. This information will be needed when the RDF groups are created. Each RDF group should be given a sensible label. RDF groups cannot be defined with leading zeros. In many implementations, the source RA and target RA may not (as shown in this example) be the equivalent port.

Note that the following is an example of a 2-site configuration. Your hardware configuration and RDF groups requirements (and hence RA-to-RDF mappings) will almost certainly be different.



Label	RDF Group	Source RAs (DMX 0822)	Target RAs (DMX 1497)
Static	1	ALL	ALL
DYN2	2	3A / 14B	3A /14B
DYN3	3	3A / 14B	3A /14B
DYN4	4	3A / 14B	3A /14B
DYN5	5	3A / 14B	3A /14B
DYN6	6	3A / 14B	3A /14B
DYN7	7	3A / 14B	3A /14B
DYN8	8	4A / 13B	4A /13B
DYN9	9	4A / 13B	4A /13B
DYN10	10	4A / 13B	4A /13B
DYN11	11	4A / 13B	4A /13B
DYN12	12	4A / 13B	4A /13B
DYN13	13	4A / 13B	4A /13B
etc.			

5.3.4 Creating dynamic RDF groups

Dynamic RDF groups are created on-the-fly with the symrdf command. For example, the following commands would be used to create RDF group 2 on the ports specified above.

```
# symrdf addgrp -label DYN2 -rdfig 2 -sid 0822 -dir 3A -link_limbo 60
-remote_rdfg 2 -remote_sid 1497 -remote_dir 3A -rem_link_limbo 60
```

```
# symrdf modifygrp -rdfig 2 -sid 0822 -add -dir 14B -remote_dir 14B
```

The first command creates the RDF group on the first pair of RAs. The second adds further RAs to the RDF group. The 60-second link_limbo timer should only be used for RDF groups that are to be used for SRDF/A. The link limbo timer is the length of time that the Symmetrix will tolerate the loss of links before concluding that the link has failed.

The mapping between RDF groups and RA ports of the dynamic RDF groups can be shown using the symcfg command.

```
# symcfg list -ra all -switched -sid <SID>
```

The example below is for a DMX pair with multiple RDF groups configured. In this example, the static RDF group between the matching ports on each DMX is group 2. An additional 4 dynamic RDF groups have been created on each port. Note also that in this example, each RDF group has so far been configured on only one port on each DMX. Each RDF group must be configured over a minimum of 2 ports on each DMX for redundancy.

Symmetrix ID: 000187700822

S Y M M E T R I X R D F D I R E C T O R S

Local				Group		Remote				
Ident	Symb	RA	Grp	Type	Name	SymmID	Ident	Symb	RA	Grp
RF-4C	04C	11	(0A)	Dynamic	DYN11	000187721497	RF-8C	08C	11	(0A)
		2	(01)	Static	SRDFA	000187721497	RF-8C	08C	2	(01)
		12	(0B)	Dynamic	DYN12	000187721497	RF-8C	08C	12	(0B)
		13	(0C)	Dynamic	DYN13	000187721497	RF-8C	08C	13	(0C)
		14	(0D)	Dynamic	DYN14	000187721497	RF-8C	08C	14	(0D)
RF-13C	13C	2	(01)	Static	SRDFA	000187721497	RF-9C	09C	2	(01)
		19	(12)	Dynamic	DYN19	000187721497	RF-9C	09C	19	(12)
		20	(13)	Dynamic	DYN20	000187721497	RF-9C	09C	20	(13)
		21	(14)	Dynamic	DYN21	000187721497	RF-9C	09C	21	(14)
		22	(15)	Dynamic	DYN22	000187721497	RF-9C	09C	22	(15)
RF-3D	03D	2	(01)	Static	SRDFA	000187721497	RF-7D	07D	2	(01)
		7	(06)	Dynamic	DYN07	000187721497	RF-7D	07D	7	(06)
		8	(07)	Dynamic	DYN08	000187721497	RF-7D	07D	8	(07)
		9	(08)	Dynamic	DYN09	000187721497	RF-7D	07D	9	(08)
		10	(09)	Dynamic	DYN10	000187721497	RF-7D	07D	10	(09)
RF-4D	04D	15	(0E)	Dynamic	DYN15	000187721497	RF-8D	08D	15	(0E)
		2	(01)	Static	SRDFA	000187721497	RF-8D	08D	2	(01)
		16	(0F)	Dynamic	DYN16	000187721497	RF-8D	08D	16	(0F)
		17	(10)	Dynamic	DYN17	000187721497	RF-8D	08D	17	(10)
		18	(11)	Dynamic	DYN18	000187721497	RF-8D	08D	18	(11)
RF-13D	13D	23	(16)	Dynamic	DYN23	000187721497	RF-9D	09D	23	(16)
		2	(01)	Static	SRDFA	000187721497	RF-9D	09D	2	(01)
		24	(17)	Dynamic	DYN24	000187721497	RF-9D	09D	24	(17)
		25	(18)	Dynamic	DYN25	000187721497	RF-9D	09D	25	(18)
		26	(19)	Dynamic	DYN26	000187721497	RF-9D	09D	26	(19)

5.3.4.1 Deleting dynamic RDF groups

Dynamic RDF groups can be deleted with the command of the form

```
# symrdf removegrp -sid 0822 -rdfig 2
```

Note that all RDF device pairs must be removed from the RDF group before it can be deleted (see below).

5.3.5 Creating dynamic RDF device pairs

A device pairs file is necessary to create RDF pairs. The device pairs file syntax contains two columns (R1 and R2 designated devices in the first and second columns respectively). Each SRDF pair must be listed on a separate line in the device file.

The creation of the dynamic SRDF pair utilizes the `createpair` option of the `symrdf` command. Similarly, the deletion of a dynamic SRDF pair utilizes the `deletpair` option of the `symrdf` command.

The pairs file should only include meta heads. In the example below, each meta is made up of 4 hypers as per the GDSE Bin file Standard. The first device in the first column has the meta head of 0157 and 3 additional hypers (0158, 0159, 015A) making up the volume. Only the meta head is included in the file.

Device file "pairsfile_DYN2"

```
0157 0157
015B 015B
015F 015F
0163 0163
0167 0167
016B 016B
016F 016F
etc.etc
```

In this example, the R1 and corresponding R2 have the same volume number. It is strongly recommended that all DMX using Dynamic RDF be configured in this way.

A pairs file can be created using the `symdev list -dynamic` command to initially get a listing of dynamic RDF capable devices. Further manipulation of the file will be necessary to get a listing of just the meta heads in the right format. In the case where the R1 devices are mapping onto R2 devices with identical volume numbers, then the following command on a Unix system will generate the pairs file. In this example, the Symm in question has a sid of 0822.

```
# symdev -sid 0822 list -dynamic | grep "(M)" | awk '{print $1 " " $1}' > pairsfile_0822
```

This pairs file for the whole DMX can be used as a template to be edited down for pairs files for individual RDF groups.

The following pair creation example uses a pairfile called "pairsfile_DYN2", which only contains the devices to be added to RDF group2. The `-type rdf1` option indicates that the devices in the first column (which are in the Symm specified by the `-sid` option) are R1 devices. If GNS is in use with RDF group mirroring enabled, then the newly created device group will be mirrored to the target site (in a 2-site configuration only) and to the other local SAN management servers.

The following command invalidates the R2, merges track table, brings up the RDF links and starts the copy process from R1 to R2. This can be a very time consuming process. When this command runs, it locks the symapi database so no other symcli commands can be executed from that host, even to other Symmetrix. Hence, the running of the command has to be carefully scheduled.

```
# symrdf -file pairsfile_DYN2 -sid 0822 -rdfig 2 createpair -type rdf1
      -establish -g DG2
```

The `-g` option creates an rdf1-type device group called DG2 that contains the devices in the pairs file and which can be used subsequently for SRDF control. Alternatively, local tools can be used to create the device group if preferred.

5.3.5.1 *Deleting dynamic RDF device pairs*

The same pairs file is used to delete dynamic SRDF pairs. To delete the pairing set up in the previous section, one would issue the command

```
# symrdf -file pairsfile_DYN2 -sid 0822 -rdfg 2 deletepair
```

Note that the symrdf deletepair command is only allowable in one of the following states, where the links are not ready status:

- Suspended
- Split
- Failed over

5.3.6 Checking the status of dynamic RDF device pairs

There are two commands for querying the dynamic RDF device pair status.

```
# symdev -sid <sid> list -dynamic
```

will show the configuration of the dynamic RDF devices.

- Devices marked as 2-Way Mir do not have SRDF enabled.
- Devices marked as either RDF1+Mir or RDF2+Mir have SRDF enabled.

```
# symrdf -sid <sid> list -dynamic
```

will show the pair state of the dynamic RDF devices (Synchronized, Failed Over etc).

Note that 2-Way mirror devices are not shown with the symrdf command.

5.4 Target site Point-In-Time image for SRDF/A resynchronisation

5.4.1 Overview of requirement

BCVs will need to be configured into the SRDF/A target site to provide the ability to take a point in time (PiT) copy of the R2 data prior to a resynchronization to ensure that there is a consistent copy of the data at the remote site at all times. During a resynchronisation, the data on the R2 will be inconsistent and hence during this period there is no valid COB copy of the data at the remote site, unless a PiT copy is first taken.

Hence, the BCVs are not required during normal SRDF/A replication operations, but are there as a safeguard during resynchronisation operations.

5.4.2 Best Practice Summary

BCVs should be used for SRDF/A resynchronization with Timefinder/Mirror. Best Practice for each disk RAID type and SRDF mode has already been covered in the planning section for PiT copies (Section 4.8.3), and is not repeated here. A detailed description of the reasoning behind the Best Practice can be found in Appendix D.

5.4.3 Mapping PIT devices to standards

During the initial configuration for the SRDF/A environment, it will be necessary to nominate one BCV in the target DMX for every SRDF/A R2, or a range of BCVs for each set of R2s in an RDF group if optimised pairing is being used (recommended – see Section 5.4.4.)

As R1 disks are allocated to RDF groups, the relevant BCVs need to also be associated with the device group (as remote BCV devices) and the BCV established with the R2.

Care must be taken to minimize backend DA and spindle contention between R2 devices and the BCV devices (see the end of Appendix D for further details). It is recommended that the `-opt` flag (or `-opt_rag -rdf` if BCVs are remote from the SE host) be used with the first full establish to create non-contentious source / target pairings.

The procedure for using these BCVs during a resynchronization is documented later in this guide.

5.4.4 Adding BCVs to device groups

BCVs can be added to groups using existing local tools used at your site for BCV management. Alternatively, this can be achieved using the syntax in the example below.

On the management server at the source site, the BCVs in the target DMX should be associated as remote BCVs.

On the management server at the target site, the BCVs in the target DMX should be associated as local BCVs. If GNS remote mirror replication is in use, the device group definition changes made at the source site will be replicated to the remote site and the BCV type changed to local automatically.

The following example shows BCVs being associated at the source site as remote BCVs. RDF group 3 contains 4 standard volumes. The group associated with this RDF group is DG3. The BCVs to be associated with the R2 volumes are 02AD, 02B1, 02B5, 02B9. The `-rdf` flag indicates that these are remote BCVs.

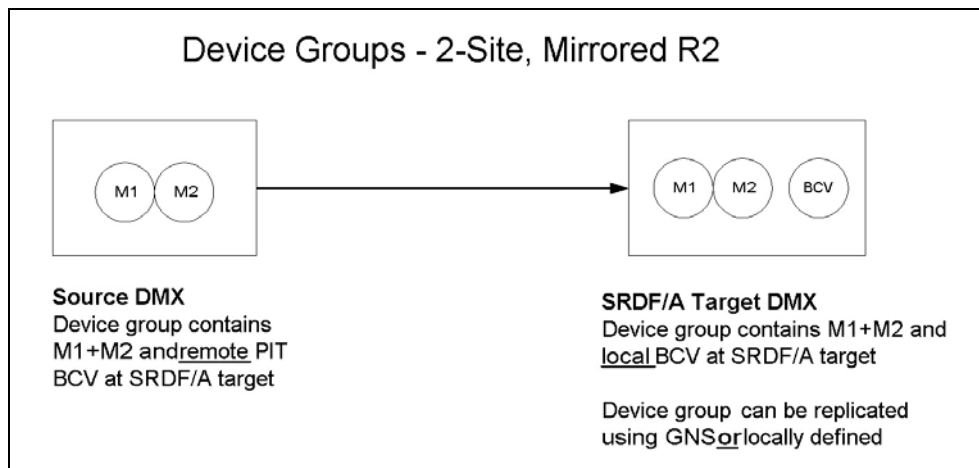
```
# symbcv -g DG3 -sid 0822 associate dev 02AD RBCV02AD -rdf -rdfg 3
# symbcv -g DG3 -sid 0822 associate dev 02B1 RBCV02B1 -rdf -rdfg 3
# symbcv -g DG3 -sid 0822 associate dev 02B5 RBCV02B5 -rdf -rdfg 3
# symbcv -g DG3 -sid 0822 associate dev 02B9 RBCV02B9 -rdf -rdfg 3

# symmir -g DG3 establish -opt_rag -rdf -nop
```

Scripts **add_PIT_BCV** and **rm_Pit_BCV** have also been provided to automate the process of adding and removing remote PiT BCVs to and from async R2 standards. See Appednix B for further details.

5.5 Device group definitions

The following device group set up is recommended for a 2-site SRDF/A configuration. Note that the setup for a 3-site configuration is considerably more complex and is covered in Section 7 for non-Star Concurrent SRDF implementations. SRDF/Star configurations are covered in the separate Solutions Guide “**SRDF/Star for Open Systems**”.



Configuration:

Config 1: 2-site R2=M1+M2

Device Group contents:

- Source DMX All R1s in RDF group + remote SRDF/A PIT BCV
- SRDF/A Target All R2s in RDF group + local PIT BCV

Device Group Replication

GNS can be used to replicate definitions to both local SRDF/A management servers and those at the remote site.

At sites where COB testing is undertaken but invoking COB and running from the R2, the BCV is only used for SRDF/A resynchronisation.

At sites where COB testing is undertaken by running from a BCV, the BCV will be used for both COB testing and SRDF/A recovery and usage will need to be co-ordinated.

The SRDF/A recovery script can be run from either site. BCVs can be controlled from either site, but for entire RDF group only.

5.6 Enabling SRDF/A replication

RDF groups are created in synchronous mode. The replication mode must be explicitly set to asynchronous. The `-rdfg <RDFG num>` is optional for a 2-site SRDF/A configuration.

The enable command enables consistency between devices within the device group.

```
# symrdf -g <DGname> -rdfg <RDFG num> set mode async -nop
# symrdf -g <DGname> -rdfg <RDFG num> enable
```

Confirm that SRDF/A is running successfully and that consistency is enabled.

```
# symrdf -g <DGname> -rdfg <RDFG num> query
```

Under the MODES section, the M column for each device pair should show A (for async), and the “RDF Pair STATE” should be Consistent. Note that it may take up to two cycles before it appears as “Consistent.”

For example

```
# symrdf -g DG19 -rdfg 19 query
```

```
Device Group (DG) Name      : DG19
DG's Type                   : RDF1
DG's Symmetrix ID           : 000187700822
Remote Symmetrix ID         : 000187721497
RDF (RA) Group Number       : 19 (12)
```

Source (R1) View					Target (R2) View					MODES	
-----					-----					-----	
Standard Logical Device		ST			LI		ST				
		A			N		A				
		T	R1 Inv	R2 Inv	K		T	R1 Inv	R2 Inv		RDF Pair
	Dev	E	Tracks	Tracks	S	Dev	E	Tracks	Tracks	MDA	STATE
-----					-----					-----	
DEV001	00BD	RW	0	0	RW	00BD	WD	0	0	A..	Consistent
DEV002	00C1	RW	0	0	RW	00C1	WD	0	0	A..	Consistent
Total											
Track(s)		-----		-----		-----		-----			
		0		0		0		0			
MB(s)		0.0		0.0		0.0		0.0			

Legend for MODES:

```
M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
D(omino)             : X = Enabled, . = Disabled
A(daptive Copy)      : D = Disk Mode, W = WP Mode, . = ACp off
```

Also, the command

```
# symstat -type cycle -reptype rdfa -rdfg <RDFG num>
```

will show a non-zero “Last Switch” entry and the “Active Cycle #” will be increasing.

5.7 Worked Example of a 2-site SRDF/A configuration

The following is an example of the end-to-end process for creating and populating an RDF group, enabling SRDF/A replication and setting up PiT BCVs at the async target site.

In this example, the device group is App1. The RDF group to be used is 3.

1. Create Asynchronous Leg RDF group

SRDF/A RDF groups should have link_limbo timer set to 60 seconds.

```
# symrdf addgrp -label DYN3 -rdfig 3 -sid 0822 -dir 3A -link_limbo 60
-remote_rdfg 3 -remote_sid 1497 -remote_dir 3A -rem_link_limbo 60

# symrdf modifygrp -rdfig 3 -sid 0822 -add -dir 14B -remote_dir 14B
```

2. Populate RDF group with LUNs and establish

The device group pairs to be added are listed in pairs file "pairsfile_App1". Use -g option to automatically create the device group App1 containing all devices.

```
# symrdf -file pairsfile_App1 -sid 0822 -rdfig 3 createpair -type rdf1
-establis -g App1
```

3 Check SRDF set up

```
# symrdf -g App1 query -rdfig 3
```

```
Device Group (DG) Name      : App1
DG's Type                   : RDF1
DG's Symmetrix ID          : 000187700822
Remote Symmetrix ID        : 000187721497
RDF (RA) Group Number      : 3
```

Source (R1) View					Target (R2) View					MODES	
-----					-----					-----	
Standard Logical Device	ST				LI	ST					
	A				N	A					
	T	R1 Inv	R2 Inv	K	T	R1 Inv	R2 Inv	MDA	RDF Pair		
	E	Tracks	Tracks	S	Dev	E	Tracks	Tracks		STATE	

DEV001	0001	RW	0	0	RW	0001	WD	0	0	S..	SyncInProg
DEV002	0005	RW	0	0	RW	0005	WD	0	0	S..	SyncInProg
DEV003	0009	RW	0	0	RW	0009	WD	0	0	S..	SyncInProg
DEV004	000D	RW	0	0	RW	000D	WD	0	0	S..	SyncInProg

Total											
Track(s)		0	0			0	0				
MB(s)		0.0	0.0			0.0	0.0				

Legend for MODES:

```
M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
D(omino)             : X = Enabled, . = Disabled
A(daptive Copy)      : D = Disk Mode, W = WP Mode, . = ACp off
```

4 Enable SRDF/A replication on the Async leg

The enable command enables consistency between devices within the device group.

```
# symrdf -g App1 -rdfg 3 set mode async -noprompt
# symrdf -g App1 -rdfg 3 enable
```

5 Add BCVs into device group. Establish BCVs.

BCVs can be added to groups either using existing local tools used at your site for BCV management or using the scripts provided as part of this Solution. A script called **add_PIT_BCV** (see Appendix B for further details) is available to automate this process using a simple STD-to-BCV input-mapping file. A script called **rm_PIT_BCV** is also available to backout the additions.

Alternatively, this can be achieved manually using the syntax in the example below.

On the management server at the source site, the BCVs in the target DMX should be associated as remote BCVs.

On the management server at the target site, the BCVs in the target DMX should be associated as local BCVs.

Example

SRDF/A RDF group 3 contains 4 standard volumes. The group associated with this RDF group is App1. The PIT BCVs to be associated with the R2 volumes are 02AD, 02B1, 02B5, 02B9. The `-rdf` flag indicates that these are remote BCVs.

```
# symbcv -g App1 -sid 0822 associate dev 02AD RBCV02AD -rdf -rdfg 3
# symbcv -g App1 -sid 0822 associate dev 02B1 RBCV02B1 -rdf -rdfg 3
# symbcv -g App1 -sid 0822 associate dev 02B5 RBCV02B5 -rdf -rdfg 3
# symbcv -g App1 -sid 0822 associate dev 02B9 RBCV02B9 -rdf -rdfg 3
```

Establish the remote BCVs

```
# symmir -g App1 establish -full -opt_rag -rdf -nop
```

6. Ensure that device group definitions at remote sites are up to date

GNS remote mirroring of device groups can be used in 2-site SRDF configurations (only). If GNS has been set up correctly, then the replication to both local replicas and those at the remote site will take place automatically.

The author is aware that the pre-GNS procedures for maintaining remote device groups differ enormously from site to site (manual entry, export/edit/import of device group definitions, scripts that process a device list, etc). Sites should continue to use these processes for concurrent SRDF device group management if not using GNS.

Tuning SRDF/A settings

There are a number of settings specific to SRDF/A that need to be made. Most are specific to each RDF group i.e. each replication session. One is a DMX wide parameter.

5.7.1 Link Limbo timer setting

The Link Limbo Timer (LLT) is a complex configuration parameter. For full details of its operation, please refer to the GDSE SRDF/A Evaluation document http://gdse.ny.smb.com/GDSE_docs/GDSE-05002-E.pdf. However, in brief the LLT is the length of time that the Symmetrix will tolerate the loss of links before concluding that the link has failed.

The LLT must be set correctly to ensure that SRDF/A does not abort during temporary network outages.

It is recommended that LLT be set to 60 seconds for all sessions. **The LLT can only be set at RDF group creation time.** It cannot be changed thereafter.

The current LLT setting be determined using the command

```
# symcfg list -rdg all
```

The column headed "LL (sec)" to the left shows the LLT setting in seconds for each RDF group.

5.7.2 Cycle time

The cycle time for each RDF group can be changed. However, It is strongly recommended that this be left at the default setting of 30 seconds.

The minimum SRDF/A cycle time for each RDF group can be determined using the command

```
# symcfg list -rdg all
```

The column headed "Cycle time" to the right shows the minimum cycle time setting in seconds for each RDF group.

5.7.3 Priority

It is possible to raise and lower the session priority of an SRDF/A group. This dictates which SRDF/A sessions should drop first should SRDF/A cache become exhausted.

SRDF/A sessions will fail in order of priority (lowest priority being the highest number and failing first). When a session is dropped, the cache it was using for SRDF/A is freed up. If there is still insufficient cache, the next lowest priority session will drop next. This will continue until there is sufficient cache for the remaining sessions (if any are left) to continue replication.

All RDF groups are created with a default priority of 33.). If multiple sessions have the same priority, then any of these can be dropped. To protect a particular session, it should be given a priority of less than 33 Sessions that should be sacrificed first should be given a priority higher than 33 (maximum is 64).

The priority of a session can be changed with symconfigure. A configuration file must first be created. In the following example, RDF group 2 is having its Priority changed to 50 i.e. it will be dropped before sessions with the default 33 priority if cache begins to run out.

```
# cat modify_rdfg_2_pri.cfg
set ra group 2, session_priority=50;

# symconfigure -sid 0822 -f modify_rdfg_2_pri.cfg -nop preview
# symconfigure -sid 0822 -f modify_rdfg_2_pri.cfg -nop prepare
# symconfigure -sid 0822 -f modify_rdfg_2_pri.cfg -nop commit
```

The priority for each RDF group can be checked using the command

```
# symcfg list -rdfg all
```

The column headed “Pri” to the far right shows priority for each RDF group. Remember, high number fail first.

5.7.4 SRDF/A Cache Utilization

When EMC qualify a particular SRDF/A configuration, they recommend how much cache is required for the SRDF/A caching.

The maximum cache usage percentage represents the fraction of user cache that SRDF/A is allowed to use if the need arises. The default setting is 94%. It can be reduced to ensure only the extra cache installed/earmarked for SRDF/A is used for that purpose i.e. that none of the original or nominally non-SRDF/A cache is used for SRDF/A.

The following example shows how to change the SRDF/A cache percentage. In this example, the percentage is changed to 50%. The value will almost certainly be different in your implementation and will depend on how much extra cache EMC determined was required for each of your DMX.

1) Check the current setting is set at the default 94%. Check for line with “SRDF/A Maximum Cache Usage” in output of

```
# symcfg list -v -sid <SID>
```

where <SID> is the SID of the DMX to be changed.

2) Create a configuration file that defines the change to be made.

```
# cat modify_rdfa_cache_percent.cfg
set Symmetrix rdfa_cache_percent=50;

# symconfigure -sid <SID> -f modify_rdfa_cache_percent.cfg -nop preview
# symconfigure -sid <SID> -f modify_rdfa_cache_percent.cfg -nop prepare
```

3) Execute the change

```
# symconfigure -sid <SID> -f modify_rdfa_cache_percent.cfg -nop commit
# symcfg sync
```

4) Confirm the change. Check for line with “SRDF/A Maximum Cache Usage” in output of

```
# symcfg list -v -sid <SID>
```

5.8 SRDF/A Replication Monitoring

This section describe how the following key SRDF/A status and metric information can be monitored and alerts generated under exceptional circumstance:

- Monitoring for inactive SRDF/A sessions
- Monitoring for cache utilization above a preset threshold
- Monitoring for cycle times above a preset threshold

All of the above tasks can be undertaken by running the script **srdfa_health** (see Appendix B for more detail) from cron at regular intervals on a Solaris server being used to manage SRDF/A. Individual scripts for each of the above monitoring tasks are also available (**mon_srdfa_inactive**, **mon_srdfa_cache** & **mon_srdfa_cycle** – see Appendix B for further details).

All of the above scripts will write a message to /var/adm/messages using the logger command if one of their criteria for raising an alert is met. The message will of type local0.error be of the form:

```
Aug  2 06:24:07 sarprod2-ny SRDFA_ALERT: [ID 702911 local0.error] Please
contact SAN Support team. Current cycle time for RDF group 25 (66 sec)
on Symmetrix 0822 has exceeded acceptable threshold (60 sec)
```

All messages will include the string SRDFA_ALERT. You will need to instruct your local Enterprise Monitoring (EMS) Team to monitor the /var/adm/messages file on the server running the SRDF/A monitoring script(s) and set their system to raise an alert to the appropriate support team should a message with the string SRDFA_ALERT be detected. The contents of the message should also be forwarded by EMS to the support team.

For SE6.2 onwards, an event daemon is available from EMC that can be used to monitor Symmetrix Operations. The event daemon provides SNMP MIB support and trap generation services required to monitor the status of the Symmetrix using third-party enterprise management frameworks. The following events relating to SRDF can be monitored for (inexhaustive list):

- SRDF/A session
 - o SRDF/A session inactive
 - o SRDF/A session dropped; write pending limit reached. Host throttling is disabled
 - o SRDF/A session dropped: write pending limit reached. Host throttling is enabled
 - o SRDF/A session dropped: device not ready. Tolerence mode is off
 - o SRDF/A session dropped: device not ready through consistency group
 - o SRDF/A dropped: no SRDF links operational
 - o SRDF/A dropped: timeout in MSC mode
- SRDF CG
 - o RDF CG trip event triggered
- SRDF Link
 - o No SRDF links in an RDF group are operational
 - o Single RDF link in an RDF group is not operational
- SRDF system
 - o Timeout writing to an R2 device; writes pending limit reached

At the time of writing, the event daemon has not yet been certified by Citigroup. This is planned for Q2 2007, targeting SE6.4 as the first release with which it will be supported.

6 2-Site Multi-Session Consistency

This section deals with issues that are specific to an implementation requiring multi-session consistency, and which are over and above those described in the Generic SRDF Configuration Section.

6.1 Overview of MSC

MSC offers the ability to provide SRDF/A replication of multiple RDF groups with I/O consistency across all the groups, whilst still retaining the ability to failover individual RDF groups to deal with, for example, application server failures, without impacting the replication of other applications.

SRDF/A MSC provides consistency protection for an asynchronous-mode consistency group by performing cycle switching and cache recovery operations across all SRDF/A sessions in that group. Both the RDF daemon and SYMAPI contain SRDF/A MSC logic to determine whether to commit or discard data in the cache when SRDF/A sessions have been deactivated.

Consistency here means that the devices act in unison to maintain the integrity of a database distributed across multiple RDF groups within a single Symmetrix, or indeed across multiple Symmetrix. If a source R1 device in the consistency group cannot propagate data to its corresponding target R2 device, SRDF/A MSC suspends data propagation from all R1 devices in the consistency group, halting all data flow to the R2 targets.

MSC cycle-switching and cache recovery operations across all SRDF/A sessions in the consistency group ensure a consistent R2 copy of the database during the time that the interruption occurred. When an interruption occurs, SRDF/A MSC examines the status of SRDF/A sessions and data. Based on the results, MSC either commits the last copy cycle to the R2 side or discards it, converting the track to a remote invalid on the R2 device.

Achieving data consistency across multiple RDF groups requires that the cycle switch process be coordinated, and that the switch occurs during a very brief period (usually a few ms) when no host writes are being serviced. MSC provides a single coordination point to drive the cycle switch process.

Implementing consistency protection involves building a composite group (a grouping of device groups) and enabling the composite group for consistency protection, at which time the group becomes a *consistency group* and is managed by the new RDF daemon.

An interruption that causes deactivation of the SRDF/A sessions and the suspension of the consistency protection can occur in several ways. Deactivation occurs when one or more R1 source devices in a consistency group cannot propagate data to their corresponding R2 target devices. For example:

- RDF links between the R1 and R2 might go down for an extended period of time
- The RDF directors on the R1 side or R2 side might fail

If I/O cannot be propagated from an R1 device in the consistency group to its R2 device, MSC suspends remote mirroring on ALL RDF links in the consistency group, in a manner that honours dependent write I/Os.

At the time of writing, MSC in a concurrent configuration is not yet supported. It is slated to be supported for Open Systems in 2006 and will require a later version of Solutions Enabler (6.1 or perhaps even later)

6.2 High-availability of MSC replication

6.2.1 Overview

It is important to understand that SRDF/A cycle switching for individual RDF groups is controlled by the DMX itself, whereas for consistency-enabled Composite groups (a.k.a. Consistency Groups), switching is controlled by daemons (storrdfd) running on fibre-attached hosts.

EMC have implemented a live-live control system. MSC daemons can be running on multiple servers and the active master for each MSC session can change with each cycle. As a consequence, the MSC service is highly available by design, and cycle switching (i.e. replication) will occur as long as there is at least one storrdfd running on a connected SAN management server. No clustering is required to provide high availability.

However, it is important to note that:

- If no storrdfd daemons are available:
 - o Cycling switching of the MSC-controlled consistency group will cease
 - o The “time R2 is behind the R1” time will increase
 - o IOs will be stored in the production DMX SRDF/A cache (at least until this becomes exhausted)

Once a daemon comes on-line again, it takes over replication control of the MSC group again. Hence, it is vital to ensure that multiple servers are configured for MSC control to ensure a continual replication service.

- The cycle number shown by symrdf query and symstat will vary when an RDF group is under MSC control.

```
# symrdf -cg <cg-name> query -rdfa
```

shows MSC cycle number which is reset to 0 when MSC control was enabled

```
# symrdf -g <dg-name> query -rdfa
```

shows MSC cycle number which is reset to 0 when MSC control was enabled

```
# symstat -type cycle
```

shows the original cycle number for the RDF group prior to MSC being enabled. Hence, different groups can be on different cycle numbers, even if they are under MSC control and switching in step with each other. Slightly confusing, especially as symstat gives different output than `symrdf -g <dg-name> query -rdfa` for a given RDF group.

6.3 Configuring MSC support on management servers

6.3.1 Gatekeeper requirements

One gatekeeper is required per Composite Group to be put under MSC control. In addition, several gatekeepers are required for status querying. If there are insufficient gatekeepers, then the switch time may take longer as operations cannot take place in parallel.

Note that it is possible to dedicate up to 20 Gatekeepers for exclusive use by the storrdfd daemon to ensure that there are plenty available for MSC control. This is achieved using the parameter

```
storrdfd:rdfd_num_dedicated_gks=20
```

in the `/var/symapi/config/daemon_options` file.

6.3.2 High-availability of RDF daemon

MSC replication control is via the RDF daemon storrdfd. High-availability of the MSC replication function is provided by ensuring that there is a minimum of two RDF daemons active at any one time at the source site.

The RDF daemon is enabled with the following entry in `/var/symapi/config/options`

```
SYMAPI_USE_RDFD = ENABLED
```

The daemon can be set to start automatically every time the local host is booted. This is known as installing the daemon. If the Daemon is stopped for some reason (other than deliberately via the storrdfd shutdown command), it can be automatically restarted by an internal Solutions Enabler watchdog mechanism. A combination of the watchdog mechanism and auto-start option above can be used to ensure that the daemon is always running.

The RDF daemon (storrdfd) can be “installed” with the following command.

```
# stord daemon install storrdfd -autostart
```

Installing the daemon ensures that the watchdog daemon restarts the storrdfd if it fails. The `-autostart` ensures that the storrdfd is started at system boot time. A daemon that is shutdown (rather than dies or is killed) is not restarted by the watchdog daemon.

Note that any symrdf command (e.g. symrdf list) will cause a shut down daemon to start up again.

The status of the storrdfd can be checked with the command

```
# stord daemon show storrdfd
```

```
Sarbbunker1# stord daemon show storrdfd
```

```
Daemon State           : Running
Daemon Start Time      : Thu Apr 13 09:18:45 2005
Version                : T6.0-623 (0.43)
Auto-Restart by Watchdog : Enabled

Total Number of Connections : 1
Number of Active Connections : 1
Total Number of Requests    : 0
```

6.4 Use of Composite groups

MSC is controlled via Composite groups. To implement MSC for a set of device groups, it is necessary to:

1. Create the Composite Group enabled for RDF daemon controlled consistency, and populate the Composite Group with the devices from the device groups that are to be bound together from a consistency perspective.
2. Replicate composite group to other SRDF/A management servers
3. Enable the Composite group for consistency i.e. enabling Multi-Session Consistency replication.

6.4.1 Scalability restrictions

For some versions of SE (6.0.x), there is a limit on the number of device groups that can be within a Composite Group for which consistency can be enabled. Experimentation found that the maximum was 18 sessions. EMC later confirmed that the maximum possible is 22 ($\sqrt{512}$), although for any given implementation, this number could be lower. Hence, the maximum number of device groups that can be brought under MSC control on a single DMX was at most 22, often less. This limit also applies the total number of device groups within Composite Groups across the whole DMX. Hence, you can have neither 24 device groups in a single Composite group, nor 4 Composite Groups each of 8 device groups.

EMC state that the above issue is now fixed with SE6.3 and later versions of Enginuity code, but this has not been tested by Citigroup.

6.4.2 Creating the Composite Group

The following is an example of how to setup a composite group called multi that binds together devices in RDF groups 3, 4 and 5. The Source DMX has a SID of 0822.

Put all the devices in RDF groups into a composite group called multi. The `-rdf_consistency` flag places the composite group under the control of the rdf daemon.

```
# symcg create -type rdf1 -rdf_consistency multi
# symcg -cg multi addall dev -rdfg 3 -sid 0822
# symcg -cg multi addall dev -rdfg 4 -sid 0822
# symcg -cg multi addall dev -rdfg 5 -sid 0822
```

Note that it is important to clean up composite groups when no longer needed (using `symcg delete`) as these are stored in the DMX (not the local host as with a device group) and can cause issues if another host attempts to use `symcg` commands. If one does not do this, the `storrdm` daemon keeps seeing groups in the DMX that are no longer being used.

The opposite of `addall` is `rmall`. The following commands will remove the Composite Group `multi`.

```
# symcg -cg multi rmall -rdfg 3
# symcg -cg multi rmall -rdfg 4
# symcg -cg multi rmall -rdfg 5
```

```
# symcg delete multi
```

Note that creating the composite group has no immediate effect on the replication of the individual RDF group cycles.

```
# symcg show multi
```

“RDF Consistency Protection Allowed” entry should be Yes.

At this point “RDF Consistency Enabled” should be “No”.

6.4.3 Synchronizing Composite Group definitions across management server

The definition of each composite group defined for MSC replication must be replicated to all management servers responsible for providing an RDF daemon for MSC control. This is best achieved by using GNS in a 2-site implementation.

GNS provides the ability to store Composite Group definitions in a shared repository located on each Symmetrix array, which then becomes visible to all hosts that are configured to run Solutions Enabler. This allows all GNS-enabled hosts to see the same group definitions across the Symmetrix environment, while sharing real-time updates to group definitions and configurations made by other hosts.

Instructions for enabling GNS can be found in the document “Solutions Enabler 6.x” Solutions Guide (http://gdse.ny.smb.com/GDSE_docs/GDSE-05016-SG.pdf).

6.4.4 Enabling MSC replication

At this point, RDF consistency will not be enabled, but individual RDF group sessions will be active (i.e. cycling under DMX control)

```
# symrdf -cg multi query -rdfa
```

In output from the above command, check “RDFA MSC Info” section. MSC Session status should be “N/A”. Consistency State will also be “N/A”

Then check that under the RDFA Info section for each RDF group, that the Session Status should be “Active”. This indicates that the cycling of that group is under DMX control, rather than MSC control.

Also, at this point at this point “RDF Consistency Enabled” should be “No”.

Now enable consistency between the individual device group devices (i.e. across the entire composite group) and confirm that multi-session consistency is active

```
# symcg -cg multi enable
```

```
# symrdf -cg multi query -rdfa
```

In output from the above command, check “RDFA MSC Info” section. MSC Session status should now be “Active”, although Consistency State will remain “INCONSISTENT” until 2 cycles have completed.

Also check that under the RDFA Info section for each RDF group, that the Session Status is “Active - MSC”. This indicates that the cycling of that group is under MSC control, rather than DMX control (where “Session Status” would be “Active”).

Also, at this point at this point “RDF Consistency Enabled” should be “Yes”.

When MSC is enabled, it will:

- wait until all current cycles have completed. This will result in some individual most session cycles being extended beyond their usual durations
- All groups will then begin cycle switching in unison

Individual session cycle numbers (as shown by symstat) will not be reset and may vary. However, all delta set switches across all the sessions will now begin to occur simultaneously. Symrdf query –rdfa will show the cycle # as initially reset to 0 when MSC is enabled.

During this process, the RDFA MSC Info (observable with # symrdf –cg multi query –rdfa) will change undergo the following changes:

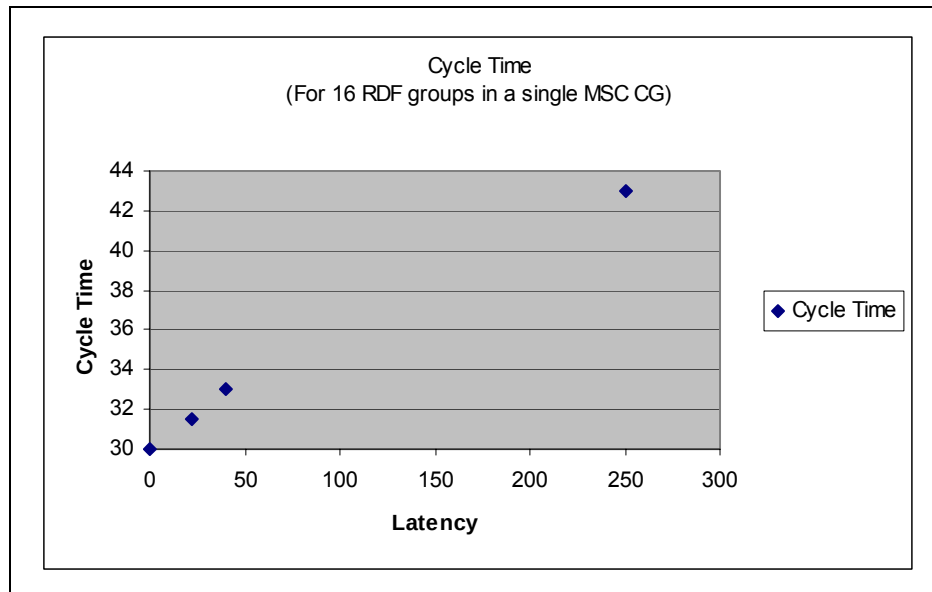
MSC Session	N/A →	Inactive →	Active →	Active
Consistency State	N/A →	Inconsistent →	Inconsistent →	Consistent

MSC replication will stop if:

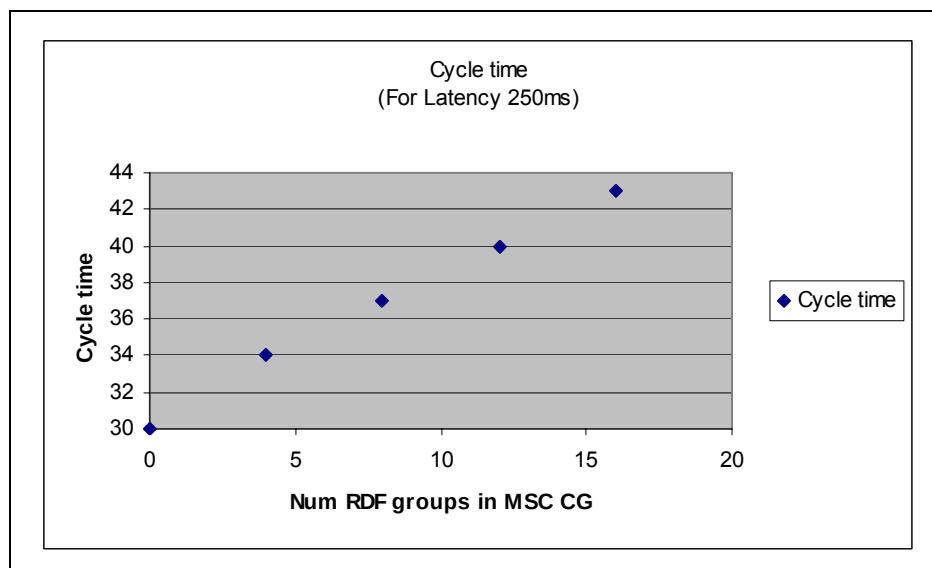
- No daemons are available
- One (or more) of the RDF groups cannot replicate (link down). Replication of the other groups will then be held to ensure consistency.

6.5 Cycle time under MSC

When RDF groups are under Multi-Session Consistency (MSC) control, cycle time increases as a function of latency. This is NOT due to lack of bandwidth but due to the overhead of the latency in controlling the SRDF/A delta set switching process. The following graph shows an example of how cycle time extends for MSC-controlled replication as a function of latency. The minimum cycle time was set to 30 seconds.



Note that the cycle time is also a function on the number of RDF groups under MSC control, and not simply a singular function of latency. The following graph demonstrates how cycle time changes as a function of the number of RDF groups under MSC control for a fixed latency (in this case 250ms).



Hence, use of MSC will extend cycle times beyond the preset minimum in most implementations with any significant latency.

7 Concurrent 3-site SRDF/A configuration

7.1 General comments

This section deals with issues that are specific to a concurrent SRDF implementation and which are over and above those described in the Generic SRDF Configuration Section. It gives Best Practice for how to configure device groups at each site depending on the synchronous target R2 configuration and desired synchronous leg failover functionality.

If Concurrent SRDF is being configured to support SRDF/Star, then much of the Best Practice in this section will be superseded by the Star-specific Best Practice in the document “SRDF/Star for Open Systems” Solution Guide.

7.2 Licensing

For concurrent SRDF, an additional SRDF license must be purchased for the source DMX so that you have both SRDF/S and SRDF/A licensed.

See the SRDF/Star Solutions Guide for licensing requirements for SRDF/Star.

7.3 Device group definitions

This section does not apply to SRDF/Star configurations, as Device Groups are not used with SRDF/Star.

Defining the device groups to be used to control an SRDF/A implementation is more complex than with a simple 2-site synchronous SRDF implementation. This is due to functionality limitations with the EMC technology.

Limitation A: Devices can only be in one device group – BCV control issue

An additional BCV is required at the SRDF/A target site for resynchronisation protection. For sites that build their R2s from an unmirrored R2 STD device and a BCV (permanently established) for COB use, then there is now an issue with controlling both sets of BCVs for the same standard using a single device group. Multiple device groups cannot be used as the R2 STD can only be on one group.

As a consequence, it is recommended that the Async R2 always be a mirrored pair (R2=M1+M2) and that the SRDF/A PIT BCV also be used for COB testing. This configuration also ensures that the R2 is protected at all times, even during COB testing. It does, however, require co-ordination on the use of the SRDF/A target site BCVs.

Limitation B: remote BCVs cannot be defined in two different RDF groups within a single device group

It is not possible to have remote BCVs defined in a device group for two different RDF groups. Hence, in a Concurrent SRDF configuration, it is not possible to control remote BCVs at both

target sites from the source site. See Section 7.3.3 & 7.3.4 for diagrams of configurations for which this would be an issue.

Limitation C: GNS remote mirroring is not supported at present with devices running concurrent SRDF

Limitation D: Devices can only be in one device group – subset failover issue

Subset failover (of a single server) down the synchronous leg requires a device group that only contains the devices owned by that server. SRDF/A failover must cover all disks in the device group. Hence, the scope of the device groups required for SRDF/A and SRDF/S failover in a concurrent configuration differ. Multiple device groups cannot be used as the R1 STD can only be on one group. *Note that subset failover on the synchronous leg of a concurrent SRDF configuration is not recommended either by EMC nor GDSE, but is necessarily being supported to meet business requirements.*

Furthermore, there are 5 different configurations that must be supported; one for 2-site and four permutations for 3-site implementations:

1. 2-site SRDF/A with the R2 comprised of a mirrored pair of disks (M1+M2)
 - Configuration impacted by Limitation A (although only where COB testing is done on a BCV).
2. 3-site SRDF/A with the Sync target R2 comprised of a mirrored pair of disks (M1+M2). Subset failover to the synchronous target site is not required.
 - Configuration impacted by Limitation C
3. 3-site SRDF/A with the Sync target R2 comprised of a mirrored pair of disks (M1+M2). Subset failover to the synchronous target site is required.
 - Configuration impacted by Limitations C & D
4. 3-site SRDF/A with the Sync Target R2 comprised of an unmirrored disk plus a BCV (M1+BCV) and async R2 comprised of a mirrored pair of disks (M1+M2). BCVs used for COB testing. Subset failover to the synchronous target site is not required.
 - Configuration impacted by Limitations A, B & C
5. 3-site SRDF/A with the Sync Target R2 comprised of an unmirrored disk plus a BCV (M1+BCV) and async R2 comprised of a mirrored pair of disks (M1+M2). BCVs used for COB testing. Subset failover to the synchronous target site is required.
 - Configuration impacted by Limitations A, B, C & D

Hence, there is no single clean solution to cover all of these configurations. A slightly different recommendation is required for each configuration, but this best practice has been formulated to minimize the differences between the configurations as much as possible. For example, with

all these configurations, the SRDF/A resynchronisation script can be run from either the source or SRDF/A target site, and the same process used in all configurations in all cases.

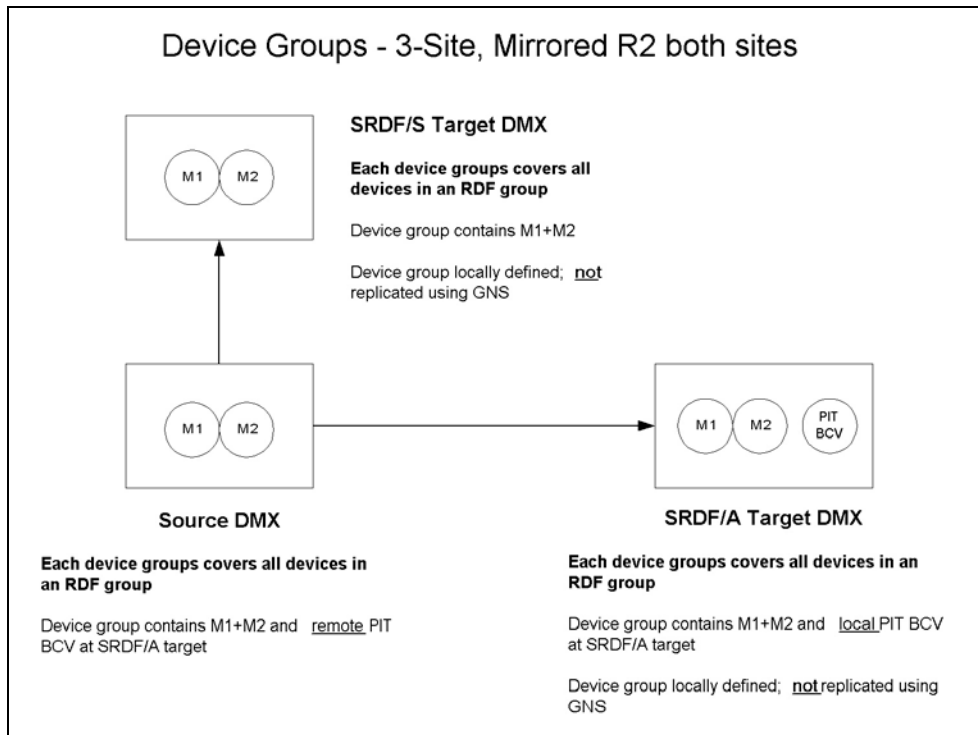
Use of device files has deliberately been avoided.

The following table (following page) attempts to summarize the device group configuration best practice. It defines what should be in the device group at each site and whether GNS remote mirroring can be used for device group replication from the source to the target sites. GNS of course can be used between redundant SAN management servers on the same site.

	Configuration	Device Group Contents			Can GNS replicate to target sites?
		Source DMX	SRDF/S target	SRDF/A target	
1	2-site R2=M1+M2	All R1s in RDF group + remote SRDF/A PIT BCV	N/A	All R2s in RDF group + local PIT BCV	Yes
2	3-site Sync R2=M1+M2 Async R2=M1+M2 No subset failover	All R1s in RDF group + remote SRDF/A PIT BCV	All R2s in RDF group	All R2s in RDF group + local SRDF/A PIT BCV	No* /**
3	3-site Sync R2=M1+M2 Async R2=M1+M2 With subset failover	All R1s in RDF group + remote SRDF/A PIT BCV	Separate DG for R2 devices belonging to each server	All R2s in RDF group + local SRDF/A PIT BCV	No*/**/**
4	3-site Sync R2=M1+BCV Async R2=M1+M2 No subset failover	All R1s in RDF group + remote SRDF/A PIT BCV (also used for COB testing)	All R2s in RDF group + local COB BCV	All R2s in RDF group + local SRDF/A PIT BCV	No */**
5	3-site Sync R2=M1+BCV Async R2=M1+M2 With subset failover	All R1s in RDF group + remote SRDF/A PIT BCV (also used for COB testing)	Separate DG for R2 devices + local COB BCV belonging to each server	All R2s in RDF group + local SRDF/A PIT BCV	No */**/**

- * GNS remote mirroring not supported for concurrent SRDF
- ** Different STDs specified at Sync target site
- *** Different (or no) BCV specified at Sync target site.

7.3.1 Configuration 2: 3 site SRDF/A with mirrored R2s. No subset failover



Configuration:

Config 2: 3-site

Sync R2=M1+M2

Async R2=M1+M2

No subset failover

Device Group contents:

- Source DMX All R1s in RDF group + remote SRDF/A PIT BCV
- SRDF/S Target All R2s in RDF group
- SRDF/A Target All R2s in RDF group + local SRDF/A PIT BCV

Device Group Replication

GNS replication to remote sites not supported in concurrent configuration. O.K. to local replicas.

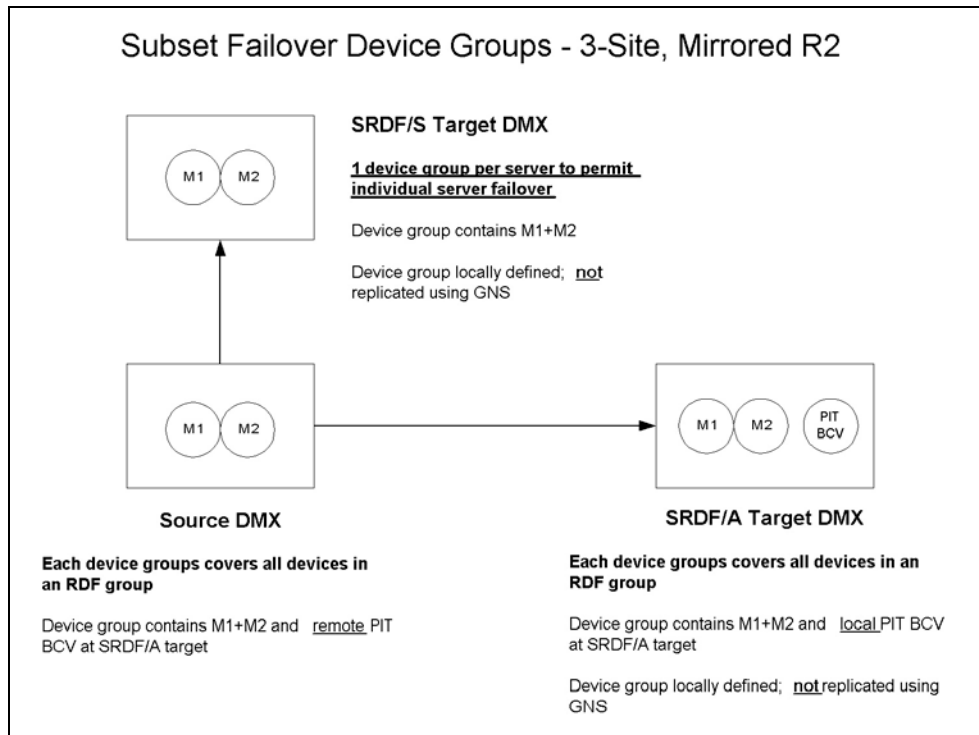
This is the first of 2 possible 3-site configurations for sites where COB testing is undertaken by invoking COB and running from the R2. In such cases, the BCV at the Async site is only used for SRDF/A resynchronisation.

BCVs can be controlled from either async site, but for entire RDF group only.

Note, however, that this configuration does not permit failover to the synchronous site at the server level; only failover of whole RDF groups (applications) is supported. However, this configuration is simpler to manage than the subset failover alternative. All management can be undertaken from the source site.

Furthermore, failover of an entire RDF group is consistent with EMC best practice and compatible with future developments of concurrent SRDF (SRDF/STAR).

7.3.2 Configuration 3: 3 site SRDF/A with mirrored R2s. With subset failover



Configuration:

Config 3: 3-site
 Sync R2=M1+M2
 Async R2=M1+M2
 With subset failover

Device Group contents:

- Source DMX All R1s in RDF group + remote SRDF/A PIT BCV
- SRDF/S Target **Separate DG for R2 devices belonging to each server**
- SRDF/A Target All R2s in RDF group + local SRDF/A PIT BCV

Device Group Replication

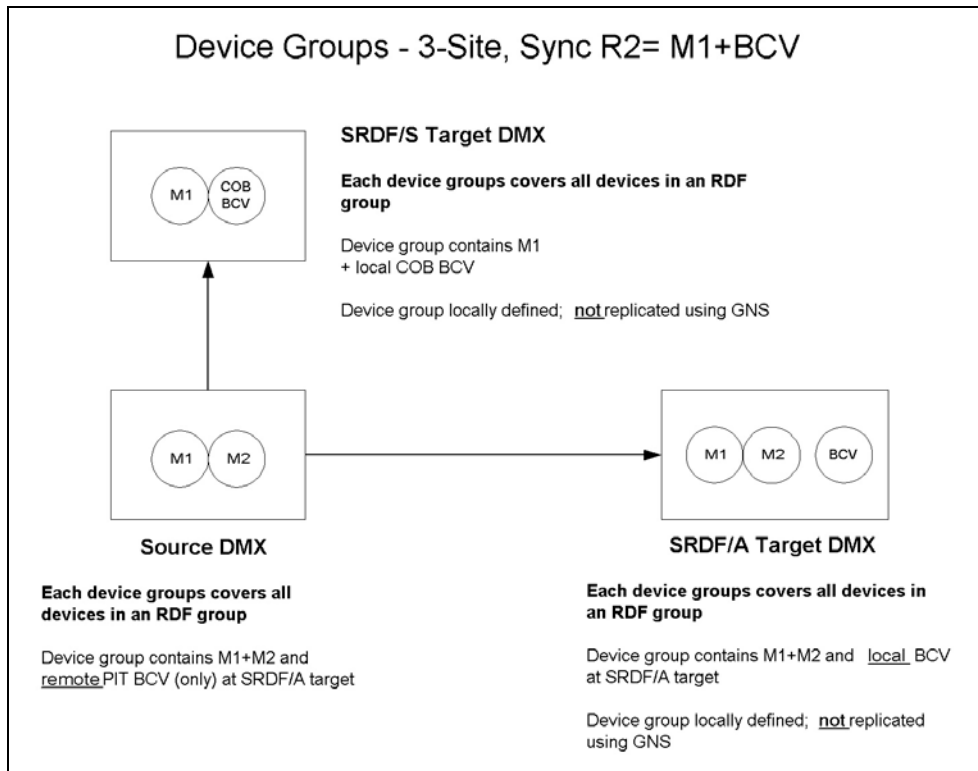
GNS replication to remote sites not supported in concurrent configuration. O.K. to local replicas.

This configuration is similar to the previous one, but also permits failover to the synchronous site at the server level.

This is more flexible than the previous configuration, but is less compatible with future developments of concurrent SRDF/A.

Note that subset failover on the synchronous leg of a concurrent SRDF configuration is not recommended either by EMC nor GDSE, but is necessarily being supported to meet business requirements.

7.3.3 Configuration 4: 3-site SRDF/A with unmirrored Sync R2 + COB BCV. No subset failover.



Configuration: Config 4: 3-site
 Sync R2=M1+BCV; Async R2=M1+M2
 No subset failover

Device Group contents:

- Source DMX All R1s in RDF group + remote SRDF/A PIT BCV (also used for COB testing)
- SRDF/S Target All R2s in RDF group + local COB BCV
- SRDF/A Target All R2s in RDF group + local SRDF/A PIT BCV

Device Group Replication GNS replication to remote sites not supported in concurrent configuration. O.K. to local replicas.

This is the first of 2 possible 3-site configurations for sites where COB testing is undertaken by running from a BCV. The BCV at the Async site will be used for both COB testing and SRDF/A recovery and usage will need to be co-ordinated.

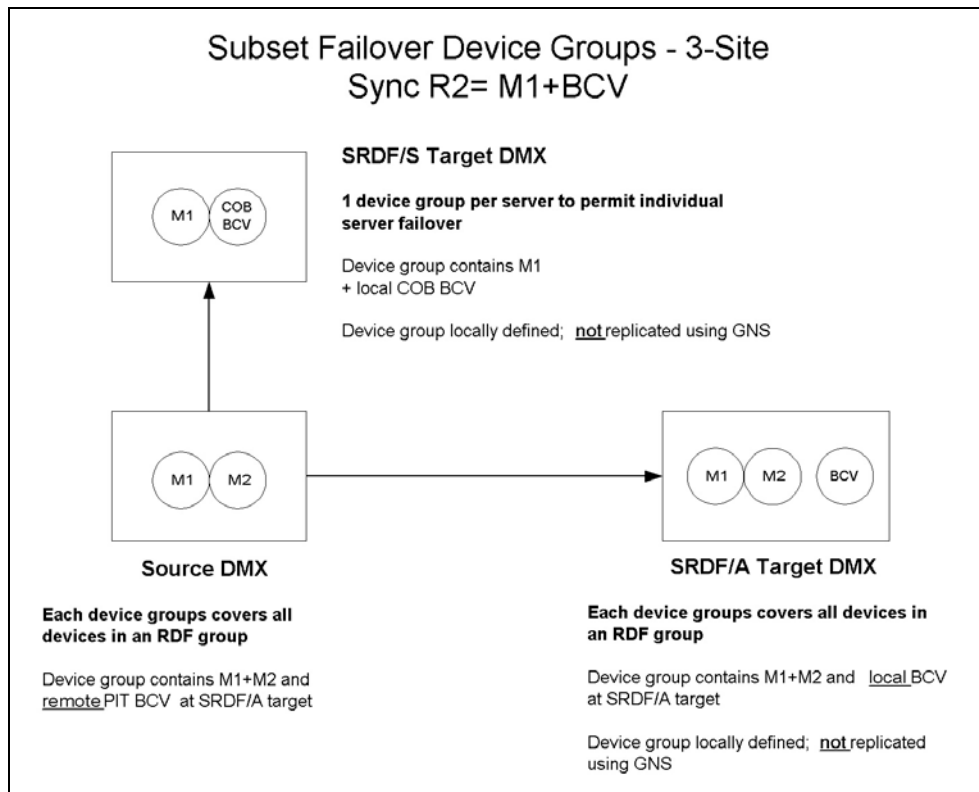
BCVs can be controlled from either async site, but for entire RDF group only.

Note, however, that this configuration does not permit failover to the synchronous site at the server level; only failover of whole RDF groups (applications) is supported. However, this configuration is simpler to manage than the subset failover alternative.

All SRDF management can be undertaken from the source site.

Furthermore, failover of an entire RDF group is consistent with EMC best practice and compatible with future developments of concurrent SRDF (SRDF/STAR).

7.3.4 Configuration 5: 3-site SRDF/A with unmirrored Sync R2 + COB BCV. With subset failover.



Configuration:	Config 5: 3-site Sync R2=M1+BCV Async R2=M1+M2 With subset failover
Device Group contents:	
• Source DMX	All R1s in RDF group + remote SRDF/A PIT BCV (also used for COB testing)
• SRDF/S Target	<u>Separate DG for R2 devices and local BCVs belonging to each server</u>
• SRDF/A Target	All R2s in RDF group + local SRDF/A PIT BCV
Device Group Replication	GNS replication to <u>remote sites</u> not supported in concurrent configuration. O.K. to local replicas.

This configuration is similar to the previous one, but also permits failover to the synchronous site at the server level. This is more flexible than the previous configuration, but is less compatible with future developments of concurrent SRDF/A i.e. SRDF/STAR. It is anticipated that this is the configuration that will be used in NAM and Tokyo.

Note that subset failover on the synchronous leg of a concurrent SRDF configuration is not recommended either by EMC nor GDSE, but is necessarily being supported to meet business requirements.

7.4 Worked example of a non-Star 3-site RDF group configuration

This example refers to a non-Star Concurrent SRDF setup. For an example of how to setup SRDF/Star RDF groups, see the SRDF/Star Solution Guide.

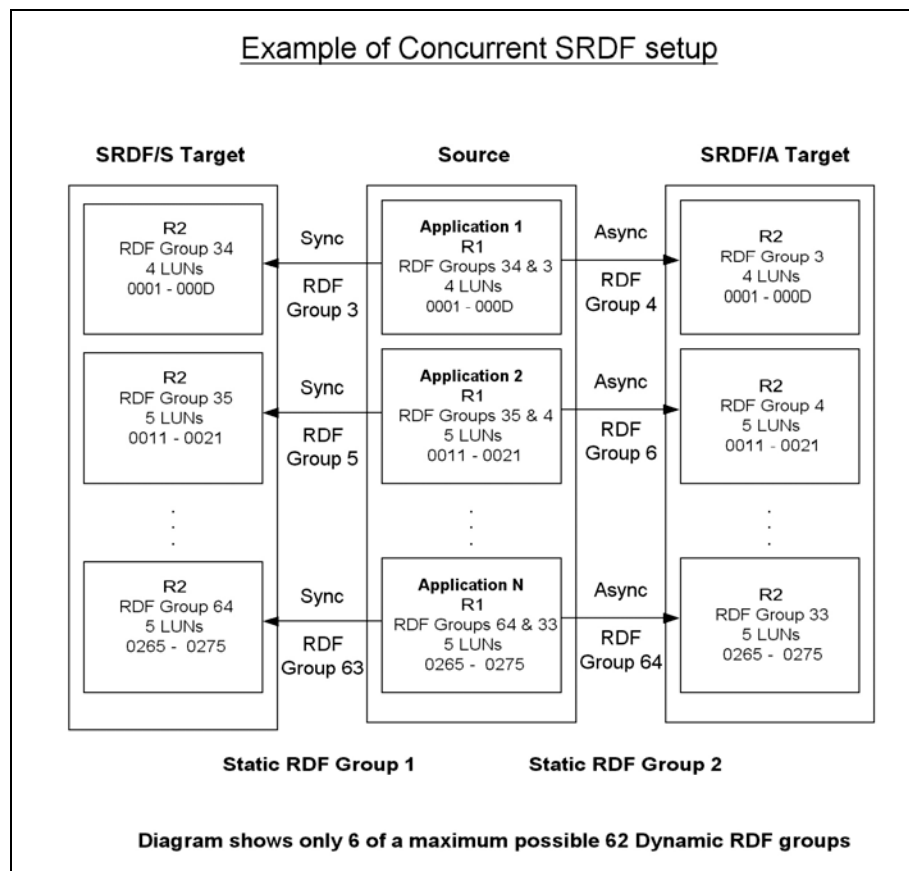
The fundamental difference between 2-site and 3-site SRDF/A implementations is that the 3-site configuration has an additional SRDF/S replication session for each source volume in addition to the SRDF/A session.

Hence, each set of disks in an RDF group to be used for SRDF/A must also be members of an additional RDF group which will be used for SRDF/S replication to the local COB DMX.

The maximum number of RDF groups that can be supported on a DMX is 64. 2 RDF groups (usually groups 1 and 2) will be used for the static RDF links that need to exist between each pair of DMX prior to creation of Dynamic RDF groups. Hence, the maximum number of Concurrent SRDF sessions is 31.

The mapping of RDF groups to RA ports for concurrent SRDF has already been covered in the Planning section of this document and is not repeated here.

The diagram below summarizes a fictitious Concurrent SRDF implementation. You may find it useful to draw up your implementation in this manner.



RDF groups 1 and 2 are used for the underlying static RDF groups between each pair of DMX. Hence, the Dynamic RDF groups start at 3. It is recommended that Synchronous legs be given odd RDF numbers, and the corresponding Async leg the following even number. This will help avoid manipulating the wrong leg in a concurrent SRDF environment. For example, for an application with RDF group 5 for the SRDF/S leg, the corresponding SRDF/A leg should use RDF group 6.

Each set of LUNs is a member of **two** RDF groups

In the following example, the entire application is to be failed over as a single logical entity, as per Configuration 2. Configuration of subset failover down the Synchronous leg of a subset of an RDF group is covered in a later section.

Concurrent SRDF configuration is comprised of the following main steps:

1. Create Synchronous Leg RDF group
2. Populate with LUNs, establish and create device group
3. Create Asynchronous Leg RDF group
4. Populate with LUNs and establish
5. Check concurrent SRDF set up
6. Enable SRDF/A replication on the appropriate leg
7. Add BCVs into device group. Establish BCVs.
8. Ensure that device group definitions at remote sites are up to date

The following example is for the Application 1 in the above diagram. RDF group 3 has been allocated to Local RAs 3A/14B (on source DMX 0822) and remote RAs 3A/14B (on Async target DMX 1497). RDF group 34 has been allocated local RAs 3C/14D and remote RAs (on Sync target DMX 1135) of 3A/14B. The device pairsfile called pairsfile_App1 contains meta devices 0001 through 000D (meta heads only). The local device group is to be called App1.

1. Create Synchronous Leg RDF group

Do not specify a link limbo timer setting. SRDF/S RDF groups should be left with the default 10 second value.

```
# symrdf addgrp -label DYN3 -rdfig 3 -sid 0822 -dir 3C -remote_rdfg 3 -
remote_sid 1135 -remote_dir 3A
```

```
# symrdf modifygrp -label DYN3 -rdfig 3 -sid 0822 -dir 14D -remote_dir
14B
```

2. Populate with LUNs, establish and create device group

Use -g option to automatically create the device group.

```
# symrdf -file pairsfile_App1 -sid 0822 -rdfig 3 createpair -type rdf1
-establis -g App1
```

3. Create Asynchronous Leg RDF group

SRDF/A RDF groups should have their link limbo timer set to 60 seconds.

```
# symrdf addgrp -label DYN4 -rdfg 4 -sid 0822 -dir 3A -link_limbo 60
-remote_rdfg 4 -remote_sid 1497 -remote_dir 3A -rem_link_limbo 60
```

```
# symrdf modifygrp -label DYN4 -rdfg 4 -sid 0822 -dir 14B -remote_dir
14B
```

4. Populate with LUNs and establish

Do not use `-g` when populating the second SRDF leg as the device group already exists.

```
# symrdf -file pairsfile_App1 -sid 0822 -rdfg 4 createpair -type rdf1
-establish
```

5. Check concurrent SRDF set up

Use the `"-rdfg all"` option to ensure that both RDF legs are shown.

```
# symrdf -g App1 query -rdfg all
```

```
Device Group (DG) Name      : App1
DG's Type                   : RDF1
DG's Symmetrix ID          : 000187700822
Remote Symmetrix ID        : 000187721497
RDF (RA) Group Number      : 4
Remote Symmetrix ID        : 000187721135
RDF (RA) Group Number      : 3
```

Source (R1) View					Target (R2) View					MODES	
ST					LI						
A					N						
T					K						
Standard	Dev	E	R1 Inv	R2 Inv	Dev	E	R1 Inv	R2 Inv	MDA	RDF Pair	STATE
Logical	Device	Dev	Tracks	Tracks	Dev	Dev	Tracks	Tracks			
DEV001	0001	RW	0	0	RW	0001	WD	0	0	S..	SyncInProg
		RW	0	0	RW	0001	WD	0	0	S..	Synchronized
DEV002	0005	RW	0	0	RW	0005	WD	0	0	S..	SyncInProg
		RW	0	0	RW	0005	WD	0	0	S..	Synchronized
DEV003	0009	RW	0	0	RW	0009	WD	0	0	S..	SyncInProg
		RW	0	0	RW	0009	WD	0	0	S..	Synchronized
DEV004	000D	RW	0	0	RW	000D	WD	0	0	S..	SyncInProg
		RW	0	0	RW	000D	WD	0	0	S..	Synchronized
Total											
Track(s)			0	0	0			0	0		
MB(s)			0.0	0.0	0.0			0.0	0.0		

Legend for MODES:

```
M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
D(omino)             : X = Enabled, . = Disabled
A(daptive Copy)      : D = Disk Mode, W = WP Mode, . = ACp off
```

This show that each source device has two RDF Pair states. In this case, one is synchronized and the other is Synchronizing. Unfortunately, the output does not clearly show which RDG group each entry is associated with.

Each RDF group can be separately interrogated. When in a concurrent SRDF configuration is necessary to specify which RDF group the command applies to.

For the Async leg

```
# symrdf -g App1 query -rdfg 4
```

For the Sync leg

```
# symrdf -g App1 query -rdfg 3
```

6. Enable SRDF/A replication on the Async leg

The enable command enables consistency between devices within the device group.

```
# symrdf -g App1 -rdfg 4 set mode async -noprompt
# symrdf -g App1 -rdfg 4 enable
```

7. Add BCVs into device group. Establish BCVs.

BCVs can be added to groups either using existing local tools used at your site for BCV management or using the scripts provided as part of this Solution. A script called **add_PIT_BCV** (see Appendix B for further details) is available to automate this process using a simple STD-to-BCV input-mapping file. A script called **rm_PIT_BCV** is also available to backout the additions.

Alternatively, this can be achieved manually using the syntax in the example below.

On the management server at the source site, the BCVs in the target DMX should be associated as remote BCVs.

On the management server at the target site, the BCVs in the target DMX should be associated as local BCVs.

Example

SRDF/A RDF group 3 contains 4 standard volumes. The group associated with this RDF group is App1. The PIT BCVs to be associated with the R2 volumes are 02AD, 02B1, 02B5, 02B9. The `-rdf` flag indicates that these are remote BCVs.

```
# symbcv -g App1 -sid 0822 associate dev 02AD RBCV02AD -rdf -rdfg 4
# symbcv -g App1 -sid 0822 associate dev 02B1 RBCV02B1 -rdf -rdfg 4
# symbcv -g App1 -sid 0822 associate dev 02B5 RBCV02B5 -rdf -rdfg 4
# symbcv -g App1 -sid 0822 associate dev 02B9 RBCV02B9 -rdf -rdfg 4

# symmir -g App1 establish -full -opt_rag -rdf -nop
```

8. Ensure that device group definitions at remote sites are up to date

GNS remote mirroring of device groups that contain Concurrent SRDF device is not supported (they do not get replicated). Hence, it will be necessary to manually define the device groups at each of the remote sites.

The author is aware that the pre-GNS procedures for maintaining remote device groups differ enormously from site to site (manual entry, export/edit/import of device group definitions, scripts that process a device list, etc). Sites should continue to use these processes for concurrent SRDF device group management.

7.5 Multi-Session Consistency with Concurrent SRDF

Multi session consistency (MSC) is only supported for Concurrent SRDF when used as part of an SRDF/Star environment. See the SRDF/Star Solutions Guide http://gdse.ny.ssmb.com/GDSE_docs/GDSE-06050-SG.pdf for further details.

8 Operational procedures

*These operational procedures refer to 2-Site SRDF/A and 3-Site Concurrent SRDF. For operational procedures for SRDF/Star, refer to the “**SRDF/Star for Open Systems**” Solutions Guide. Similarly, the operational scripts referenced in this section in general do not apply to SRDF/Star implementations.*

This section covers operational issues that are likely to be encountered on a regular basis by the SAN management team (or SAN BAU team) after initial configuration has been completed.

For each activity, best practice examples of each process are given. In addition, for most activities, scripts are being provided that will automate the processes.

Note, however, that some scripts have certain necessary assumptions built into them which means that they may not be fully compatible with all existing local practice and procedures. Also there may be the need for some additional manual steps to be undertaken, depending on local practice. In particular there may be issues relating to:

- Device group management (both creation and “replication” of device group definitions to local and remote SAN management servers).
- Formation of synchronous target R2s. These scripts are designed to work with mirrored R2s. If R2s are comprised of unmirrored R2 + BCV, then additional steps will be required to managed these additional BCVs.
- Local naming conventions for logical devices and device groups.

Nevertheless, they do provide fully working scripts which can be used "as is" where the various assumptions are valid. Alternatively, they can be taken and have minor modifications made to them to work within your environment.

These scripts and procedures assume that the SRDF/A target machine R2 is a mirrored pair. They have been tested against an environment where the synchronous target R2 is also a mirrored pair. This will not be the case in all regions and some additional changes may be necessary to deal with any additional management of synchronous target BCVs.

It is assumed that the reader is familiar with existing synchronous SRDF and TimeFinder operations, and related activities such as device group manipulation. **This document only deals with new SRDF/A specific issues.**

8.1 Storage provisioning with multi-session SRDF/A

8.1.1 Overview of issues

Historically, most SRDF/S implementations within Citigroup have used a single static RDF group between production (R1) and corresponding COB (R2) Symmetrix. R1/R2 pairs are typically all members of the same RDF group and will typically be pre-established before the storage is allocated out to end-users.

SRDF/A offers a new set of challenges. In a typical deployment:

- Each application will have its own RDF groups for SRDF/A (two RDF groups for a 3-site concurrent SRDF implementation).
- As devices are allocated to an application (either initial allocation or through growth in storage requirements for an existing application), they will now also need to be added to the appropriate RDF group (and device groups) and synchronized. Corresponding SRDF/A PIT BCVs will also need to be configured as the SRDF/A target site.
- A procedure will now also be needed to remove devices from an RDF group when they are no longer needed by the application.

Each region has different procedures and scripts for managing device groups and hence it is not possible to provide a single script that will meet everyone's needs. Instead the steps required are shown, and these can be worked into local scripts as required. Alternatively, it may be possible for local practices and procedures to be changed to use the scripts provided as is.

8.1.2 Creating a dynamic RDF group

If the storage is for an existing SRDF/A replicated application, there is no need to create a new RDF group.

If this is a new application that is moving to SRDF/A usage, then new RDF group(s) will be required for use with these applications. The syntax for creating RDF groups for both 2-site SRDF/A and 3-site concurrent SRDF has already been discussed in Sections 5 (Generic SRDF/A) and Section 7 (Concurrent SRDF/A) and are not covered here.

8.1.3 Setting RDF group parameters

Parameters specific to the RDF group may need to be changed from default (e.g. session priority). This is optional.

Configuration of RDF group parameters has already been covered in Section 5.4.

8.1.4 Adding devices to a dynamic RDF group – 2-site implementation

The process for adding the initial set of devices for an application into an RDF group has already been discussed in Sections 5 and 7. For additional devices that are allocated to an application at a later time, it will be necessary to:

1. Disable consistency for the RDF group
2. Set the replication mode to adaptive copy
3. Add the new devices into the RDF group and establish RDF pair
4. Add the new devices into the local device group
5. Set the replication mode back to asynchronous and enable consistency
6. Add the remote SRDF/A PIT BCV into the device group

7. Establish the new SRDF/A PIT BCV.
8. Update device groups at remote site.

A script called **2site_add_device** is available to automate the process for multiple devices. See Appendix B for further details.

The following example shows the steps involved for a 2-site SRDF/A implementation. The following assumes the Async R2 is a mirrored standard.

The example shows the addition of a single device 0005 to an application. Both the R1 and R2 have Volume ID 0005. The application uses RDF group 3. The device group containing existing devices for this application is DG03. The SRDF/A point-in-time BCV that is nominated for use with STD 0005 is BCV 02B1. The SID of the source DMX is 0822.

First, a pairsfile for the new device must be created of the following form:

```
# cat pairsfile_DYN03_0005
0005 0005
```

It is then necessary to change the replication mode for RDF group 3 from asynchronous mode as it is not possible to add devices to an SRDF/A-enabled RDF group. Suspending replication is not sufficient.

Change mode to adaptive copy. Disable consistency first.

```
# symrdf -g DG03 disable -nop
# symrdf -g DG03 set mode acp_wp -nop
```

Add new device to RDF group and device group

```
# symrdf -file pairsfile_DYN03_0005 -sid 0822 -rdrg 3 createpair -type rdf1 -establish -nop
# symld -g DG03 add dev 0005
```

Put RDF group back into async mode and enable consistency

```
# symrdf -g DG03 set mode async -nop
# symrdf -g DG03 enable -nop
```

Add the nominated SRDF/A PiT BCV to the device group and just establish the new PiT BCV

```
# symbcv -g DG03 -sid 0822 associate dev 02B1 RBCV02B1 -rdf
# symmir -g DG03 attach DEV002 bcv ld RBCV02B1 -rdf -nop
# symmir -g DG03 establish -full DEV002 bcv ld RBCV02B1 -rdf -nop
```

Confirm RDF and TimeFinder establishes have completed

```
# symrdf -g DG03 query
# symmir -g DG03 query -rdf
```

If GNS is enabled (recommended), device groups on other local (redundant) SAN management servers will be updated. If not, then they will need to be manually updated.

If GNS remote mirroring is enabled (recommended), remote device groups will be updated automatically. If not, they will need to be updated manually.

8.1.5 Adding devices to a dynamic RDF group – Non-Star 3-site implementation

*The following procedure refers to a **non-Star** Concurrent SRDF configuration. See the “SRDF/Star for Open Systems” Solution Guide for the process for that type of configuration.*

The process for adding a device in a concurrent environment requires several additional steps:

1. Add device to synchronous leg RDF group and establish pair
2. Update device group at Synchronous target site

Also, in a concurrent RDF environment, it is mandatory to always define which RDF group the symrdf command must apply to using the `-rdfg <rdf-group-number>`.

Also symrdf query commands on a device group can use the “`-rdfg all`” option to ensure that information on both replication legs is shown, if required.

A script called **3site_add_device** is available to automate the process for multiple devices. See Appendix B for further details.

The following example is identical to the 2-site example in the previous subsection, except that the devices are now in a concurrent SRDF configuration. The RDF group for the synchronous leg is 64 in this example. The SID of the local DMX is 0822. It assumes that the Sync Target R2 is mirrored (M1+M2).

Change mode on asynchronous leg to adaptive copy. Disable consistency first.

```
# symrdf -g DG03 disable -rdfg 3 -nop
# symrdf -g DG03 set mode acp_wp -rdfg 3 -nop
```

Add new device to async RDF group 3 and device group

```
# symrdf -file pairsfile_DYN03_0005 -sid 0822 -rdfg 3 createpair -type rdf1 -establish -nop
# symld -g DG03 add dev 0005
```

Put async RDF group back into async mode

```
# symrdf -g DG03 set mode async -rdfg 3 -nop
# symrdf -g DG03 enable -rdfg 3 -nop
```

Add new device to sync RDF group 64

```
# symrdf -file pairsfile_DYN03_0005 -sid 0822 -rdfg 64 createpair -type rdf1 -establish -nop
```

Add the nominated SRDF/A PiT BCV to the device group and just establish the new PiT BCV. The rdf group must be specified for the associate command to indicate down which RDF leg the remote BCV can be found.

```
# symbcv -g DG03 -sid 0822 associate dev 02B1 RBCV02B1 -rdfg 3 -rdf
```

```
# symmir -g DG03 attach DEV002 bcv Id RBCV02B1 -rdf -nop  
# symmir -g DG03 establish -full DEV002 bcv Id RBCV02B1 -rdf -nop
```

Confirm RDF and TimeFinder establishes have completed

```
# symrdf -g DG03 query -rdfg all  
# symmir -g DG03 query -rdf
```

Update of device groups at remote sites will need to be done manually as GNS device group mirroring to a remote site does not work for devices that are running concurrent SRDF. GNS can, however, be used to replicate to the other local SAN management servers,

At the SRDF/A target site, the BCVs should, of course, be added as local BCVs. Device groups at the target sites must of course be of type R2.

For example, for the change in the previous example, the following steps would be required to update remote device groups. It is assumed that the R1 and R2 have the same volume ID and that the device group is of the same name. This may not be the case as local naming conventions may differ.

For example, to manually update device groups:

Manually add standard to SRDF/A target site device group on remote SAN mgt server
symld -g DG03 add dev 0005

Manually add standard to SRDF/S target site device group on remote SAN mgt server
symld -g DG03 add dev 0005

Manually add local BCV to target site device group on remote SAN mgt server
symbcv -g DG03 -sid 0822 associate dev 02B1 BCV02B1 -rdfg 3

8.1.6 Removing devices from a dynamic RDF group – 2 site implementation

Less frequently, it will be necessary to remove LUNs from an application server. The following example shows the steps involved for a 2-site SRDF/A implementation. Note that this does not cover any data erasure requirements that might be required before the LUN can be re-used as this is outside the scope of the present document. It assumes that the STD volume has already been removed (via LUN masking) from the application host.

Note that the PiT BCV relationship at the target site does not strictly need to be removed, as it will need to be recreated if the LUN is reallocated. However, the steps required are shown here as cleaning up the BCV relationship will result in a more consistent configuration on your DMX.

1. Split PiT BCV from STD to be removed
2. Cancel relationship between PiT BCV and STD
3. Remove BCV from device group
4. Disable consistency and suspend asynchronous replication
5. Remove STD from device group
6. Delete RDF pairing relationship
7. Resume SRDF replication and enable consistency
8. Update device group at target site

A script called **2site_rm_device** is available to automate the process for multiple devices. See Appendix B for further details.

The following example shows the steps involved. Note that if your sync target R2s are comprised of an unmirrored standard and a BCV, then additional steps will be required. The following assumes the sync target R2 is a mirrored standard.

The example shows the steps relating to SRDF/A for the removal of a single device 0005 from an application server. LUN masking is not shown. Both the R1 and R2 have Volume ID 0005. The application uses RDF group 3. The device group this application is DG03. The logical device name of volume 0005 in device group DG03 is DEV002. The SRDF/A point-in-time BCV that is used with STD 0005 is BCV 02B1 (logical device name RBCV02B1).

Split the BCV, cancel relationship with standard and remove BCV from device group

```
# symmir -g DG03 split DEV002 bcv ld RBCV02B1 -rdf -nop
# symmir -g DG03 cancel DEV002 bcv ld RBCV02B1 -rdf -nop
# symbcv -g DG03 disassociate ld RBCV02B1 -rdf
```

Remove device from RDF group 3 and device group DG03. This can be done with the group still in asynchronous mode, but suspended.

```
# symrdf -g DG03 disable -nop
# symrdf -g DG03 suspend -nop
# symld -g DG03 remove DEV002 -force
# symrdf -file pairsfile_DYN03_0005 -sid 0822 -rdfig 3 deletpair -nop
```

Resume SRDF/A replication for remaining device in RDF group 3

```
# symrdf -g DG03 resume -nop
# symrdf -g DG03 enable -nop
```

Update device groups on other local SAN mgt. servers using GNS or via manual intervention.

Update device groups at target site either automatically via GNS remote mirroring or via manual intervention.

8.1.7 Removing devices from a dynamic RDF group – non-Star 3-site concurrent implementation

*The following procedure refers to a **non-Star** Concurrent SRDF configuration. See the “SRDF/Star for Open Systems” Solution Guide for the process for that type of configuration.*

The process for a non-Star 3-site implementation is almost identical to the 2-site. The only additional steps involve the removal of the device from the synchronous RDF group immediately after you remove it from the asynchronous RDF group, and update of the target site device groups.

Also, as this is a concurrent configuration, it is necessary to always define which RDF group is to be operated on with symrdf operations.

The following example shows the steps involved. Note that if your sync target R2s are comprised of an unmirrored standard and a BCV, then additional steps will be required. The following assumes the sync target R2 is a mirrored standard.

A script called **3site_rm_device** is available to automate the process for multiple devices. See Appendix B for further details.

The following example shows the steps relating to SRDF/A for the removal of a single device 0005 from an application server. LUN masking is not shown. Both the R1 and R2 have Volume ID 0005. The application uses RDF group 3 for the async leg and 64 for the sync leg. The device group this application is DG03. The logical device name of volume 0005 in device group DG03 is DEV002. The SRDF/A point-in-time BCV that is used with STD 0005 is BCV 02B1 (logical device name RBCV02B1).

Split the BCV, remove relationship with standard and remove BCV from device group

```
# symmir -g DG03 split DEV002 bcv Id RBCV02B1 -rdf -nop
# symmir -g DG03 cancel DEV002 bcv Id RBCV02B1 -rdf -nop
# symbcv -g DG03 disassociate Id RBCV02B1 -rdf
```

Remove device from RDF group 3 and device group DG03. This can be done with the group still in asynchronous mode, but suspended.

```
# symrdf -g DG03 disable -rdfg 3 -nop
# symrdf -g DG03 suspend -rdfg 3 -nop
# symrdf -g DG03 suspend -rdfg 64 -nop
```

Remove device from device group

```
# symld -g DG03 remove DEV002 -force
```

Remove devices from RDF groups

```
# symrdf -file pairsfile_DYN03_0005 -sid 0822 -rdfg 3 deletepair -nop
# symrdf -file pairsfile_DYN03_0005 -sid 0822 -rdfg 64 deletepair -nop
```

Resume aysnc replication for remaining device in RDF group 3
symrdf -g DG03 resume -rdfg 3 -nop
symrdf -g DG03 enable -rdfg 3 -nop

Resume sync replication for remaining device in RDF group 64
symrdf -g DG03 resume -rdfg 64 -nop

Update device groups on other local SAN mgt. servers using GNS or via manual intervention.

Update device groups at sync and async target sites via manual intervention as GNS remote mirroring is not supported with Concurrent SRDF at the time of writing.

8.1.8 Removing a dynamic RDF group

This process, should it be required, has already been discussed in Section 5.3.5. In practice, if an application is retired, there is not need to remove the RDF group as it may be needed again in the future for another application.

8.2 Starting and stopping SRDF/A

8.2.1 Starting SRDF/A

There are two ways of starting an SRDF/A session:

- Setting the SRDF mode to Asynchronous. If replication was already taking place in a different mode, replication will continue in asynchronous mode, with the R2 becoming consistent after 2 cycles.

The RDF group must be specified in a concurrent SRDF configuration. This is optional in a 2-site configuration. Consistency protection for the SRDF/A devices in the device group should also be enabled.

```
# symrdf -g <DGname> set mode async -nop [ -rdfg <rdfg-num> ]
# symrdf -g <DGname> enable
```

- Establishing or resuming an existing suspended SRDF/A link. An SRDF/A session will be *inactive* until it is either resumed (following a graceful suspend operation) or established.

```
# symrdf -g <DGname> resume -nop [-rdfg rdfg-num]
# symrdf -g <DGname> establish -nop [-rdfg rdfg-num]
```

In a concurrent configuration, the two RDF legs can be stopped and started completely independently of each other. However, only one can be in asynchronous mode.

Hence to temporarily stop SRDF/A, suspend the link. To permanent disable SRDF/A change to another SRDF mode.

8.2.2 Suspending SRDF/A

```
# symrdf -g <DGname> disable -nop [-rdfg rdfg-num]
# symrdf -g <DGname> suspend -nop [-rdfg rdfg-num]
```

The RDF group must be specified in a concurrent configuration. If consistency protection for the SRDF/A devices in the device group is enabled, it should be disabled before the suspend command is issued.

SRDF suspend operations will take much longer with SRDF/A as the process waits until the end of the current cycle before executing the suspend request. For example, in testing a suspend on an SRDF/S replication group took 5 seconds but the same procedure for the same disks running SRDF/A took 45 seconds.

This may be an issue if you need to suspend multiple groups. In testing it could take over 20 minutes to suspend all 32 test SRDF/A groups. For installations with 250ms latency, the process could take a couple of hours.

Note that the `-immediate` flag which suspends SRDF immediately should **not** be used under normal circumstance as this usually requires a lengthy establish operation to be undertaken when restarting SRDF/A.

```
# symrdf -g <DGname> suspend -immediate -nop [-rdfg rdfg-num]
```

Use of suspend –immediate puts the devices in NR state immediately i.e. does not wait until the end of the current cycle. This results in tracks being converted to invalids on both the R1 and the R2 sides of the relationship. Resuming SRDF/A would then require resolving the invalids in the normal way i.e. via the establish operation. Note that dropping out of SRDF/A mode does not compromise the consistency of the data on the R2 side. SRDF/A always provides a consistent image of data at the remote site at all time.

8.2.3 Restarting SRDF/A after controlled suspension

A controlled suspension of an SRDF/A is defined as one using a command of the form

```
# symrdf -g <DGname> suspend -nop [-rdfg rdfg-num]
```

An RDF group suspended cleanly in this way can simply be resumed with the command

```
# symrdf -g <DGname> resume -nop [-rdfg rdfg-num]
```

Note that if there has been a lot of activity on the R1 whilst the link was suspended, then resynchronisation may take an extended time. In this case, it would be prudent to first take a BCV PIT copy to ensure that there is an IO consistent image at the Async target site at all times. This process is described shortly.

8.2.4 Restarting SRDF/A after uncontrolled suspension

An uncontrolled suspension of an SRDF/A session is one in which:

- a) SRDF/A replication is terminated by the system, for example due to lack of cache or inability to communicate with the remote DMX within SRDF/A operational parameters.
- b) The session was manually suspended but using the –immediate flag

Following an uncontrolled suspension, you should attempt to resume the replication as one would following a controlled suspension. If this fails (which it will if IO was occurring to devices in the RDF group at the time of suspension), then an incremental establish will be required instead.

```
# symrdf -g <DGname> establish -nop [-rdfg rdfg-num]
```

Again, if there has been a lot of activity on the R1 whilst the link was suspended, then resynchronisation may take an extended time. In this case, it would be prudent to first take a BCV PIT copy to ensure that there is an IO consistent image at the Async target site at all times. See Section 8.3 for more details. Furthermore, a script called **resume_async** is available to automate this complex process. See Appendix B for further details.

8.2.5 Processing multiple device groups

At times it will be necessary to perform an operation on multiple RDF groups, and hence device groups. This can be done by using a for/foreach loop or by writing a simple script file on-the-fly if you want to retain a record of the operation or for reuse.

For example, if you want to suspend several groups (which may take up to ~1 minute per group) with a single command, the syntax might be:

csch syntax

```
# foreach i ( 3 4 5 6 7 8 9 10 11 12 13 14)
? symrdf -g DG0$i suspend -rdfg $i-nop
? end
```

sh/ksh syntax

```
# for i in 3 4 5 6 7 8 9 10 11 12 13 14
> do
> symrdf -g DG0$i suspend -rdfg $i-nop
> done
```

Create on-the-fly script (one of many possible methods)

In this example, a file that maps device group names to RDF group numbers is used to create command files for device group manipulation on-the-fly

```
# cat DGlist
App3      3
Otms      4
Flexcube  5
App6      6
App7      7
Multifonds 8
App9      9
App10     10
App11     11
App12     12
App13     13
App14     14

# cat DGlist | awk '{print "symrdf -g DG0"$1 " suspend -rdfg " $2" -nop" }' > ./myscript

# cat ./myscript
symrdf -g App3 suspend -rdfg 3 -nop
symrdf -g Otms suspend -rdfg 4 -nop
symrdf -g Flexcube suspend -rdfg 5 -nop
symrdf -g App6 suspend -rdfg 6 -nop
symrdf -g App7 suspend -rdfg 7 -nop
symrdf -g Multifonds suspend -rdfg 8 -nop
symrdf -g App9 suspend -rdfg 9 -nop
symrdf -g App10 suspend -rdfg 10 -nop
symrdf -g App11 suspend -rdfg 11 -nop
symrdf -g App12 suspend -rdfg 12 -nop
symrdf -g App13 suspend -rdfg 13 -nop
symrdf -g App14 suspend -rdfg 14 -nop

# ./myscript
```

8.3 Restarting SRDF/A after extended network outage

8.3.1 Introduction

During a temporary network outage, cycle switching will stop and the write IOs will begin to accumulate in the Capture Delta Set in cache on the source DMX. If the network recovers before SRDF/A aborts (either because of cache exhaustion or before a link timeout has been reached) then replication will continue. Cycle times will be extended for a temporary period whilst the backlog of IOs are shipped across to the remote site.

For detailed information on the link timeout settings (link limbo timer) see the SRDF/A Evaluation document, section 4.5.1.

For further information on how SRDF/A deals with temporary network outages without aborting, and cycle time extension, see SRDF/A Evaluation document, section 6.10.1

In this section, we discuss the process for recovering after an extended network outage when SRDF/A has aborted i.e. suspended in an uncontrolled manner. Furthermore, after an extended outage when IO has occurred on the source volumes, the time taken for the RDF pairs to resynchronise may be considerable. During this time, the R2 image will be inconsistent and useless from an application recovery perspective. Hence, additional steps must be taken to protect against the loss of consistency.

8.3.2 Recovering after loss of links

Prior to starting a resynchronisation after link recovery, a BCV copy of the R2 volumes should be split off (consistently) to ensure that there is a consistent (albeit aging) point-in-time copy at all times in the remote site during the re-synchronization process when the R2 will be inconsistent.

If the link between sites is down for an extended period, and there is a large build up of invalid tracks on the R1 side, resumption of the link with SRDF/A (the default behaviour in Open Systems) can cause a heavy surge of link traffic, created by the large 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 a large build up of invalid tracks.

Note, however, that if:

- IO rates are low
- or
- There is no shortage of cache
- or
- The number of invalid tracks is low

then it will be acceptable to re-enable replication whilst still in asynchronous mode

Note, however, that SRDF/A has a limit of 30,000 tracks per cycle. Hence, resynchronisation when in asynchronous mode may take longer than when in adaptive copy mode when no such limitation exists.

If the R2 was consistent at the time that the SRDF/A session was suspended, the procedure for recovering back to a replicating state, whilst still ensuring that a consistent point-in-time copy is available at the target site at all times is as follows:

1. Verify BCV is fully established.
2. Split PIT BCV off at SRDF/A target site
3. Wait until BCVs finished splitting in the background
4. [Optional: Set mode to adaptive copy write pending mode]
5. Establish RDF replication for device group
6. Wait until RDF Pair Status synchronized
7. [Optional: Set mode to async copy and re-enable device group consistency]
8. Establish PIT BCV

A script called **resume_async** is available to automate this process. See Appendix B for further details.

If SRDF session was suspended whilst the RDF pairs were resynchronising, then the R2 will be inconsistent. Hence, the PIT BCV must NOT be resynchronised with the R2 as this will result in the loss of the only remaining IO consistent copy of data at the SRDF/A target site.

If the above procedure is being followed, then the fact that the BCV is split off indicates that the previous resynchronisation exercise process did not complete. Hence, if the BCV is already found to be split off, then the resynchronisation process should begin immediately (Steps 4 or 5) as the R2 may be inconsistent and it is not safe to re-establish the BCV without potential loss of the only consistent copy of data at the Async target site.

8.4 Single Application Failover/Failback – 2-site SRDF/A configuration

SRDF/A failover occurs at the RDF group level. Each application should have its own RDF group, and the device group for the application should contain all devices in the RDF group to ensure IO consistency between the devices.

Note that single application failover will usually occur when there is an application or server issues rather than a storage or building infrastructure problem. Hence, no data loss should occur between the R1 and R2 devices (as with synchronous SRDF) and two cycles after source IO stops, the R2 will be identical to the R1.

Scripts called **2site_async_failover** and **2site_async_failback** are available to automate the process. See Appendix B for further details.

The process for failing over a single RDF group is:

1. If IO has ceased at the source site, confirm that the number of track not yet committed to the R2 side is 0 i.e. all data has been replicated to the remote site.
2. Disable consistency for RDF group
3. Failover RDF group via device group

Example

```
# symrdf -g <DGname> -rdfa | grep Committed
# symrdf -g <DGname> disable -nop
# symrdf -g <DGname> failover -nop
```

In its simplest terms, the failback process where changes made at R2 end need to be copied back to R1 is as follows. Local processes and procedures may differ.

1. Optional: Run update command to back copy changes made at R2 to R1 before application is shutdown
2. Shutdown application at target site
3. Failback RDF group via device group
4. Enable consistency for RDF group
5. Confirm replication has restarted
6. Inform application owners that they can restart application

In a real DR scenario, once failover to the SRDF/A target site has been executed, the PIT BCV should be split off to provide a golden copy of the last image of data from the production site.

Example

```
# symrdf -g <DGname> update -nop
```

When update completed:

```
# symrdf -g <DGname> failback -nop
# symrdf -g <DGname> enable -nop
# symrdf -g <DGname> query -rdfa
```

8.5 Single Application Failover/Failback – non-Star 3-site configuration

*The following procedure refers to a **non-Star** Concurrent SRDF configuration. See the “SRDF/Star for Open Systems” Solution Guide for the process for that type of configuration.*

Failover in a non-Star concurrent SRDF environment is somewhat more complex than the 2-site version as there will be an interaction between the SRDF/S and SRDF/A replication legs. Furthermore, it becomes mandatory to define which RDF leg the command is to operate on

8.5.1 Concurrent SRDF - Synchronous leg failover/failback

This subsection describes failover of entire RDF groups (i.e. typically entire applications). Failover of individual servers is covered in a later section.

Note that failover will usually be to the Synchronous target following an application outage at the Production site.

Scripts called **3site_sync_failover** and **3site_sync_failback** are available to automate the process. See Appendix B for further details.

To failover the synchronous leg, the proper process is:

1. If IO has ceased at the source site, confirm that the number of tracks not committed to the Async R2 side is 0, i.e. all outstanding data has been copied across.
2. Suspend the SRDF/A leg first, disabling consistency first
3. Execute a failover of the SRDF/S leg
4. Confirm status of both legs with symrdf query

In the following examples, the device group that contains the concurrent source devices is DG03. The RDF group number is 3 for the async leg, and 64 for the sync leg.

Example

```
# symrdf -g DG03 query -rdfg 3 | grep Committed
# symrdf -g DG03 disable -rdfg 3 -nop
# symrdf -g DG03 suspend -rdfg 3 -nop
# symrdf -g DG03 failover -rdfg 64 -nop
# symrdf -g DG03 -rdfg all
```

Note in the above command that “-rdfg all” is used to show both legs

Note that it is essential to execute the control commands on the correct leg.

The failback process is:

1. Optional: Run update command to back copy changes made at Sync target site R2 to R1 before application is shutdown
2. Shutdown application at Sync target site
3. Failback SRDF/S link and wait until no R1 invalids on this link
4. Confirm replication on synchronous leg
5. Resume SRDF/A replication and enable consistency
6. Confirm replication on both legs
7. Inform application owners that they can restart application

Example

```
# symrdf -g DG03 update -rdfg 64 -nop
```

When update completed:

```
# symrdf -g DG03 failback -rdfg 64 -nop
```

```
# symrdf -g DG03 query -rdfg 64
```

Wait until there are no invalid tracks on the sync leg and then resume replication on the async leg.

```
# symrdf -g DG03 resume -rdfg 3 -nop
```

```
# symrdf -g DG03 enable -rdfg 3 -nop
```

```
# symrdf -g DG03 query -rdfg all
```

8.5.2 Concurrent SRDF - Asynchronous leg failover/failback

This subsection describes failover of entire RDF groups (i.e. typically entire applications).

Note that failover will usually be to the Synchronous target following an application outage at the Production site. Asynchronous failover will usually only occur during COB testing of the Async target site or during an event that renders failover to the Sync target impossible.

Scripts called **3site_async_failover** and **3site_async_failback** are available to automate the process. See Appendix B for further details.

To failover the asynchronous leg, the proper process is:

1. Suspend the SRDF/S leg first
2. If IO has ceased at the source site, confirm that the number of tracks not committed to the Async R2 side is 0, i.e. all outstanding data has been copied across.
3. Execute a failover of the SRDF/A leg
4. Confirm status of both legs with symrdf query

In a real DR scenario, once failover to the SRDF/A target site has been executed, the PIT BCV should be split off to provide a golden copy of the last image of data from the production site.

In the following examples, the device group that contains the concurrent source devices is DG03. The RDF group number is 3 for the async leg, and 64 for the sync leg.

Example

```
# symrdf -g DG03 suspend -rdfg 64 -nop
```

```
# symrdf -g DG03 query -rdfg 3 | grep Committed
```

```
# symrdf -g DG03 disable -rdfg 3 -nop
```

```
# symrdf -g DG03 failover -rdfg 3 -nop
```

```
# symrdf -g DG03 -rdfg all
```

Note in the above command that “-rdfg all” is used to show both legs

Note that it is essential to execute the control commands on the correct leg.

The failback process is:

1. Optional: Run update command to back copy changes made at Async target site R2 to R1 before application is shutdown
2. Shutdown application at Async target site
3. Failback SRDF/A link and enable consistency
4. Confirm replication on Asynchronous leg
5. Wait until there are no invalid tracks on the async leg
6. Resume SRDF/S replication
7. Confirm replication on both legs
8. Inform application owners that they can restart application

Example

```
# symrdf -g DG03 update -rdfg 3 -nop
```

When update completed:

```
# symrdf -g DG03 failback -rdfg 3 -nop
```

```
# symrdf -g DG03 enable -rdfg 3 -nop  
# symrdf -g DG03 query -rdfg 3
```

Wait until there are no invalid tracks on the async leg and then resume replication on the sync leg.

```
# symrdf -g DG03 resume -rdfg 64 -nop  
# symrdf -g DG03 query -rdfg all
```

8.6 Subset failover/failback on Synchronous leg – non-Star 3-site configuration

8.6.1 Introduction

*The following procedure refers to a **non-Star** Concurrent SRDF configuration. **Subset failover is not possible in an SRDF/Star configuration.***

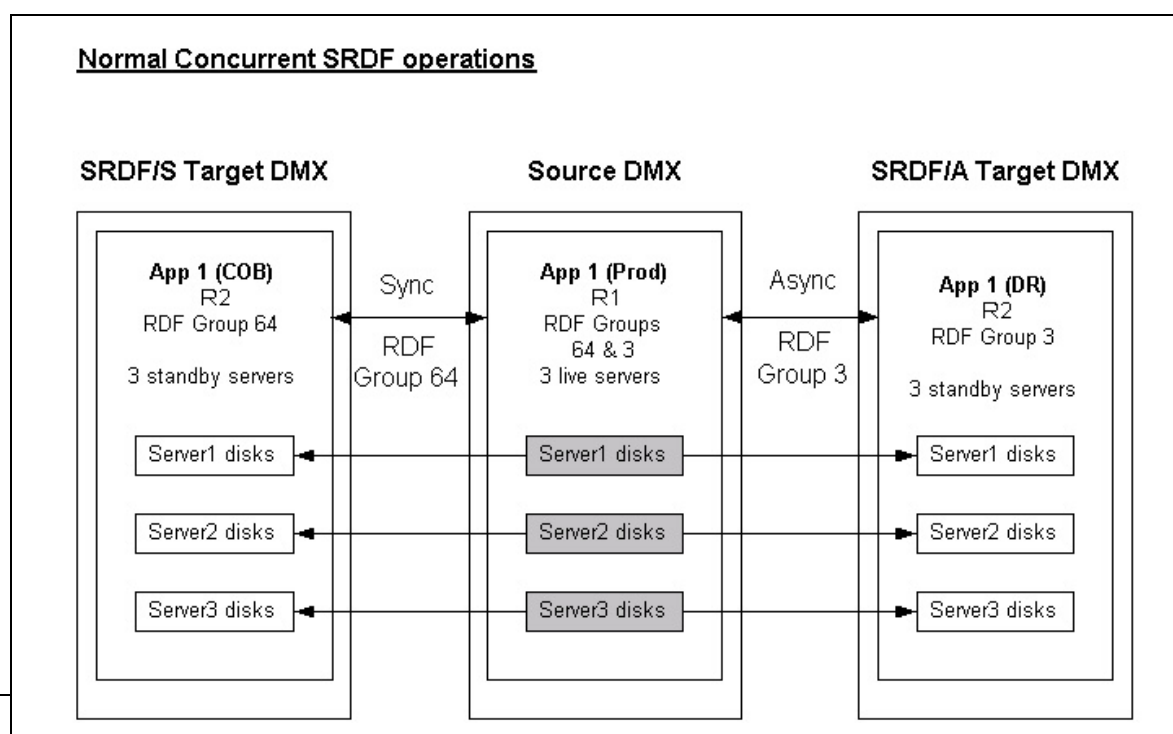
This section covers the most complex failover/failback scenario involving only a subset of disks within an RDF group. A certain amount of explanation of the scenario is required prior to documenting the process. Subset failover only applies to the Sync leg of a concurrent SRDF implementation.

This process is required in the scenario where a subset (one or more) of the servers that comprise an application have failed, and COB for the application is being invoked by only failing over that subset of servers to the synchronous SRDF target site.

Note that subset failover on the synchronous leg of a concurrent SRDF configuration is not recommended either by EMC nor GDSE, but is necessarily being supported to meet business requirements.

The figure below shows an example of part of a Concurrent SRDF implementation. Application App1 uses RDF group 64 for its synchronous replication and RDF group 3 for its Async replication. Note that in production, it is recommended that synchronous replication RDF groups be odd numbers and their corresponding async groups the following even number.

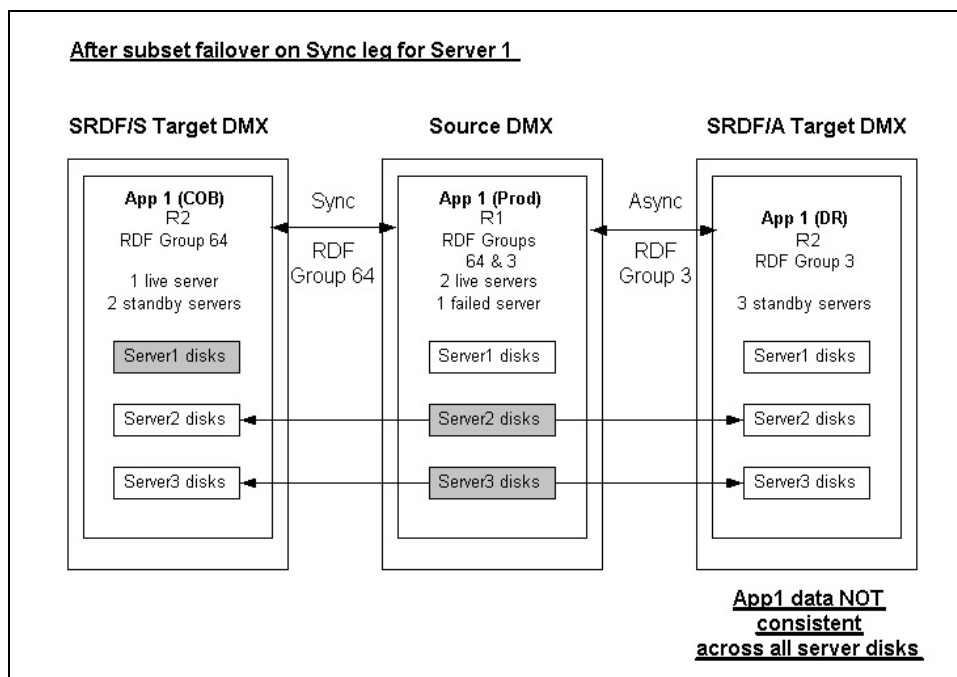
App1 has three servers running in the production site with storage in the Source DMX. As all of these disks are part of the same RDF group on each leg, then the R2s at each remote site are IO consistent with each other.



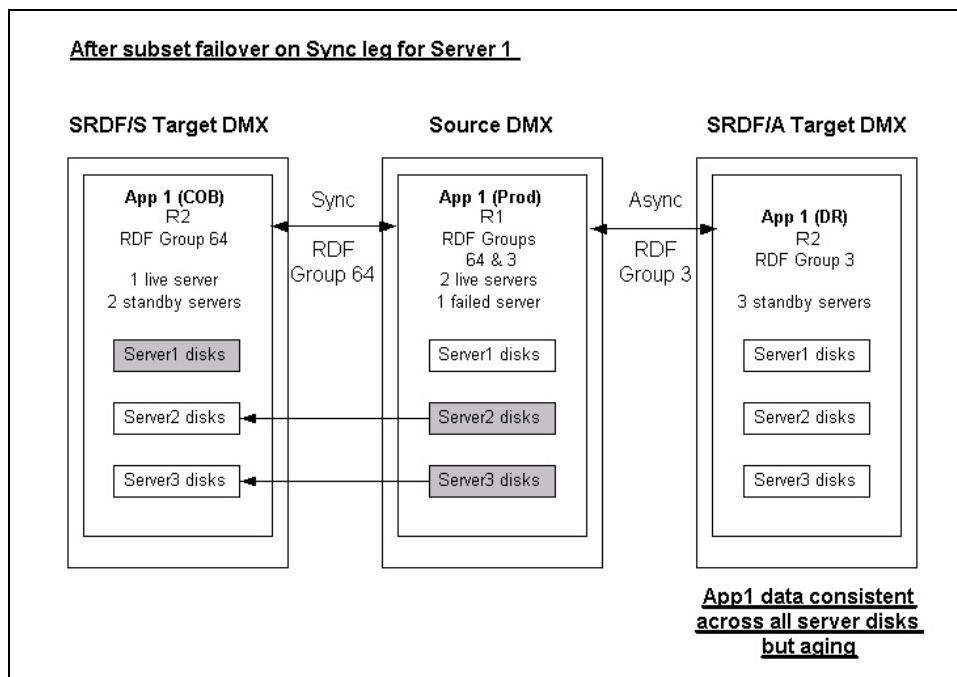
If server1 were to fail, there are two options for failover:

1. Failover entire application i.e. entire RDF group across to the SRDF/S target
2. Failover the subset of disks belonging to server 1, in a manner similar to what we do today with 2-site SRDF/S.

Option 2 will often be needed as failing over a single server is often preferred to failing over the entire application. Note, however, if SRDF/A replication for Server2 and Server3 were restarted, then the overall data set for App1 at the SRDF/A target would not be self-consistent.



Alternatively, if a consistent set of data is required to be preserved at the SRDF/A target site, it will be necessary not to restart SRDF/A replication for Server2 and Server3. The image on the R2 disks would then be IO consistent but aging.



It will be necessary to understand from the application owner if they prefer to have:

1. Continued async replication for those servers which have not failed over to the local COB site. Async target data will not be self-consistent across the entire application. This should only be undertaken if the application owner is sure that they can recover at the async site with inconsistent data between servers.
2. Cease all replication to the async site for that application. The data will be a consistent point-in-time copy of production but dropping in value rapidly as it ages.

This second options is engineering's recommended default approach as it does provide data set from which it is guaranteed one can recover from should out-of-region COB be invoked following a regional disaster.

8.6.2 Executing subset failover/failback

All previous failover/failback examples in this guide have been executed on entire RDF groups.

In subset failover, a subset of disks within an RDF group is failed over. This produced certain challenges in terms of device group definition in a concurrent SRDF environment. The recommended Concurrent SRDF device group definition standards are described in Section 7.3, including the more complex configurations where subset failover is required.

The following examples assume that device groups have been configured for subset failover support as per Section 7.3 and are as follows. Refer to the diagrams on the previous pages for the server and RDF group setup. The asynchronous leg used RDF group 3 and the synchronous leg used RDF group 64. Server1 is assumed to have failed.

Device group defined at source site:

- App1 (includes all devices in RDF group 3 and consequently RDF group 64 as this is a concurrent configuration)

Device groups defined at synchronous target site:

- App1_Server1 (includes all R2 devices paired with R1s owner by server1)
- App1_Server2 (includes all R2 devices paired with R1s owner by server2)
- App1_Server3 (includes all R2 devices paired with R1s owner by server3)

The subset failover process is more complex than the non-subset concurrent failover as the SRDF/S failover has to be executed from the SRDF/S target, but the SRDF/A sections need to be done on the source DMX.

The recommended failover process is as follows. It will require a window open on both the Source site SAN mgt. Server and also the Sync target site SAN mgt server.

1. SOURCE SITE: Disable consistency on the SRDF/A leg
2. SOURCE SITE: Suspend the SRDF/A leg
3. **SYNC TARGET SITE:** "Subset" failover the disks belonging to Server1
4. SOURCE SITE: **Optional:** Resume the SRDF/A leg
(if it has been decided that replicating only Server2 and Server3 data to SRDF/A target is appropriate)
5. SOURCE SITE: **Optional:** Enable consistency

Note that as the suspension of the SRDF/A leg is very brief, a resume of the SRDF/A replication without first taking a PIT copy may be acceptable.

Note that if SRDF/A replication is restarted for the RDF group, then the disks that have been failed over to the SRDF/S target will have an Invalid RDF Pair State on the Async leg.

The recommended failback process is as follows. It will require a window open on both the Source site SAN mgt. Server and also the Sync target site SAN mgt server. Note that steps 7 & 8 will only be needed if optional steps 4 & 5 above were executed.

6. **SYNC TARGET SITE:** **Optional:** Run update command to back copy changes made at the sync target site R2 to R1 before application is shutdown
7. **SOURCE SITE:** Disable consistency on the SRDF/A leg
8. **SOURCE SITE:** Suspend the SRDF/A leg
9. **SYNC TARGET SITE:** "Subset" failback the disks belonging to Server1
10. **SOURCE SITE:** Resume the SRDF/A leg
11. **SOURCE SITE:** Enable consistency

Hence, in terms of specific commands, the **failover** process is:

At Source site:

```
# symrdf -g App1 disable -rdfg 3 -nop
# symrdf -g App1 suspend -rdfg 3 -nop
```

At Sync Target site

```
# symrdf -g App1_Server1 failover -rdfg 64 -nop
```

At Source site:

Note that enabling SRDF/A replication will cause the R2 at the SRDF/A target *not to be IO consistent across the entire RDF group*. Hence, SRDF/A should *only* be re-enabled if the business feels that a more up-to-date copy of Server2 and Server3 is preferable to having consistent, but aging, data at the SRDF/A target site for all 3 servers .

```
# symrdf -g App1 resume -rdfg 3 -nop
# symrdf -g App1 enable -rdfg 3 -nop
```

A query on the RDF pair status will show:

Device belonging to	Sync RDF Pair Status	Async RDF Pair Status
Server1 devices	Failed Over	<i>Invalid</i>
Server2 devices	Synchronized	Consistent
Server3 devices	Synchronized	Consistent

Note that the Async RDF Pair Status for the devices that are failed over down the synchronous leg is Invalid.

The **failback** process following the above failover process is:

At Sync Target site (Optional)

```
# symrdf -g App1 update -rdfg 64 -nop
```

At source site:

If SRDF/A was resumed for the rest of the RDF group, it will be necessary to suspend async replication (for the whole RDF group) to re-commence async for Server1 devices. A –force will be required as the RDF Pair State is not in the expected state for all devices.

```
# symrdf –g App1 disable –rdfg 3 –nop
# symrdf –g App1 suspend –rdfg 3 –force –nop
```

At Sync Target site

```
# symrdf –g App1_Server1 failback –rdfg 64 -nop
```

At source site:

```
# symrdf –g App1 resume –rdfg 3 –nop
# symrdf –g App1 enable –rdfg 3 -nop
```

Note that the SRDF/A leg resynchronisation may take an extended period of time. However, as the R2 devices were not in an IO consistent state across the whole RDF group prior to the resynchronisation exercise, then there is limited value in first taking a PIT copy of the data. Nevertheless, one can be taken to provided limited protection during the resync.

Note also that it is possible to force the failover from the Sync Target Site, without first stopping the Async leg replication. However, recovery back to the original configuration is more complex and this activity is not recommended.

A set of scripts is available to help automate this complex process. See Appendix B for further details.

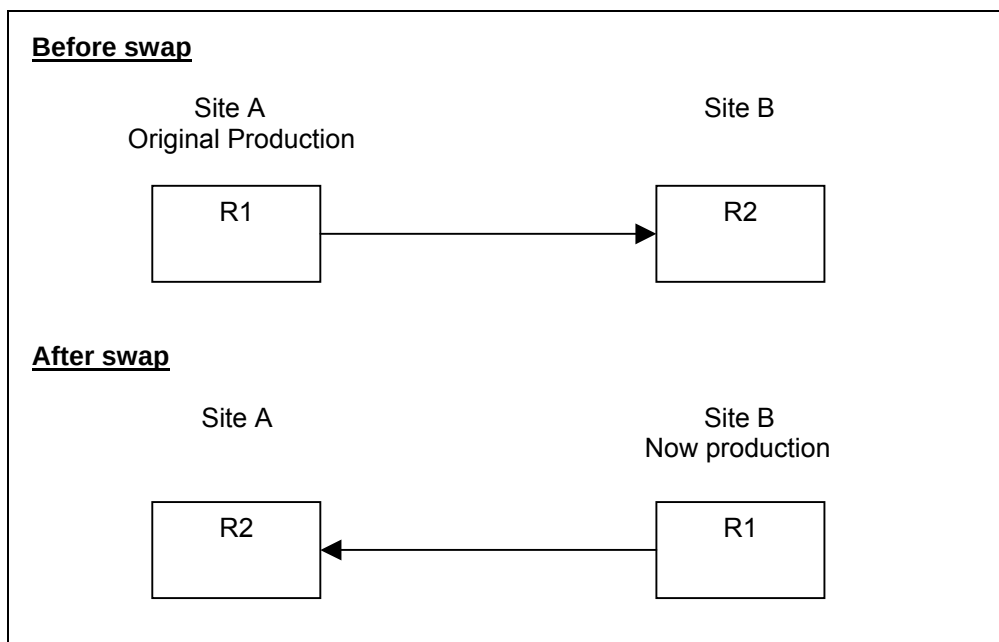
3-site sync leg subset failover		
Script	Location script must be run from	Action
3site_subset_failover_phase1	Run from source site.	Suspends Async leg
3site_subset_failover_phase2	Run from Sync target site.	Fails over subset of disks in RDF group
3site_subset_failover_phase3	Run from source site.	Optional: Resume Async leg for remaining devices in RDF group
3-site sync leg subset failback		
Script	Location script must be run from	Action
3site_subset_failback_phase1	Run from source site.	Required only if async replication was resumed in failover Phase 3: Suspends Async leg
3site_subset_failback_phase2	Run from Sync target site.	Fail back subset of disks in RDF group
3site_subset_failback_phase3	Run from source site.	Resume Async leg for all devices in RDF group

8.7 SRDF Personality Swapping

With traditional SRDF, Primary site data is replicated to the Secondary site. The source volumes in the Primary site are known as R1 volumes, and the corresponding target volumes in the Secondary site as R2 volumes. The replication direction is fixed.

With dynamic RDF, you can swap the R1/R2 personality of RDF pairs. Source R1 devices become target R2 devices, and vice versa. Swapping SRDF devices allows the original R2 side to take over operations while retaining a remote mirror on the original R1 side i.e. the original R1 becomes an R2 and vice versa (see figure below).

This is used in some regions to ensure that when a failover to COB has occurred, that off-site COB is still available (i.e. back to the original production site). For example, in the diagram below, if an application that usually runs in Site A has failed over to Site B due to server hardware failure, then personality swapping ensures that synchronous off-site replication still continues.



8.7.1 2-site process (non-MSD)

A script called **2site_async_swap** is available to automate the whole process. See Appendix B for further details.

The manual procedure for swapping SRDF/A replication is as follows:

- Disable consistency
- Executing a failover. RDF links are suspended.
- Execute the Swap
- Execute an establish (no data moves)
- Enable consistency
- Confirm RDF group type has changed (e.g. RDF1 to RDF2) and replication has recommenced.

Example:

```
# symrdf -g DG03 disable -nop
# symrdf -g DG03 failover -nop
# symrdf -g DG03 swap -nop
# symrdf -g DG03 establish -nop
# symrdf -g DG03 enable -nop
# symrdf -g DG03 query -rdfa
```

Note that “fast-swap” (failover –establish) is not supported on SRDF/A enabled devices and hence is not used here.

8.7.2 2-site process (MSC)

Swapping via Composite Groups is not supported (“Invalid action” error) and hence MSC bound groups cannot be swapped as a single entity.

8.7.3 3-site process

Swapping is not supported in a non-Star Concurrent SRDF configurations. However, site swapping is supported with SRDF/Star, within the limits set by Citigroup's Data Centre Strategy. See the **SRDF/Star for Open Systems** Solutions Guide for further details.

8.8 2-Site Multi-Session Consistency Operations

8.8.1 Independent RDF group failover/failback

One of the benefits of Multi-Session Consistency in an SRDF/A environment is that it allows one to guarantee IO consistency at the R2-end between RDF groups (and hence individual applications), whilst still permitting failover independence for each application.

The process for failing over a single RDF group within a Composite Group replicating under MSC control is as follows.

1. Temporarily disable consistency on the composite group
2. Temporarily remove the devices in device group to be failed over from the Composite Group. Composite group definition on all local servers running RDF daemons for MSC control must be updated by GNS or manually.
3. Re-enable consistency on the composite group
4. Failover the device group
5. Confirm other groups in composite group are still replicating
6. Confirm CG definitions on remote CG servers are also up to date.

Scripts called **2site_MSC_failover** and **2site_MSC_failback** are available to automate the process. See Appendix B for further details.

In a real DR scenario, once failover to the SRDF/A target site has been executed, the PIT BCV should be split off to provide a golden copy of the last image of data from the production site.

For example, to failover RDF group 3 (device group DG03) on DMX 0822, which is part of a Composite Group multi replicating under MSC control:

```
# symcg disable -cg multi
# symcg -cg multi rmall dev -rdfg 3
```

Ensure that local Composite Group definitions on local servers controlling MSC replication are updated, either automatically via GNS (strongly recommended) or manually, before proceeding with re-enabling MSC replication.

```
# symcg enable -cg multi
# syrmdf -g DG03 failover
# symstat -type cache -sid 0822 -reptype rdfa -i 10 -g all
```

GNS remote mirroring should be used to ensure updated CG definitions are visible to all management servers at the remote site too.

The process for failing back the RDF group and adding it back under MSC control is:

1. Failback RDF group
2. Temporarily disable consistency on the composite group
3. Add devices in RDF group back into Composite Group
4. Composite group definition on all local servers running RDF daemons for MSC control must be updated by GNS or manually.
5. Re-enable consistency on the composite group
6. Confirm all groups in composite group are replicating together

```
# syrmdf -g DG03 failback
# symcg disable -cg multi
# symcg -cg multi addall dev -rdfg 03 -sid 0822
```

Ensure that local Composite Group definitions on local servers controlling MSC replication are updated, either automatically via GNS (strongly recommended) or manually, before proceeding with re-enabling MSC replication.

```
# symcg enable -cg multi  
# symstat -type cache -sid 0822 -reptype rdfa -i 10 -g all
```

GNS remote mirroring should be used to automatically replicate the updated CG definition to the remote site. If GNS is not in use, the remote CG definition will need to be updated manually.

8.8.2 Composite group failover/failback

An entire Composite Group can also be failed over. This would be required in the case where there has been an issue at the source site and DR is being invoked. In such cases, replication usually already have ceased due to the MSC Consistency Group tripping. Failover via the Composite group (rather than via the underlying RDF groups) ensures consistency at the R2 end.

When MSC replication is halted due to the MSC Composite Group tripping, a clean up process is required to determine if the receive Delta Sets (on the Target DMX) for each RDF group should be applied to the R2 or discarded. A description of this process is beyond the scope of the present document, but can be found in Appendix H of the SRDF/A Evaluation document.

Scripts called **2site_CG_failover** and **2site_CG_failback** are available to automate the process. See Appendix B for further details.

The command to failover a consistency group is

```
# symrdf -cg <CGname> failover -force -symforce -nop
```

The **-force** option allows you to perform control operations on SRDF devices when they are not in the expected state. In this case, it is needed as consistency is enabled.

The **-symforce** option allows you to force an operation that overrides instances where they are normally rejected, usually as they will result in data loss. Note: To enable the **-symforce** option for RDF use, a behavior parameter called SYMAPI_ALLOW_RDF_SYMFORCE in the options file must be set to **TRUE**.

In this case **-symforce** is needed as the post-trip MSC clean up process can leave the invalid tracks tables in a state that requires the use of this flag.

In a real DR scenario, once failover to the SRDF/A target site has been executed, the PIT BCV should be split off to provide a golden copy of the last image of data from the production site.

Failback process requires the use of just the **-force** flag

```
# symrdf -cg <CGname> failback -force -nop
```

8.9 Changing replication mode

Changing mode to SRDF/A requires an SRDF/A license key. Changing mode to both SRDF/S or either Adaptive Copy mode can be undertaken with an SRDF license key. Changing mode to Adaptive Copy (only) can be achieved using an SRDF/DM (Data Movement) key.

A concurrent SRDF configuration will have both SRDF and SRDF/A licenses. A 2-site SRDF/A configuration may only have the SRDF/A license, for which EMC are obliged to provide both an SRDF/A key and an SRDF/DM key to allow use of adaptive copy modes for SRDF/A management.

8.9.1 Changing to Async mode from Sync or Adaptive copy

```
# symrdf -g <DGname> set mode async -nop
# symrdf -g <DGname> enable -nop
```

8.9.2 Changing from Async to Adaptive copy mode

```
# symrdf -g <DGname> disable -nop
# symrdf -g <DGname> set mode acp_wp
```

or

```
# symrdf -g <DGname> disable -nop
# symrdf -g <DGname> set mode acp_disk
```

8.9.3 Changing from Async to Sync mode

It is possible to switch from async to sync mode whilst maintaining consistency at the R2 end at all time. However, this is not the default and an additional flag is needed.

```
# symrdf -g <DGname> disable -nop
# symrdf -g <DGname> set mode sync -consistent -nop
```

Note that in practice, this change will never be used where SRDF/A is running over a long distance leg, as the performance would be unacceptable.

8.10 Interactive monitoring of SRDF/A configuration and replication

This section covers the commands that can be used to monitor and interrogate the system on the status of SRDF/A replication.

Often there is more than one-way in which to get a piece of information. The most appropriate command will depend on the exact circumstances.

The main commands to be used are on:

1. `symstat -reptype rdfa`
 - a. `-type cycle`
 - b. `-type cache`
 - c. `-type requests`
2. `symrdf list -concurrent`
3. `symrdf -g <DGname> query -rdfa -rdfg all`
4. `symcfg list -rdfg all`
5. `symcfg list -ra all -switched`
6. `symevent list`

8.10.1 symstat

The `symstat` command now has an option `-reptype rdfa`, which can be used to request SRDF/A session statistics. There are number of additional flags that can be used in various combinations to get differing output.

When using `-reptype rdfa`, there are three further report type options that can be specified:

-type cycle

This is a particularly useful for:

- real time monitoring of the replication process
- confirming last cycle time
- confirming cycle a.k.a. Delta Set sizing (in units of 32K bytes)
- Note: only shows ACTIVE cycles unless `-g` or `-rdfg` options are added to the command.

-type cache

This is a particularly useful for:

- showing cache utilization on the specified system
- showing session priority which determines the order in which SRDF/A sessions will abort should cache exhaustion occur (high numbers i.e. low priority drop out first)
- Note: only shows ACTIVE cycles unless `-g` or `-rdfg` options are added to the command.

-type requests

This is a particularly useful for:

- observing both host and RDF write rates
- Note: only shows ACTIVE cycles unless `-g` or `-rdfg` options are added to the command.

Furthermore, the scope of the report can be controlled as follows:

-g all

- Shows device group names
- Most clearly shows concurrent configuration
- Shows both active SRDF/A and inactive groups (including SRDF/S sessions)
- Does not show sessions that do not have a device group associated with it (e.g. those being controlled via a device file)

-g <DGname> can also be used to report on a single group. This is particularly useful for looking at concurrent SRDF configurations.

-rdfg all

- Shows information on ALL SRDF/A sessions, including those not managed by a device group
- Does not show device group names
- Presents output in an unsorted order
- Shows both active SRDF/A and inactive groups (including SRDF/S sessions)

-rdfg <rdf-group-number> can also be used to report on a single group

A number of examples of symstat output can be found in Appendices F & I in the SRDF/A Evaluation document, and are not repeated here. Only a few choice examples are included here.

The following example shows symstat output type cycle for a single device group (DG03). The device group is for a set of disks running concurrent SRDF. This is shown by the fact that the device group has 2 RDF groups associated with it.

Group 3 is the SRDF/A leg, as the columns relating to SRDF/A metrics are non-zero (e.g. Last Switch). Group 64 is the SRDF/S leg. The columns relating to SRDF/A metrics are not relevant for SRDF/S and hence contain no meaningful data.

This session is active as can be inferred from the fact that the Last Switch time is shown (22 seconds in this case). The active cycle # is the number of cycles that have occurred since the session was started (last resume or establish).

```
# symstat -reptype rdfa -g DG03 -type cycle -i 5
```

SRDF/A Session Cycle Summary Information

```
Symmetrix Id   : 000187700822
Timestamp      : 10:22:25
```

Device Group Name	RA Grp	Session			Cycle Time (sec)				Cycle Size	
		Type	Number	Active Cycle#	Min	Avg	Last	Switch	Active	Inactive
DG03	3	RDF1	2	3575	29	29	29	22	598	0
	64	-	63	0	-	0	0	-	0	0

Legend for the Attribute of Cycle Size:

```
RDF1: Active = Capture   Inactive = Transmit
RDF2: Active = Receive   Inactive = Apply
```

The following example shows symstat output type cache for all device groups.

```
# symstat -reptype rdfa -g all -type cache -i 5
```

SRDF/A Session Cache Summary Information

Symmetrix Id : 000187700822
Timestamp : 10:36:23

System Write Pending Limit : 3065494 (93.55 GB)
Cache Slots available for all SRDF/A sessions : 2881564 (87.94 GB)
Total Local Write Pending Count : 0
Total System Write Pending Count : 0

Device Group Name	RA Grp	Session				Cache Slots In Use	%Available Cache Used
		Type	Number	Priority	Status		
DG03	3	RDF1	2	50	Active	0	0.0
	64	-	63	33	Inactive	0	0.0
DG19	19	RDF1	18	33	Active	0	0.0
DG20	20	RDF1	19	33	Active	0	0.0
DG21	21	RDF1	20	33	Active	0	0.0
DG22	22	RDF1	21	33	Active	0	0.0
DG23	23	RDF1	22	33	Active	0	0.0
DG24	24	RDF1	23	33	Active	0	0.0
DG25	25	RDF1	24	33	Active	0	0.0
DG26	26	RDF1	25	33	Active	0	0.0
DG27	27	RDF1	26	33	Active	0	0.0
DG28	28	RDF1	27	33	Active	0	0.0
DG29	29	RDF1	28	33	Active	0	0.0
DG30	30	RDF1	29	33	Active	0	0.0
DG31	31	RDF1	30	33	Active	0	0.0
DG32	32	RDF1	31	33	Active	0	0.0
DG33	33	RDF1	32	33	Active	0	0.0
DG34	34	RDF1	33	33	Active	0	0.0
DG35	35	-	34	33	Inactive	0	0.0
DG36	36	-	35	33	Inactive	0	0.0
DG37	37	-	36	33	Inactive	0	0.0
DG38	38	-	37	33	Inactive	0	0.0
DG39	39	-	38	33	Inactive	0	0.0
DG40	40	-	39	33	Inactive	0	0.0
Total						0	0.0
Slots						0.00	
GB							

This view shows:

- That DG03 is the only concurrent SRDF group as it is the only device group that has two RDF groups associated with it. All other groups are 2-site, each having a single RDF group associated with it.
- Some groups are not actively replicating with SRDF/A. Note that it is not possible to tell if these groups are inactive SRDF/A or active SRDF/S sessions from this view.

8.10.2 symrdf list –concurrent

A useful view of concurrent replication for an entire DMX can be obtained as follows. Note that this output shows both meta heads and meta members.

The RDF group entries can be used to filter the output to get a view for a particular set of devices associated with an application.

The RDF

```
# symrdf list –concurrent –sid 0822
```

Symmetrix ID: 000187700822

Local Device View											
Sym Dev	RDev	RDF Typ:G	STATUS			MODES MDA	R1 Inv Tracks	R2 Inv Tracks	RDF S T A T E S		
			SA	RA	LNK				Dev	RDev	Pair
0001	0001	R1:3	RW	RW	RW	A..	0	0	RW	WD	Consistent
	0001	R1:35	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0002	0002	R1:3	RW	RW	RW	A..	-	-	RW	WD	Consistent
	0002	R1:35	RW	RW	RW	S..	-	-	RW	WD	Synchronized
0003	0003	R1:3	RW	RW	RW	A..	-	-	RW	WD	Consistent
	0003	R1:35	RW	RW	RW	S..	-	-	RW	WD	Synchronized
0004	0004	R1:3	RW	RW	RW	A..	-	-	RW	WD	Consistent
	0004	R1:35	RW	RW	RW	S..	-	-	RW	WD	Synchronized
0005	0005	R1:3	RW	RW	RW	A..	0	0	RW	WD	Consistent
	0005	R1:35	RW	RW	RW	S..	0	0	RW	WD	Synchronized
.	.	.	.	.	.	.	.	.	.	.	.
0275	0275	R1:32	RW	RW	RW	A..	0	0	RW	WD	Consistent
	0275	R1:64	RW	RW	RW	S..	0	0	RW	WD	Synchronized
0276	0276	R1:32	RW	RW	RW	A..	-	-	RW	WD	Consistent
	0276	R1:64	RW	RW	RW	S..	-	-	RW	WD	Synchronized
0277	0277	R1:32	RW	RW	RW	A..	-	-	RW	WD	Consistent
	0277	R1:64	RW	RW	RW	S..	-	-	RW	WD	Synchronized
0278	0278	R1:32	RW	RW	RW	A..	-	-	RW	WD	Consistent
	0278	R1:64	RW	RW	RW	S..	-	-	RW	WD	Synchronized
Total							-----				
Track(s)							0	0			
MB(s)							0.0	0.0			

Legend for MODES:

M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
D(omino) : X = Enabled, . = Disabled
A(daptive Copy) : D = Disk Mode, W = WP Mode, . = ACp off

8.10.3 symrdf -g <device group> query -rdfa -rdfg all

This is the key query command for Concurrent SRDF/A management.

In particular it shows:

3. SRDF/A replication status each leg (-rdfa flag). Only one leg should have SRDF/A active. For that leg it shows:
 - The time that the R2 is behind the R1
 - The last cycle time
 - Cache utilization for that group
4. Status of each leg (-rdfg all flag):
 - RDF pair status
 - If device group consistency is enabled (different to RDF pair state being consistent)
 - Only meta heads are shown (unlike symrdf list, which also meta members)

Status of individual legs can be determined by specifying the rdfg associated with that leg.

In a concurrent SRDF configuration, one RDF group (only) associated with a device group will be asynchronously replicating, and the other will be inactive. In this case, we can see that RDF group 3 is Active (SRDF/A replicating) and RDF group 64 is inactive, and would be the synchronous leg.

The Asynchronous R2 target is 57 seconds behind the R1. For a 30 second cycle time, the async R2 will always be at least 30 seconds behind the R1, but not more than 60 seconds under normal conditions.

symrdf -g otms query -rdfa -rdfg all

```
Device Group (DG) Name      : otms
DG's Type                  : RDF1
DG's Symmetrix ID          : 000187700822

RDFA Session Number        : 2
RDFA Cycle Number          : 3641
RDFA Session Status        : Active
RDFA Minimum Cycle Time    : 00:00:30
RDFA Avg Cycle Time        : 00:00:30
Duration of Last cycle     : 00:00:30
RDFA Session Priority       : 33
Tracks not Committed to the R2 Side: 0
Time that R2 is behind R1   : 00:00:57
RDFA R1 Side Percent Cache In Use : 0
RDFA R2 Side Percent Cache In Use : 0

Remote Symmetrix ID        : 000187721497
RDF (RA) Group Number      : 3 (02)

RDFA Session Number        : 34
RDFA Cycle Number          : 0
RDFA Session Status        : Inactive
RDFA Minimum Cycle Time    : 00:00:30
RDFA Avg Cycle Time        : 00:00:00
Duration of Last cycle     : 00:00:00
RDFA Session Priority       : 33
Tracks not Committed to the R2 Side: 0
Time that R2 is behind R1   : 00:00:00
RDFA R1 Side Percent Cache In Use : 0
RDFA R2 Side Percent Cache In Use : 0

Remote Symmetrix ID        : 000187721135
RDF (RA) Group Number      : 35 (22)
```

Source (R1) View					Target (R2) View					MODES	
-----					-----					-----	
Standard	ST				LI	ST					
Logical	A				N	A					
Device	T	R1 Inv	R2 Inv		K	T	R1 Inv	R2 Inv		RDF Pair	
Dev	E	Tracks	Tracks		S	Dev	E	Tracks	Tracks	MDAC	STATE
-----					-----					-----	
DEV002	0005	RW	0	0	RW	0005	WD	0	0	A..X	Consistent
		RW	0	0	RW	0005	WD	0	0	S...	Synchronized
DEV003	0009	RW	0	0	RW	0009	WD	0	0	A..X	Consistent
		RW	0	0	RW	0009	WD	0	0	S...	Synchronized
DEV004	000D	RW	0	0	RW	000D	WD	0	0	A..X	Consistent
		RW	0	0	RW	000D	WD	0	0	S...	Synchronized
DEV001	0001	RW	0	0	RW	0001	WD	0	0	A..X	Consistent
		RW	0	0	RW	0001	WD	0	0	S...	Synchronized
Total											
Track(s)		-----		-----		-----		-----		-----	
MB(s)		0.0		0.0		0.0		0.0		0.0	

Legend for MODES:

M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino) : X = Enabled, . = Disabled
 A(daptive Copy) : D = Disk Mode, W = WP Mode, . = ACp off
 C(onsistency State): X = Enabled, . = Disabled, - = N/A

8.10.4 symcfg list –rdfg all

The (edited) listing below can be used to show every RDF group on a DMX in a single view. It shows:

- If SRDF/A is active or inactive
- If RDF group level consistency is enabled on the rdf group
- If MSC is active for that RDF group

It can also be used to identify Inactive Async groups which appear as **.IS** in the Flags column. This will be used for the basis of the monitoring solution for inactive SRDF/A sessions.

State	Note	Description
XAS	Normal	Active Async session (no MSC control) with consistency enabled
.AS	Normal	Active Async session (no MSC control) with consistency disabled
XIS	Suspended Async group	Inactive Async session (no MSC control) with consistency enabled.
.IS	Suspended Async group	Inactive Async session (no MSC control) with consistency disabled
-IS	Non-Async	Non-Async session (usually synchronous, but not defined here)

```
# symcfg list -rdfg all -sid 0822
```

Symmetrix ID : 000187700822

S Y M M E T R I X R D F G R O U P S										
Local		Remote		Group			RDFA Info			
RA-Grp	LL (sec)	RA-Grp	SymmID	T	Name	Flags LPD	Dir Cfg	Flags CSR	Cycle time	Pri
1 (0)	10	1 (0)	000187721135	S	SRDFS	XX.	F-S	-IS	30	33
2 (1)	10	2 (1)	000187721497	S	SRDFA	XX.	F-S	-IS	30	33
3 (2)	10	3 (2)	000187721497	D	DYN03	XX.	F-S	.AS	30	33
4 (3)	10	4 (3)	000187721497	D	DYN04	XX.	F-S	.AS	30	33
5 (4)	10	5 (4)	000187721497	D	DYN05	XX.	F-S	.AS	30	33
6 (5)	10	6 (5)	000187721497	D	DYN06	XX.	F-S	.AS	30	33
.										
30 (1D)	10	30 (1D)	000187721497	D	DYN30	XX.	F-S	.AS	30	33
31 (1E)	10	31 (1E)	000187721497	D	DYN31	XX.	F-S	.AS	30	33
32 (1F)	10	32 (1F)	000187721497	D	DYN32	XX.	F-S	.AS	30	33
33 (20)	10	33 (20)	000187721497	D	DYN33	XX.	F-S	.IS	30	33
34 (21)	10	34 (21)	000187721497	D	DYN34	XX.	F-S	.IS	30	33
35 (22)	10	35 (22)	000187721135	D	DYN35	XX.	F-S	-IS	30	33
36 (23)	10	36 (23)	000187721135	D	DYN36	XX.	F-S	-IS	30	33
37 (24)	10	37 (24)	000187721135	D	DYN37	XX.	F-S	-IS	30	33
38 (25)	10	38 (25)	000187721135	D	DYN38	XX.	F-S	-IS	30	33
-										
60 (3B)	10	60 (3B)	000187721135	D	DYN60	XX.	F-S	-IS	30	33
61 (3C)	10	61 (3C)	000187721135	D	DYN61	XX.	F-S	-IS	30	33
62 (3D)	10	62 (3D)	000187721135	D	DYN62	XX.	F-S	-IS	30	33
63 (3E)	10	63 (3E)	000187721135	D	DYN63	XX.	F-S	-IS	30	33
64 (3F)	10	64 (3F)	000187721135	D	DYN64	XX.	F-S	-IS	30	33

Legend:

Group (T)ype : S = Static, D = Dynamic

Group Flags :

Prevent Auto (L)ink Recovery : X = Enabled, . = Disabled

Prevent RAs Online Upon (P)ower On: X = Enabled, . = Disabled

Link (D)omino : X = Enabled, . = Disabled

Director (C)onfig : F-S = Fibre-Switched, F-H = Fibre-Hub

G = GIGE, E = ESCON, T = T3, - = N/A

RDFA Flags :

(C)onsistency : X = Enabled, . = Disabled, - = N/A

(S)tatus : A = Active, I = Inactive, - = N/A

(R)DFA Mode : S = Single-session, M = MSC, - = N/A

8.10.5 symcfg list -ra all -switched

Shows the mapping between RDF groups and RA ports. An example can be found in Section 5.3.4.

8.10.6 symevent list

Symevent list gives information on SRDF/A sessions that have been aborted by the system. This may be due to cache exhaustion or communication errors between the DMX.

Note, however, that it was found in testing that:

- Not all SRDF/A stop/start events are recorded
- The RDF group associated with the event are not recorded, only the director, which is of limited use in a multi-session environment

EMC acknowledge these issues and will be fixing them in a future release (post-GA) of the microcode.

symevent list

Symmetrix ID: 000187700822
Time Zone : EST

Detection time	Dir	Src	Category	Severity	Error Num
Thu Feb 17 05:52:33 2005	RF-14C	Symm	RDF	Error	0x004e
SRDF/A Session dropped, no RDF links operational.					
Thu Feb 17 05:52:33 2005	RF-14C	Symm	RDF	Warning	0x0010
No RDF links in an RDF group are operational					
Thu Feb 17 05:52:33 2005	RF-14C	Symm	RDF	Error	0x004e
SRDF/A Session dropped, no RDF links operational.					
Thu Feb 17 05:52:33 2005	RF-14C	Symm	RDF	Warning	0x0010
No RDF links in an RDF group are operational					
Thu Feb 17 05:52:33 2005	RF-14C	Symm	RDF	Error	0x004e
SRDF/A Session dropped, no RDF links operational.					

9 Other Operational Issues

9.1 Use of BCVs at concurrent SRDF/A source sites

In a Symmetrix there are 4 mirror positions available for each volumes. The two mirrored standards consume two of these positions. An established BCV requires another. Each SRDF link requires an additional mirror position.

Hence, it is possible to have either:

- 2-site configuration: Mirrored STD with ONE SRDF link and a BCV
- or
- 3-site configuration: Mirrored STD with TWO SRDF links (i.e. concurrent SRDF)

Hence, it is not possible to use BCVs at the source site of a 3-site concurrent SRDF configuration. The sync and async target sites in a concurrent configuration each only have one SRDF link and hence can continue to use a BCV.

If BCV-type functionality is required at the source site in a 3-site configuration, it will be necessary to use TimeFinder/Clone Technology. Clones have already been certified for use within Citigroup for this purpose.

TimeFinder Clone Solutions Guide	http://gdse.ny.ssmc.com/GDSE_docs/GDSE-05008-SG.pdf
TimeFinder Clone Release Notes	http://gdse.ny.ssmc.com/GDSE_docs/GDSE-05008-RN.pdf

9.2 Using the SRDF/A PIT BCVs for COB testing

As discussed previously in this document, it is recommended that a mirrored R2 be used at the Async target site and hence (due to mirror position limitations) only a single BCV can be established at any one time at that site. It is also recommended that only a single BCV be configured for each R2 at the async target site. Hence, this BCV will need to be used for both SRDF/A resynchronisations and COB testing (for those regions that test COB from a BCV rather than by failing over to the R2).

The following gives some guidance on how to manage the shared usage of these BCVs.

Under normal circumstances, the BCV for each R2 should be established with the R2. This way, the BCV is already synchronized and ready to be split off should an SRDF/A resynchronisation be needed.

The resume_async (See Appendix B) script that can be used to automate the whole SRDF/A resynchronisation process, including production of the BCV copy, assumes that the BCV is already full synchronized with the R2 standard, and will abort by design if this is not the case.

If the BCV is being used for COB test purposes at the time an SRDF/A resynchronisation is required, then it will be necessary to choose one of the following in conjunction with the business unit that owns the application and is doing the COB testing:

a) Stop the COB testing, resynchronize the BCV with the R2 standard and then run the resume_async script. Once resynchronization has completed, COB testing can continue, although the BCV will now contain the latest copy of production data and any changes made to the data during the COB testing will be lost. **Impact:** This will significantly disrupt COB testing.

b) Wait until the COB test is completed. Then resynchronize the BCV with the R2 standard and then run the resume_async script. **Impact:** This will delay resumption of replication.

c) Restart async replication manually without first taking a BCV Point-in-time copy. **Impact:** If a second outage were to occur during the resynchronisation, the only good copy of data at the async site would potentially be the BCV copy being used for COB testing. If this has been modified, there potentially will be no good copies at the SRDF/A target site.

All options have negative points. Which option you take will depend on the perceived criticality of conducting the COB testing versus also having an up-to-date copy of the production data on the R2 during the COB test.

9.3 SRDF/A over long latency links

A number of performance-related issues will impact operations over a long-latency link. Here we focus on the issues that were found in testing with a latency of 250ms, which is of the order of the round trip latency that will be experienced between Singapore and London.

a) Reduced throughput and stability with packet loss at 250ms latency

Limited testing showed that simulated packet loss:

- reduces effective replication throughput as a function of latency
- causes SRDF/A to drop out, with greater sensitivity to packet loss as a function of latency

For further details see Section 6.11 of the SRDF/ A Evaluation document http://gdse.ny.ssmc.com/GDSE_docs/GDSE-05002-E.pdf.

b) Reduced effective bandwidth as a function of latency

Even in an environment with a “perfect” network link (no jitter, no packet loss, no packet re-ordering etc), there is a reduction in throughput as a function of latency. This was expected and is documented more fully in the IPS Evaluation document. Effective bandwidths may be just 10% or less of the maximum potential bandwidth of the link for very long latencies (e.g. 250ms).

c) Cycle time increases with latency for MSC replication

If RDF groups are under MSC control, cycle time will increase beyond the pre-set value in proportion to latency on the network and the number of sessions under MSC control. This was NOT due to lack of bandwidth but due to the overhead of the latency in controlling the SRDF/A process. For further details see Section 6.10.1 of the SRDF/ A Evaluation document.

d) Long runtime for control/query commands

High latency also has a negative impact on the runtime of many control and query operations. This is because many of these commands are written in such a way that they communicate multiple times with the remote DMX for each invocation.

Whilst no issues were observed with the running of these commands, it should be noted that set up and management would take considerably longer for very long distance configurations.

For multiple query operations, the issue can be partially alleviated by running `symcfg scan` first to update the local database, and then running subsequent commands using the `–offline` option. The option causes the command to query cached information in the local database and not the Symmetrix. Consequently, it is significantly faster, with query commands taking less than a second.

For further details see Section 6.11.5 of the SRDF/ A Evaluation document.

9.4 Initial Master Program Load (IMPL) issues

When a DMX does an Initial Master Program Load (IMPL) it resets asynchronous replication modes to synchronous. EMC have been alerted to this fact and are considering a solution. Hence, if a DMX is to be power-cycled, it will be necessary to first record which RDF groups are in async mode prior to the IMPL, and then reset them again back to async mode after the IMPL.

This can be achieved using the script **post_IMPL_reset** (see Appendix B for further details). If this script is run before the IMPL, it will write to `STDOUT` a script that can then be used to reset the relevant RDF groups back to asynchronous mode afterwards.

9.5 Troubleshooting network issues

Lack of appropriate connectivity between sites may lead to problem with SRDF/A replication. At the time of writing, all SRDF/A testing has been conducted over simulated long-distance networks and no implementations have been deployed over real-world networks. Hence, there is limited knowledge on troubleshooting network issues. No end-to-end troubleshooting tools are available at this time.

However, the following advice can be given based on experiences during testing.

9.5.1 `srdfa_health` script

The monitoring script `srdfa_health` (see Appendix B for more details) should be run regularly from cron to alert you of the following issues:

Inactive SRDF/ A sessions – this will indicate any SRDF/A sessions that have dropped. If these have not been manually stopped, then they will have been aborted by the system either due to poor connectivity between the sites or due to cache exhaustion.

SRDF/A cache utilization above a preset limit – this will either indicate an excessive, atypical IO load or a problem with network connectivity giving reduced bandwidth. It accompanied by aborting SRDF/A sessions, but before cache has been exhausted, then network connectivity is the likely cause.

Extended cycle times above a preset threshold – this will either indicated reduced bandwidth availability or more likely excess IO load. It accompanied by a large increase in cache utilization, then increased IO load is the likely cause. Note that in this case, SRDF/A is still running and hence the network is still of sufficient quality to support replication. A temporary increase in loading is the more likely cause.

9.5.2 symevent

Symevent list nominally shows unscheduled SRDF/A start/stop events i.e. SRDF/A sessions that have been aborted by the systems. This may be due to cache exhaustion or communication errors between the DMX.

However, it was found in testing not to record all events. Furthermore, when it does record that SRDF/A has started/stopped, it does not report which RDF group was involved, only the director port, which is of limited use in a multi-session environment. EMC acknowledge these issues and will be fixing them in a future release of the microcode.

An example of symevent list output is shown below. It shows both dropped sessions and when there are no RDF links available for an RDF group.

```
# symevent list -sid 0822
```

```
Symmetrix ID: 000187700822
Time Zone : EST
```

Detection time	Dir	Src	Category	Severity	Error Num
Thu Feb 17 05:52:33 2005	RF-14C	Symm	RDF	Error	0x004e
SRDF/A Session dropped, no RDF links operational.					
Thu Feb 17 05:52:33 2005	RF-14C	Symm	RDF	Warning	0x0010
No RDF links in an RDF group are operational					
Thu Feb 17 05:52:33 2005	RF-14C	Symm	RDF	Error	0x004e
SRDF/A Session dropped, no RDF links operational.					
Thu Feb 17 05:52:33 2005	RF-14C	Symm	RDF	Warning	0x0010
No RDF links in an RDF group are operational					
Thu Feb 17 05:52:33 2005	RF-14C	Symm	RDF	Error	0x004e
SRDF/A Session dropped, no RDF links operational.					

9.5.3 IPS troubleshooting

The IPS performance can be monitored real time via the Element Manager for a particular IPS. Each monitoring/Statistics window will monitor certain portions of the IPS. From the Statistics/Info Pull down menu, an administrator can monitor several different aspects of the IPS performance. The key areas to monitor to make sure both the WAN connections and the IPS gateways are performing at a good level are the following:

- Port Traffic
- Remote Connections
- GE Ports
- Fabric Ports

The most important performance statistics to monitor are "TCP Slow Starts" and "Retransmitted Segments" on the Remote Connections monitoring screen. See "Appendix B" of the McData IPS SAN Router Solutions Guide (http://gdse.ny.ssb.com/GDSE_docs/GDSE-04064-SG.pdf) for details about how these statistics can impact bandwidth performance.

10 Appendix A – Basic Requirements Questionnaire

This questionnaire is not supposed to be exhaustive, but is designed simply to collect the basic requirements information to allow SRDF/A implementation design work to begin and to permit meaningful discussions with EMC. EMC will need to collect much more detailed information for their SRDF/A implementation certification known as Solutions Qualification.

This questionnaire does not cover SRDF/Star implementations.

10.1 Definitions

The following definitions are used within this questionnaire:

COB site - Continuance of Business site

May be used to deal with a variety of failure scenarios including:

- Failure of individual servers
- Failure of Storage subsystems
- Failure of Datacentre infrastructure

Most failures are temporary in nature.

DR site – Disaster Recover site.

Would be used in the event of a disaster that rendered the Production site and local COB site (if one exists) unusable. Note that in some cases (e.g. 2-site configuration), the COB site and the DR site may be one and the same.

Local

Refers to a site that is close enough to the Production datacentre to permit the use of synchronous replication.

Remote

Refers to a site that is not close enough to the Production datacentre to permit the use of synchronous replication

RTO – Recovery Time Objective

Recovery Time Objective is the time within which business functions or applications must be restored (includes time before the disaster is declared and time to perform the tasks). Note that substantially lower RTOs can be achieved if non-zero data loss is acceptable.

RPO – Recovery Point Objective

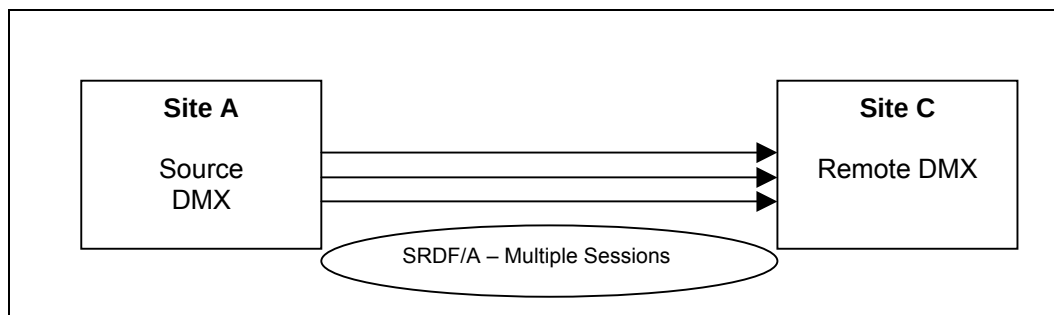
Recovery Point Objective is the point-in-time to which data must be restored to successfully resume processing (often thought of as time between last backup and when outage occurred).

10.2 Questions

Q1. Which of the following two SRDF/A configuration(s) so you intend to deploy?

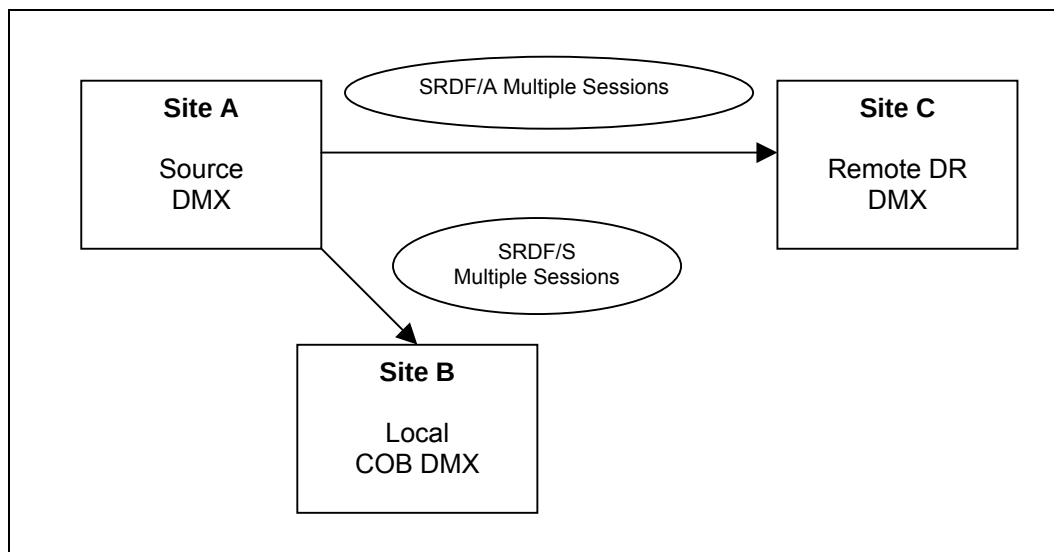
a) 2-site with asynchronous replication between Production and remote COB/DR site.

Replication is via multiple SRDF/A sessions, each controlling the data replication for individual servers/applications and allowing independent failover. All devices in an SRDF/A replication session must be failed over together. Hence, failover independence can only be achieved by putting each application into a different RDF group.



b) 3-site with Synchronous replication between Production and local COB, and Concurrent asynchronous replication between Production and remote DR site.

Source data is simultaneously replicated to two targets. Replication is via multiple sessions, each controlling the data replication for individual servers/applications. Provides zero RPO for COB.



Q2. Approximately how many servers/applications are in scope for SRDF/A replication in your region (state 2- and 3-site requirements separately)?

Note that for a 2-site solution, a single DMX can support up to 62 independent SRDF/A replication sessions to the remote site i.e. 62 independently managed applications. For a 3-site solution, a single DMX can support up to 31 independently managed applications.

Here we define independent as the ability to failover a server/application over to the COB/DR site without affecting replication of other servers/applications, or have that server/application's replication impacted by the failover of other servers/applications

Q3. Approximately what is the total amount of data in scope for each type of SRDF/A replication (state 2- and 3-site requirements separately)?

This, combined with the answer to the previous question determines how many DMX will be required for each implementation; the number of applications that can be supported per DMX will either be limited by the total capacity of the DMX or the number of independent replication sessions if can support.

Q4. Do you require I/O consistency of data at the COB and DR sites to be enforced by EMC technology between applications?

i.e. do you want to ensure that if replication stops for one application, it is also halted for certain other applications to ensure that the data sets for these applications remain point-in-time consistent at the remote site?

If the answer is yes, then each set of apps that need to be IO self-consistent must be documented. See Section 4 for further details of the issues.

Q5. Do you currently use consistency groups to enforce I/O consistency between applications on your Synchronous SRDF setup?

If not, and the answer to the previous question was yes, then consider:

1. If you do really need inter-application consistency enforced at the storage level, as opposed to the application level
2. If you currently have an exposure as no inter-application consistency is enforced today

Q6. What are your RTO requirements?

Q7. What are your RPO requirements?

Synchronous SRDF offers an RPO of zero i.e. zero data loss. SRDF/A offers an RPO of 60 seconds with default settings (and low latency) in the case of storage, network link or site failure. SRDF/A can provide an RPO of zero in the case of server failure.

Q8. What is the distance between your Production and COB/DR site?

If greater than ~80km, use of SRDF/A in place of SRDF/S is mandatory

Q9. What is the round-trip network latency between these sites?

This is required to determine if there are likely to be latency-related issues with the implementation (e.g. reduced effective network throughput, increased likelihood of dropouts, etc)

Q10. Are there any bandwidth limitations between your sites?

SRDF/A requires sufficient bandwidth to accommodate IO write rate. SRDF/A may not be suitable if there is insufficient network bandwidth.

Q11. When a server or application temporarily invokes COB (to the local site), is the temporary loss of replication of that server/application's data from the local COB site (now production) to the remote DR site acceptable?

Note that at the present time, it is not possible to provide replication to the DR site when the application is failed over the COB site. The business needs to be informed of this limitation if they were expecting this functionality at this time.

Q12. Do you use BCVs at your production site?

Traditional BCVs at the source site are not compatible with concurrent SRDF replication of mirrored storage. Hence, it will be necessary convert to using clones instead of BCVs for Point-In-Time copies of source volumes.

10.3 Summary of Basic Requirements Questionnaire results

	Question	Planned Configuration	Result	Notes
Q1	Configuration in scope?	2-site config	Y/N	
		3-site config	Y/N	
Q2	Number of independent applications?	2-site config		
		3-site config		
Q3	Approx. amount of data (GB)?	2-site config		
		3-site config		
Q4	Inter-application I/O consistency required?	2-site config		
		3-site config		Not easily supported
Q5	Consistency groups used today for these applications?	2-site config		
		3-site config		
Q6	DR site RTO requirements	2-site config		
		3-site config		
Q7	DR site RPO requirements	2-site config		
		3-site config		
Q8	Distance between Prod and DR site	Both		
Q9	Round-trip network latency between Prod and DR site	Both		
Q10	Bandwidth limitations between Prod and DR sites	Both		
Q11	Replication required to DR when COB invoked?	2-site config		N/A
		3-site config		Not supported
Q12	BCVs used today at Production site?	2-site config		
		3-site config		Clones required

11 Appendix B – Operational Scripts Summary

The following scripts have been tested in the SRDF/A proof-of-concept test environment. This environment has mirrored R2s at both the Sync and Async target sites. They are not guaranteed to work in all locations due to variations in local practice relating to:

- Device group management (both creation and “replication” of device group definitions to local and remote SAN management servers).
- Formation of synchronous target R2s. These scripts are designed to work with mirrored R2s. If R2s are comprised of unmirrored R2 + BCV, then additional steps will be required to managed these additional BCVs.
- Local naming conventions for logical devices and device groups.

Nevertheless, they do provide fully working scripts which can be used "as is" where the various assumptions are fulfilled. Alternatively, they can be taken and have minor modifications made to them to work within your environment.

Unless otherwise stated, all scripts should be run from a SAN management server located at the production site.

A package containing these scripts can be obtained from the GDS distribution server at

/net/stealth/export1/home1.localhost/sw/PKG/PRsrdfa1.0_A0

11.1 Summary of scripts

*** IMPORTANT ***

The following scripts are for 2-Site SRDF/A and 3-Site non-Star Concurrent SRDF. They are not appropriate for SRDF/Star environments. For operations procedures for SRDF/Star, see the “SRDF/Star for Open Systems” Solutions Guide.

Monitoring scripts

mon_srdfa_inactive	Logs errors in /var/adm/messages if inactive SRDF/A sessions found
mon_srdfa_cycle	Logs errors in /var/adm/messages if current cycle time for any replication group is greater than a preset threshold
mon_srdfa_cache	Logs errors in /var/adm/messages if overall SRDF/A cache utilization is greater than a preset threshold
srdfa_health	Single script combining all three of the above functions (RECOMMENDED)

Adding/removing SRDF/A devices

add_PIT_BCV	Script to add remote BCVs to async R2 standards.
rm_PIT_BCV	Script to remove BCVs from async R2 standards.

2site_add_device	Add devices to existing RDF and device groups and syncs up. Configures PIT BCVs for SRDF/A recovery.
2site_rm_device	Removes subset of devices from RDF and device groups. Deconfigures PIT BCVs used for SRDF/A recovery.
3site_add_device	Add devices to existing RDF and device groups and syncs up. Configures PIT BCVs for SRDF/A recovery.
3site_rm_device	Removes subset of devices from RDF and device groups. Deconfigures PIT BCVs used for SRDF/A recovery.

Recovery scripts	
post_IMPL_reset	Creates a script that can be used to reset asynchronous mode on RDF groups after a DMX IMPL (power cycle)
resume_srdfa	Script to resume SRDF/A replication that take a point-in-time copy before the resynchronisation starts

Failover/failback scripts	
2site_async_failover	Controls failover of async RDF groups
2site_async_failback	Control failback of async RDF groups
3site_sync_failover	Control of failover down the sync leg for <i>entire</i> RDF group in a concurrent configuration
3site_sync_failback	Control of failback down the sync leg for <i>entire</i> RDF group in a concurrent configuration
3site_async_failover	Control of failover down the async leg for <i>entire</i> RDF group in a concurrent configuration
3site_async_failback	Control of failback down the async leg for <i>entire</i> RDF group in a concurrent configuration
2site_async_swap	Controls personality swap of async RDF groups
2site_MSC_failover	Control independent failover of an RDF group participating in multi-session consistency replication
2site_MSC_failback	Control independent failback of an RDF group formerly participating in multi-session consistency replication
2site_CG_failover	Control failover of an <i>entire</i> Composite Group participating in multi-session consistency replication
2site_CG_failback	Control failback of an <i>entire</i> Composite Group participating in multi-session consistency replication

* Note that 3site MSC is not yet supported by EMC and hence no scripts will be provided

3-site sync leg subset failover		
Script	Location script must be run from	Action
3site_subset_failover_phase1	Run from source site.	Suspends Async leg

3site_subset_failover_phase2	Run from Sync target site.	Fails over subset of disks in RDF group
3site_subset_failover_phase3	Run from source site.	Resume Async leg for remaining devices in RDF group
3-site sync leg subset failback		
Script	Location script must be run from	Action
3site_subset_failback_phase1	Run from source site.	Suspends Async leg
3site_subset_failback_phase2	Run from Sync target site.	Fail back subset of disks in RDF group
3site_subset_failback_phase3	Run from source site.	Resume Async leg for all devices in RDF group

11.2 Script `mon_srdfa_inactive`

11.2.1 Overview

This script will monitor for inactive SRDF/A sessions using `symcfg`. It accomplishes this by looking for all RDF groups in asynchronous mode that are not active. The script will work for SRDF/A sessions under both MSC and native DMX replication control.

11.2.2 Invocation and required arguments

```
# Invocation:  mon_srdfa_inactive <SID> <support_team>
#
# Arguments:  <SID> - Symmetrix ID
#             <support_team> - String with support team name
#                               (e.g. "SAN Support team")
#
# Example:    mon_srdfa_inactive 0822 "SAN support team"
```

11.2.3 Usage notes

Script **`mon_srdfa_inactive`** should be run from cron at regular intervals on a Solaris server being used to manage SRDF/A. Once every 5 minutes may be a suitable time interval.

The script will write a message to `/var/adm/messages` using the logger command if one of the criteria for raising an alert is met.

All messages will include the string `SRDFA_ALERT`. You will need to instruct your local Enterprise Monitoring (EMS) Team to monitor the `/var/adm/messages` file on the server running the SRDF/A monitoring script(s) and set their system to raise an alert to the appropriate support team should a message with the string `SRDFA_ALERT` be detected. The contents of the message should also be forwarded by EMS to the support team.

11.3 Script `mon_srdfa_cache`

11.3.1 Overview

This script will monitor the symmetrix wide cache that has been assigned for SRDF/A usage. If the usage exceeds a preset percentage, it will log an error.

11.3.2 Invocation and required arguments

```
# Invocation:  mon_srdfa_cache  <SID> <cache_threshold>
                                           <support_team>
#
# Arguments:  <SID> - Symmetrix ID
#             <cache_threshold> - % cache threshold
#             <support_team>    - String with support team name
#                               (e.g. "SAN Support team")
#
# Example:    mon_srdfa_cache 0822 80 "SAN support team"
```

11.3.3 Usage notes

Script `mon_srdfa_cache` should be run from cron at regular intervals on a Solaris server being used to manage SRDF/A. Once every 5 minutes may be a suitable time interval.

The script will write a message to `/var/adm/messages` using the `logger` command if one of the criteria for raising an alert is met.

All messages will include the string `SRDFA_ALERT`. You will need to instruct your local Enterprise Monitoring (EMS) Team to monitor the `/var/adm/messages` file on the server running the SRDF/A monitoring script(s) and set their system to raise an alert to the appropriate support team should a message with the string `SRDFA_ALERT` be detected. The contents of the message should also be forwarded by EMS to the support team.

11.4 Script `mon_srdfa_cycle`

11.4.1 Overview

This script will monitor the cycle time of all SRDF/A session usage. If the time since the last cycle switch exceeds a preset value for any SRDF/A sessions, it will log an error for each such session.

11.4.2 Invocation and required arguments

```
# Invocation:  mon_srdfa_cycle <SID> <cycle_threshold> <support_team>
#
# Arguments:  <SID> - Symmetrix ID
#             <cycle_threshold> - maximum cycle time threshold
#             <support_team>    - String with support team name
#                               (e.g. "SAN Support team")
#
# Example:    mon_srdfa_cycle 0822 60 "SAN support team"
```

11.4.3 Usage notes

Script **`mon_srdfa_cycle`** should be run from cron at regular intervals on a Solaris server being used to manage SRDF/A. Once every 5 minutes may be a suitable time interval.

The script will write a message to `/var/adm/messages` using the `logger` command if one of the criteria for raising an alert is met.

All messages will include the string `SRDFA_ALERT`. You will need to instruct your local Enterprise Monitoring (EMS) Team to monitor the `/var/adm/messages` file on the server running the SRDF/A monitoring script(s) and set their system to raise an alert to the appropriate support team should a message with the string `SRDFA_ALERT` be detected. The contents of the message should also be forwarded by EMS to the support team.

11.5 Script `srdfa_health`

11.5.1 Overview

This is the primary SRDF/A monitoring script. It embodies all the functionality of the previous 3 scripts and provides:

- Monitoring for inactive SRDF/A sessions
- Monitoring for cache utilization above a preset threshold
- Monitoring for cycle times above a preset threshold

If script `srdfa_health` is used, (recommended) it is not necessary to run the previous three `mon_srdfa_*` scripts.

11.5.2 Invocation and required arguments

```
# Invocation:   srdfa_health <SID> <cache_threshold>
                                     <cycle_threshold> <support_team>
#
# Arguments:   <SID> - Symmetrix ID
#               <cache_threshold> - cache threshold (percentage)
#               <cycle_threshold> - cycle threshold (seconds)
#               <support_team>    - String with support team name
#                                   (e.g. "SAN Support team")
#
# Example:     srdfa_health 0822 50 90 "SAN support team"
```

11.5.3 Usage notes

Script **`srdfa_health`** should be run from cron at regular intervals on a Solaris server being used to manage SRDF/A. Once every 5 minutes may be a suitable time interval.

The script will write a message to `/var/adm/messages` using the `logger` command if one of their criteria for raising an alert is met.

All messages will include the string `SRDFA_ALERT`. You will need to instruct your local Enterprise Monitoring (EMS) Team to monitor the `/var/adm/messages` file on the server running the SRDF/A monitoring script(s) and set their system to raise an alert to the appropriate support team should a message with the string `SRDFA_ALERT` be detected. The contents of the message should also be forwarded by EMS to the support team.

11.6 Script add_PIT_BCV

11.6.1 Overview

This script can be used at the source site to add remote BCVs to a local device group. It is designed to be used when setting up SRDF/A to add the SRDF/ PIT BCVs at the async target site to the source site device group for the relevant RDF group.

The script executes the following steps:

- Associate the remote SRDF/A PIT BCV with the device group
- Attach the BCV to its nominated R2 standard
- Full establish the BCV with the R2 standard
- Remind user to update device group on other local management servers if GNS is not being used
- Remind user to update device group at remote site if GNS remote mirroring is not being used

11.6.2 Invocation and required arguments

```
# Invocation:    add_PIT_BCV <DGName> <async RDF group>
                                     <BCV pairing file> <local SID>
#
# Arguments:    <DGName>             - Device Group Name
#               <async RDF group>    - Async RDF group number
#               <BCV pairing file>   - BCV pairing file
#               <local SID>         - local Symmetrix ID
#
# Example:      add_PIT_BCV DG03 3   ./bcv_pairs_newdisks 0822
```

11.6.3 Usage notes

This script is intended to be used during initial setup of SRDF/A.

The script requires an input file that defines which BCV is to be paired with which async target R2. One row is needed for each STD-BCV pairing. The first entry in a row is the Symmetrix Volume ID of the R2 and the second entry is the Symmetrix Volume ID of the BCV to be associated with the R2. Only metaheads should be included in the file.

For example the following BCV pairing file would pair remote BCV 0521 with R2 00D1 and remote BCV 0525 with R2 00D5.

```
00D1 0521
00D5 0525
```

11.7 Script rm_PIT_BCV

11.7.1 Overview

This script can be used at the source site to remove remote BCVs from a local device group. It is designed to be used when removing SRDF/ PIT BCVs at the async target site.

The script executes the following steps:

- Split PIT BCV from STD to be removed
- Cancel relationships between PIT BCV and STD
- Remove BCV from device group
- Remind user to update device group on other local management servers if GNS is not being used
- Remind user to update device group at remote site if GNS remote mirroring is not being used

11.7.2 Invocation and required arguments

```
# Invocation:   rm_PIT_BCV <DGName> <async RDF group>
                                     <BCV pairing file> <local SID>
#
# Arguments:   <DGName>              - Device Group Name
#               <async RDF group>    - Async RDF group number
#               <BCV pairing file>   - BCV pairing file
#               <local SID>          - local Symmetrix ID
#
# Example:     rm_PIT_BCV DG03 3    ./bcv_pairs_newdisks 0822
```

11.7.3 Usage notes

The script requires an input file that defines which BCVs are paired with which async target R2s. One row is needed for each STD-BCV pairing. The first entry in a row is the Symmetrix Volume ID of the R2 and the second entry is the Symmetrix Volume ID of the BCV to be associated with the R2. Only metaheads should be included in the file.

For example the following BCV pairing file would remove remote BCV 0521 from R2 00D1 and remote BCV 0525 from R2 00D5.

```
00D1 0521
00D5 0525
```

11.8 Script 2site_add_device

11.8.1 Overview

This script can be used to add new devices to an existing RDF group in a 2-site SRDF/A configuration. This would be needed, for example, if additional storage were added to a server after the initial SRDF/A deployment. The script also sets up the remote SRDF/A PIT BCV at the async target site to the local device group as a remote BCV.

The script executes the following steps:

- Disable consistency for device group
- Change mode to adaptive copy
- Add the new device(s) to the RDF group and establish RDF pair
- Add the new device(s) into the local device group
- Set mode back to async replication
- Add the remote SRDF/A PIT BCV into the device group and establish
- Remind user to update device group on other local management servers if GNS is not being used
- Remind user to update device group at remote site if GNS remote mirroring is not being used

11.8.2 Invocation and required arguments

```
# Invocation:      2site_add_device <DGName> <RDF group>
                   <RDF pairsfile> <BCV pairing file> <local SID>
#
# Arguments:      <DGName>           - Device Group Name
#                 <RDF group>        - RDF group number
#                 <RDF pairsfile>    - RDF pairsfile
#                 <BCV pairing file> - BCV pairing file
#                 <local SID>       - local Symmetrix ID
#
# Example:
#
#                 2site_add_device DG19 19
#                 ./pairsfile_DYN19_newdisks ./bcv_pairs_newdisks 0822
```

11.8.3 Usage notes

The script requires a standard RDF pairs file that defines which local and remote standard devices are to be formed into an R1-R2 pair. One row is needed for each R1-R2 pairing. The first entry in a row is the Symmetrix Volume ID of the device in the local Symmetrix to be used as the R1. The second entry is the Symmetrix Volume ID of the device in the remote box to be used as the R2. Only metaheads should be included in the file.

Note that the volume ID for the R1 and R2 will usually be the same. For example

```
0001 0001
0005 0005
```

The script also requires an input file that defines which BCV is to be paired with which async target R2. One row is needed for each STD-BCV pairing. The first entry in a row is the

Symmetrix Volume ID of the R2 and the second entry is the Symmetrix Volume ID of the BCV to be associated with the R2. Only metaheads should be included in the file.

For example the following BCV pairing file would pair remote BCV 0521 with R2 00D1 and remote BCV 0525 with R2 00D5.

```
00D1 0521  
00D5 0525
```

11.9 Script 2site_rm_device

11.9.1 Overview

This script can be used to remove existing devices from an RDF group in a 2-site SRDF/A configuration. This would be needed, for example, if storage were being removed from a server. The script also removes the remote SRDF/A PIT BCV at the async target site to the local device group as a remote BCV.

The script executes the following steps:

- Split PIT BCV from STD to be removed
- Cancel relationships between PIT BCV and STD
- Remove BCV from device group
- Suspend asynchronous replication
- Remove STDs from device group
- Delete RDF pairing relationship
- Resume SRDF replication and enable consistency
- Remind user to update device group on other local management servers if GNS is not being used
- Remind user to update device group at remote site if GNS remote mirroring is not being used

11.9.2 Invocation and required arguments

```
# Invocation:      2site_rm_device <DGName> <RDF group>
                   <RDF pairsfile> <BCV pairing file> <local SID>
#
# Arguments:      <DGName>           - Device Group Name
#                  <RDF group>        - RDF group number
#                  <RDF pairsfile>    - RDF pairsfile
#                  <BCV pairing file> - BCV pairing file
#                  <local SID>        - local Symmetrix ID
#
# Example:
#
#                  2site_rm_device DG19 19
#                  ./pairsfile_DYN19_newdisks ./bcv_pairs_newdisks 0822
```

11.9.3 Usage notes

The script requires a standard RDF pairs file that defines which local and remote standard devices form the R1-R2 pairs to be removed. One row is needed for each R1-R2 pairing. The first entry in a row is the Symmetrix Volume ID of the device in the local Symmetrix used as the R1. The second entry is the Symmetrix Volume ID of the device in the remote box used as the R2. Only metaheads should be included in the file.

Note that the volume ID for the R1 and R2 will usually be the same. For example

```
0001 0001
0005 0005
```

The script also requires an input file that defines which BCVs are paired with which async target R2s. One row is needed for each STD-BCV pairing. The first entry in a row is the Symmetrix

Volume ID of the R2 and the second entry is the Symmetrix Volume ID of the BCV to be associated with the R2. Only metaheads should be included in the file.

For example the following BCV pairing file would remove remote BCV 0521 from R2 00D1 and remote BCV 0525 from R2 00D5.

```
00D1 0521  
00D5 0525
```

11.10 Script 3site_add_device

11.10.1 Overview

This script can be used to add new devices to an existing RDF group in a 3-site SRDF/A configuration. This would be needed, for example, if additional storage were added to a server after the initial SRDF/A deployment. The script also sets up the remote SRDF/A PIT BCV at the async target site to the local device group as a remote BCV.

The script executes the following steps:

- Disable consistency for async leg
- Change async mode to adaptive copy
- Add the new device(s) to the Async RDF group and establish Async RDF pairs
- Add the new device(s) into the local device group
- Set async leg mode back to async and enable consistency
- Add the new device(s) to the sync RDF group and establish sync RDF pairs
- Add the remote SRDF/A PIT BCV into the device group and establish
- Remind user to update device group on other local management servers if GNS is not being used
- Remind user to update device group at remote site as GNS remote mirroring is not supported in concurrent configurations.

11.10.2 Invocation and required arguments

```
# Invocation:  3site_add_device <DGName> <Sync RDF group>
               <Async RDF group> <RDF pairsfile>
               <BCV pairing file> <local SID>

#
# Arguments:   <DGName>           - Device Group Name
#              <Sync RDF group>    - Sync leg RDF group number
#              <Async RDF group>   - Async leg RDF group number
#              <RDF pairsfile>     - RDF pairsfile
#              <BCV pairing file>  - BCV pairing file
#              <local SID>        - local Symmetrix ID
#
# Example:
#
# 3site_add_device DG03 64 3 ./pairsfile_DYN03_newdisks
                           ./bcv_pairs_newdisks 0822
```

11.10.3 Usage notes

The script requires a standard RDF pairs file that defines which local and remote standard devices are to be formed into an R1-R2. **The same file is used for both legs.** One row is needed for each R1-R2 pairing. The first entry in a row is the Symmetrix Volume ID of the device in the local Symmetrix to be used as the R1. The second entry is the Symmetrix Volume ID of the device in the remote box to be used as the R2. Only metaheads should be included in the file.

Note that the volume ID for the R1 and R2 will usually be the same. For example

```
0001 0001
0005 0005
```

The script also requires an input file that defines which BCV is to be paired with which async target R2. One row is needed for each STD-BCV pairing. The first entry in a row is the Symmetrix Volume ID of the R2 and the second entry is the Symmetrix Volume ID of the BCV to be associated with the R2. Only metaheads should be included in the file.

For example the following BCV pairing file would pair remote BCV 0521 with R2 00D1 and remote BCV 0525 with R2 00D5.

```
00D1 0521  
00D5 0525
```

11.11 Script 3site_rm_devices

11.11.1 Overview

This script can be used to remove existing devices from an RDF group in a 3-site SRDF/A configuration. This would be needed, for example, if storage were being removed from a server. The script also removes the remote SRDF/A PIT BCV at the async target site to the local device group as a remote BCV.

The script executes the following steps:

- Split PIT BCV from STD to be removed
- Cancel relationships between PIT BCV and STD
- Remove BCV from device group
- Suspend async replication
- Suspend sync replication
- Remove STDs from device group
- Delete async RDF pairing relationship
- Delete sync RDF pairing relationship
- Resume async replication and enable consistency
- Resume sync replication
- Remind user to update device group on other local management servers if GNS is not being used
- Remind user to update device group at remote site as GNS remote mirroring is not supported in concurrent configurations.

11.11.2 Invocation and required arguments

```
# Invocation:      3site_rm_device <DGName> <Sync RDF group>
                  <Async RDF group> <RDF pairsfile>
                  <BCV pairing file> <local SID>

#
# Arguments:      <DGName>           - Device Group Name
#                  <Sync RDF group>   - Sync leg RDF group number
#                  <Async RDF group>   - Async leg RDF group number
#                  <RDF pairsfile>     - RDF pairsfile
#                  <BCV pairing file>  - BCV pairing file
#                  <local SID>        - local Symmetrix ID
#
# Example:
#
# 3site_rm DG03 64 3 ./pairsfile_DYN03_newdisks
                  ./bcv_pairs_newdisks 0822
```

11.11.3 Usage notes

The script requires a standard RDF pairs file that defines which local and remote standard devices form the R1-R2 pairs to be removed. **The same file is used for both legs.** One row is needed for each R1-R2 pairing. The first entry in a row is the Symmetrix Volume ID of the device in the local Symmetrix used as the R1. The second entry is the Symmetrix Volume ID of the device in the remote box used as the R2. Only metaheads should be included in the file.

Note that the volume ID for the R1 and R2 will usually be the same. For example

```
0001 0001
0005 0005
```

The script also requires an input file that defines which BCVs are paired with which async target R2s. One row is needed for each STD-BCV pairing. The first entry in a row is the Symmetrix Volume ID of the R2 and the second entry is the Symmetrix Volume ID of the BCV to be associated with the R2. Only metaheads should be included in the file.

For example the following BCV pairing file would remove remote BCV 0521 from R2 00D1 and remote BCV 0525 from R2 00D5.

```
00D1 0521
00D5 0525
```

11.12 Script post_IMPL_reset

11.12.1 Overview

When a DMX does an Initial Master Program Load (IMPL) it resets asynchronous replication modes to synchronous. Hence, if a DMX is to be power-cycled, it will be necessary to first record which RDF groups are in asynchronous mode prior to the IMPL, and then reset them again back to async mode after the IMPL.

11.12.2 Invocation and required arguments

```
# Invocation:      post_IMPL_reset  <SID>
#
# Arguments:      <SID> - Symmetrix ID
#
# Example:  post_IMPL_reset 0822
```

11.12.3 Usage notes

This script must be run before the IMPL. It will write to STDOUT a bespoke command file containing the appropriate symrdf set mode commands that can then be run to reset the relevant RDF groups back to asynchronous mode afterwards.

11.13 Script resume_srdfa

11.13.1 Overview

Prior to starting resynchronisation after SRDF/A link recovery, a BCV copy of the R2 volumes should be split off (consistently) to ensure that there is a consistent (albeit aging) point-in-time copy at all times in the remote site during the re-synchronization process when the R2 will be inconsistent.

If the link between sites is down for an extended period, and there is a large build up of invalid tracks on the R1 side, resumption of the link with SRDF/A (the default behaviour in Open Systems) can cause a heavy surge of link traffic, created by the large 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 a large build up of invalid tracks.

This script handles the creation of the PIT copy and also the switch into and out of adaptive copy mode before and after resynchronisation takes place.

The script executes the following steps:

- Verify remote BCV is fully established
- Split remote BCV
- Wait until finished splitting in the background
- Set mode to adaptive copy
- Establish RDF
- Wait until RDF is full synced up
- Set mode to async mode
- Enable consistency
- Establish remote BCV
- Wait until remote BCV fully established before exiting

11.13.2 Invocation and required arguments

```
# Invocation:  resume_async <DGName> <RDF group>
#
# Arguments:   <DGName>      - Device Group Name
#              <RDF group>   - RDF group number of async link
#
# Example:    resume_async App1 4
```

11.13.3 Usage notes

The script requires that the PIT BCV is established. Note this script will gracefully exit if BCVs are not established as this may indicate that the SRDF/A outage occurred during a previously incomplete resync operation and hence the BCV should NOT be established against a potentially inconsistent R2. Alternatively it may indicate that they are being used for COB test purposes.

11.14 Script 2site_async_failover

11.14.1 Overview

This script executes an SRDF failover down the async leg in a 2-site SRDF/A configuration.

The script executes the following steps:

- Temporarily disable consistency on the device group
- Confirming no outstanding tracks not committed to async R2 side
- Failover the device group

11.14.2 Invocation and required arguments

```
# Invocation:      2site_async_failover <DGname>
#
# Arguments:      <DGName>      - Device groups to be failed over
#
# Example:        2site_async_failover DG21
```

11.14.3 Usage notes

Trivial script included for completeness.

11.15 Script 2site_async_failback

11.15.1 Overview

This script executes an SRDF failback down the async leg in a 2-site SRDF/A configuration.

The script executes the following steps:

- Failback RDF group
- Wait until there are no invalid tracks on the async leg
- Renable consistency

11.15.2 Invocation and required arguments

```
# Invocation:      2site_async_failback <DGname>
#
# Arguments:       <DGName>      - Device groups to be failed back
#
# Example: 2site_async_failback DG21
```

11.15.3 Usage notes

The script will loop whilst the R1 is updating from the R2 and not exit until this process has completed.

11.16 Script 3site_sync_failover

11.16.1 Overview

This script executes an SRDF failover down the sync leg in a 3-site SRDF/A configuration.

The script executes the following steps:

- Confirm no outstanding tracks not committed to async R2 side
- Suspend async leg
- Execute failover of sync leg
- Confirm status of both legs with a symrdf query command

11.16.2 Invocation and required arguments

```
# Invocation:      3site_sync_failover <DGName> <sync leg RDF group>
#                  <async leg RDF group>
#
# Arguments:      <DGName>                - Device Group Name
#                  <sync leg RDF group>    - RDF group number of sync link
#                  <async leg RDF group>   - RDF group number of async link
#
# Example:  3site_sync_failover DG03 64 3
```

11.16.3 Usage notes

No special notes.

11.17 Script 3site_sync_failback

11.17.1 Overview

This script executes an SRDF failback down the sync leg in a 3-site SRDF/A configuration.

The script executes the following steps:

- Failback sync leg
- Wait until there are no invalid tracks on the sync leg
- Resume async leg replication
- Enable consistency on the async leg
- Confirm status of both legs with a symrdf query command

11.17.2 Invocation and required arguments

```
# Invocation:      3site_sync_failback <DGName> <sync leg RDF group>
#                  <async leg RDF group>
#
# Arguments:      <DGName>                - Device Group Name
#                  <sync leg RDF group>    - RDF group number of sync link
#                  <async leg RDF group>   - RDF group number of async link
#
# Example:        3site_sync_failback DG03 64 3
```

11.17.3 Usage notes

The script will loop whilst the R1 is updating from the Sync R2 and not resume replication on the async leg until that update has completed.

11.18 Script 3site_async_failover

11.18.1 Overview

This script executes an SRDF failover down the async leg in a 3-site SRDF/A configuration.

The script executes the following steps:

- Suspend Sync leg
- Confirm no outstanding tracks not committed to async R2 side
- Execute failover of async leg
- Confirm status of both legs with a symrdf query command

11.18.2 Invocation and required arguments

```
# Invocation:  3site_async_failover <DGName> <sync leg RDF group>
#              <async leg RDF group>
#
# Arguments:   <DGName>           - Device Group Name
#              <sync leg RDF group> - RDF group number of sync link
#              <async leg RDF group> - RDF group number of async link
#
# Example:     3site_async_failover DG03 64 3
```

11.18.3 Usage notes

No special notes.

11.19 Script 3site_async_failback

11.19.1 Overview

This script executes an SRDF failback down the async leg in a 3-site SRDF/A configuration.

The script executes the following steps:

- Failback async link
- Enable consistency on async leg
- Wait until there are no invalid tracks on the async leg
- Resume sync leg replication
- Confirm status of legs with a symrdf query

11.19.2 Invocation and required arguments

```
# Invocation: 3site_async_failback <DGName> <sync leg RDF group>
#              <async leg RDF group>
#
# Arguments:   <DGName>           - Device Group Name
#              <sync leg RDF group> - RDF group number of sync link
#              <async leg RDF group> - RDF group number of async link
#
# Example:     3site_async_failback DG03 64 3
```

11.19.3 Usage notes

The script will loop whilst the R1 is updating from the Async R2 and not resume replication on the Sync leg until that update has completed.

11.20 Script 2site_async_swap

11.20.1 Overview

With dynamic RDF, you can swap the R1/R2 personality of RDF pairs. Source R1 devices become target R2 devices, and vice versa. Swapping SRDF devices allows the original R2 side to take over operations while retaining a remote mirror on the original R1 side i.e. the original R1 becomes an R2 and vice versa (see figure below).

This is used in some regions to ensure that when a failover to COB has occurred, that off-site COB is still available (i.e. back to the original production site). For example, in the diagram below, if an application that usually runs in Site A has failed over to Site B due to server hardware failure, then personality swapping ensures that synchronous off-site replication still continues.

This script conducts a personality swap of an RDF group running SRDF/A

The script executes the following steps:

- Temporarily disable consistency on the device group
- Confirming no outstanding tracks not committed to async R2 side
- Execute failover
- Execute swap
- Execute establish
- Enable consistency on the device group
- Print out current RDF group type

11.20.2 Invocation and required arguments

```
# Invocation:      2site_async_swap <DGname>
#
# Arguments:      <DGName>      - Device groups to be swapped
#
# Example:  2site_async_swap DG21
```

11.20.3 Usage notes

Note that “fast-swap” (failover –establish) is not supported on SRDF/A enabled devices and hence is not used in this script.

11.21 Script 2site_MSC_failover

11.21.1 Overview

One of the benefits of Multi-Session Consistency (MSC) in an SRDF/A environment is that it allows one to guarantee IO consistency at the R2-end between RDF groups (and hence individual applications), whilst still permitting failover independence for each application.

This script executes an SRDF failover ***of a single RDF group*** participating in Multi-Session Consistency (MSC) replication down the async leg in a 2-site SRDF/A configuration.

This is an unusual process, as it requires that the RDF group to be failed over be removed from the Composite Group being used for MSC replication before the failover can take place. MSC replication of the remaining RDF groups in the re-defined Composite group is then restarted.

The script executes the following steps:

- Temporarily disable consistency on the composite group (CG)
- Remove devices to be failed over from CG
- Re-enable consistency on the composite group (CG)
- Confirm no outstanding tracks not committed to async R2 side
- Failover the RDF group (DG)

11.21.2 Invocation and required arguments

```
# Invocation:  2site_MSC_failover <CGName> <DGname>
                                     <RDF group to failover>
#
# Arguments:  <CGName>      - Composite Group Name
#              <DGname>      - Device group to be failed over
#              <RDF group to failover>
#                                     - RDF group number to be failed over
#
# Example:    2site_MSC_failover App1CG DG22 22
```

11.21.3 Usage notes

No special notes

11.22 Script 2site_MSC_failback

11.22.1 Overview

One of the benefits of Multi-Session Consistency (MSC) in an SRDF/A environment is that it allows one to guarantee IO consistency at the R2-end between RDF groups (and hence individual applications), whilst still permitting failover independence for each application.

This script executes an SRDF failback ***of a single RDF group*** formerly participating in Multi-Session Consistency (MSC) replication down the async leg in a 2-site SRDF/A configuration.

This is an unusual process, as it requires that the RDF group to be failed over be replaced into the Composite Group being used for MSC replication after the failover has taken place. MSC replication of the all RDF groups in the Composite Group is then restarted.

The script executes the following steps:

- Failback the device group (DG)
- Wait until there are no invalid tracks on the async leg
- Temporarily disable consistency on the composite group (CG)
- Add devices from RDF group just failed back into CG
- Re-enable consistency on the composite group (CG)

11.22.2 Invocation and required arguments

```
# Invocation: 2site_MSC_failback <CGName> <DGname>
               <RDF group to failback>
#
# Arguments:  <CGName>      - Composite Group Name
#              <DGname>      - Device groups to be failed back
#              <RDF group to failback>
#                               - RDF group number to be failed back
#
# Example: 2site_MSC_failback App1CG 22
```

11.22.3 Usage notes

No special notes

11.23 Script 2site_CG_failover

11.23.1 Overview

An entire Composite Group can also be failed over. This would be required in the case where there has been an issue at the source site and DR is being invoked. In such cases, replication usually already have ceased due to the MSC Consistency Group tripping. Failover via the Composite group (rather than via the underlying RDF groups) ensures consistency at the R2 end.

This script executes a Composite Group failover down the async leg in a 2-site SRDF/A configuration.

The script executes the following steps:

- Failover the composite group

11.23.2 Invocation and required arguments

```
# Invocation:      2site_CG_failover <CGname>
#
# Arguments:      <CGName>                - Composite group to be failed
over
#
# Example:  2site_CG_failover App1CG
```

11.23.3 Usage notes

When MSC replication is halted due to the MSC Composite Group tripping, a clean up process is required to determine if the receive Delta Sets (on the Target DMX) for each RDF group should be applied to the R2 or discarded. A description of this process is beyond the scope of the present document, but can be found in Appendix H of the SRDF/A Evaluation document.

The failover command uses the `-force` and `-symforce` options. These are sometime required if the failover is occurring after a rolling disaster when the MSC has tripped and replication has stopped. The SRDF/A MSC cleanup process can result in Delta Sets being discarded on the R2 side leaving the Symmetrix in a state with invalids on both sides. For further details see Appendix H of the SRDF/A Evaluation document.

11.24 Script 2site_CG_failback

11.24.1 Overview

This script executes a Composite Group failback down the async leg in a 2-site SRDF/A configuration.

The script executes the following steps:

- Failback the composite group

11.24.2 Invocation and required arguments

```
# Invocation:      2site_CG_failback <CGname>
#
# Arguments:      <CGName> - Composite group to be failed back
#
# Example:       2site_CG_failback App1CG
```

11.24.3 Usage notes

The `-force` flag is needed as consistency is still necessarily enabled on the Composite Group.

11.25 3site_subset_failover

11.25.1 Overview

Subset failover is ***extremely*** complex as it requires actions on management stations on both the local and the remote sites. Each phase must be completed before moving onto the next.

Note that subset failover on the synchronous leg of a concurrent SRDF configuration is not recommended either by EMC nor GDSE, but is necessarily being supported to meet claimed business requirements.

A terminal session will be required on both a local SAN management server (for phases 1 and 3) and also on a SAN management server at the remote sync target site (for phase 2). If phases were to be controlled from the local server, Phase 2 activities would need to be undertaken using device files.

Phase 1 script - runs at source site. Suspend SRDF/A for RDF group.

Note that with subset failover, there may still be outstanding tracks not committed to async R2 side for the async RDF group as other servers using the same group may still be running. Hence, this is NOT tested for in this script.

Instead the async is suspended regardless at the end of the current cycle.

This script executes the following steps:

- Disable consistency on the SRDF/A leg
- Suspend async leg
- Confirm status of both legs with a symrdf query command

Phase 2 script - runs at SYNC TARGET site. Failover subset of disks in RDF group on the SYNC leg.

This script executes the following steps:

- Subset failover the disks belonging that are part of subset
- Confirm status of RDF for the subset device group with a symrdf query command

Phase 3 script - runs at source site. Resumes SRDF/A for remaining disks in RDF group

Note that phase 3 is optional as the business requirements may be to preserve an IO consistency PIT copy at the SRDF/A target site for the entire RDF group (application). Please refer to section 8.6.1 before deciding to run Phase 3 failover script.

This script executes the following steps:

- Resume async leg
- Enable consistency on the SRDF/A leg
- Confirm status of both legs with a symrdf query command

11.25.2 Invocation and required arguments

Phase 1 Script to be run at source site

```
# Invocation: 3site_subset_failover_phase1 <DGName>
               <async leg RDF group>
#
# Arguments:  <DGName>           - Device Group Name
#             <async leg RDF group> - RDF group number of async link
#
# Example:    3site_subset_failover_phase1 DG03 3
```

When Phase 1 completed move onto Phase 2

Phase 2 Script to be run at SYNC target site

```
# Invocation: 3site_subset_failover_phase2 <Subset DGName> <sync
leg RDF group>
#
# Arguments:  <Subset DGName>     - Device Group Name
#             <sync leg RDF group> - RDF group number of sync link
#
# Example:    3site_subset_failover_phase2 server1 4
```

When Phase 2 completed move onto Phase 3

Phase 3 Script to be run at source site

```
# Invocation: 3site_subset_failover_phase3 <DGName> <async leg RDF
group>
#
# Arguments:  <DGName>           - Device Group Name
#             <async leg RDF group> - RDF group number of async link
#
# Example:    3site_subset_failover_phase3 DG03 3
```

11.25.3 Usage notes

Subset failover and failback is very complex and hence error prone and should be avoided if at all possible. You have been warned.

Phase 3 failover is optional as the business requirements may be to preserve an IO consistency PIT copy at the SRDF/A target site for the entire RDF group (application). Please refer to section 8.6.1 before deciding to run Phase 3 failover script.

11.26 3site_subset_failback

11.26.1 Overview

Subset failback is **extremely** complex as it requires actions on management stations on both the local and the remote sites. Each phase must be completed before moving onto the next.

Note that subset failover on the synchronous leg of a concurrent SRDF configuration is not recommended either by EMC nor GDSE, but is necessarily being supported to meet claimed business requirements.

A terminal session will be required on both a local SAN management server (for phases 1 and 3) and also on a SAN management server at the remote sync target site (for phase 2). If phases were to be controlled from the local server, Phase 2 activities would need to be undertaken using device files.

Phase 1 script - runs at source site. Suspend SRDF/A for RDF group.

Phase1 failback should only be run if Phase 3 failover was executed during the original subset failover.

Note that with subset failback, there may still be outstanding tracks not committed to async R2 side for the async RDF group as other servers using the same group may still be running. Hence, this is NOT tested for in this script.

Instead the async is suspended regardless at the end of the current cycle.

This script executes the following steps:

- Disable consistency on the SRDF/A leg
- Suspend async leg
- Confirm status of both legs with a symrdf query command

Phase 2 script - runs at SYNC TARGET site. Failback subset of disks in RDF group on the SYNC leg.

This script executes the following steps:

- Subset failover the disks belonging that are part of subset
- Wait until there are no invalid tracks on the sync leg
- Confirm status of RDF for the subset device group with a symrdf query command

Phase 3 script - runs at source site. Resumes SRDF/A for remaining disks in RDF group

- Resume async leg
- Enable consistency on the SRDF/A leg
- Confirm status of both legs with a symrdf query command

11.26.2 Invocation and required arguments

Phase 1 Script to be run at source site

```
# Invocation: 3site_subset_failback_phase1 <DGName>
               <async leg RDF group>
#
# Arguments:  <DGName>                - Device Group Name
#             <async leg RDF group> - RDF group number of async link
#
# Example:    3site_subset_failback_phase1 DG03 3
```

When Phase 1 completed move onto Phase 2

Phase 2 Script to be run at SYNC target site

```
# Invocation: 3site_subset_failback_phase2 <Subset DGName> <sync
leg RDF group>
#
# Arguments:  <Subset DGName>          - Device Group Name
#             <sync leg RDF group>    - RDF group number of sync link
#
# Example:    3site_subset_failback_phase2 server1 4
```

When Phase 2 completed move onto Phase 3

Phase 3 Script to be run at source site

```
# Invocation: 3site_subset_failback_phase3 <DGName> <async leg RDF
group>
#
# Arguments:  <DGName>                - Device Group Name
#             <async leg RDF group> - RDF group number of async link
#
# Example:    3site_subset_failback_phase3 DG03 3
```

11.26.3 Usage notes

Subset failover and failback is very complex and hence error prone and should be avoided if at all possible. You have been warned.

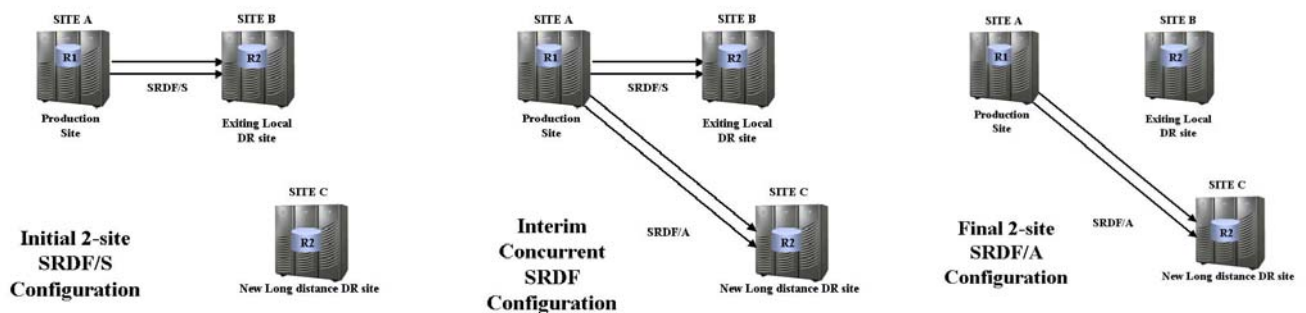
Phase1 failback should only be run if Phase 3 failover was executed during the original subset failover.

12 Appendix C - Migrating existing source DMX from 2-site SRDF/S to 2-site SRDF/A, using Concurrent SRDF

12.1 Introduction

The main documentation describes how to configure a 2-site SRDF/A on a DMX that is either new or not previously running SRDF/S.

This appendix gives an overview of the main steps required to convert an existing metro-distance 2-site SRDF/S configuration into a 2-site long distance SRDF/A configuration, in the scenario where the Production (source) Symmetrix location is not changing. Concurrent SRDF is used to “swing” the DR function over to the remote site.



Overview of Concurrent SRDF “swing” process

Site A is the production site, before and after the migration. The DMX at Site B (existing DR site) can be re-deployed/retired at the end of the process. The DMX at Site C (new DR site) is a new purchase, or a redeployed DMX from elsewhere. The DMX units at Sites A and C must be “balanced” systems (see Section 4.7 for further details).

The setup of existing deployments will vary in their specific, so a fully comprehensive step-by-step guide cannot be created. The main steps are listed here, but it must be understood that variations will occur in actual migrations. Additional or different steps may be needed at your site. The purpose of this appendix is to give an overview of the anticipated steps required, but not to provide a fully documented procedure. No migrations of this type have yet taken place (at the time of writing) and hence the procedures have not been tested.

Although the diagram may make the process look simple, there are many complications that need to be understood and addressed. The following are in general issues that apply only to this migration scenario. Generic DMX deployment steps are not covered in detail here, although references to the main documentation are given.

Note that in many cases it may be simpler to build a new SRDF/A environment and migrate applications into that environment. Also, the process described here is only really applicable when the entire Symmetrix is changing exclusively to 2-Site SRDF/A i.e. all applications are migrating.

12.2 SAN environment planning preparation

12.2.1 General planning and preparation

1) Ensure Production is running on the DMX, which is to be used for SRDF/A source (Site A). If Production is currently running out of Site B, Production will first need to be permanently swapped to Site A.

2) Determine IP bandwidth and additional SRDF/A cache requirements by measuring the I/O profile of existing application as described in Section 4.3. *This is a non-trivial exercise undertaken by EMC, which must be started as soon as any migration project is initiated.*

3) Ensure that the target symmetrix can be configured to be suitable as an SRDF/A target.

- The SRDF/A target unit at Site C should have disk of the same size and rotational speed as the source box for the R2 units. For example, if you are running with 146GB at the source site, one should not use 300GB disks at the R2 end. The RAID type at both ends should also match. It is essential that SRDF/A systems are “balanced” i.e. same performance at both end, or that the R2 end is at least as fast as the R1 end. Do not put larger, slower disks into Site C for the R2 devices.
- SRDF/A PiT copy BCVs will need to be installed in the Site C unit. The Site C unit must have sufficient capacity to provide all the R2 volumes and their associated SRDF/A PiT copy volumes.

12.2.2 Network infrastructure

1) Ensure sufficient dedicated SRDF/A network bandwidth is available between Sites A and C. See Section 4.5 and references therein for further information on this major task.

2) Install IPS Eclipse infrastructure at Sites A and C. See Section 5.2 and references therein for further information.

12.2.3 Symmetrix changes (not in any order of priority)

1) Ensure that DMX at all sites are running an identical version of 5x71/5x72 microcode (latest GDSE certified versions). There are three certified configurations for the SRDF/A leg:

- a) DMX3 → DMX3 (Preferred configuration)
- b) DMX2 → DMX2
- c) DMX2 → DMX3 (Not preferred. Known issues and not tested by GDSE)

Minimum microcode versions for DMX2 → DMX3 SRDF:

DMX2 - 5671.33.xx or later

DMX3 - 5771.81.90 or later

There are also operational constraints imposed by not running a balanced system. For example, an SRDF swap may result in SRDF/A failing as the R2 will be running on a slower system than the R1. Hence, where possible, SRDF/A between dissimilar units should be avoided.

Note that DMX3 → DMX2 SRDF/A is not supported by GDSE.

2) Determine which RAs on the Symmetrix will be used for SRDF/A replication:

- Determine if there are sufficient spare “a” port Ras available on the source Symmetrix for SRDF/A usage. If possible, use spare “a” ports to avoid load on shared CPU. See Section 4.7 to determine how many RAs are required. Note that RAs currently in use for SRDF/S cannot simultaneously be used for SRDF/A.
- “b” ports cannot be used on the source Symmetrix to provide SRDF/A connectivity in parallel to “a” ports being used for SRDF/S. “b” ports are usually not in use on DMX but as they share the same processor as the neighbouring “a” port, they cannot be used for SRDF/A.
- If there is insufficient “a” ports available for RAs, additional directors may be needed to support the SRDF/A leg and execute the swing.

At this time, it is worth nominating which RDF groups numbers will be used for each RA to ensure that there are no conflicts.

3) Install additional “SRDF/A” cache (quantity determined by EMC through the aforementioned sizing exercise) into the Source DMX (Site A) and SRDF/A target DMX (Site C).

4) Additional disk space may need to be installed in the SRDF/A target DMX for SRDF/A point-in-time (TimeFinder) copies to be used during SRDF/A resynchronizations. A TimeFinder BCV is needed for every R2. See Section 5.4 for further details.

5) Define and document which TimeFinder BCV will be associated with each R2 device.

6) Ensure that all three DMX are enabled for dynamic RDF. The following four flags must be enabled on the DMX.

```
Switched RDF Configuration State      : Enabled
Concurrent RDF Configuration State    : Enabled
Dynamic RDF Configuration State       : Enabled
Concurrent Dynamic RDF Configuration : Enabled
```

See Section 5.3.2 for further details.

7) Ensure that all R1/R2 devices in all three DMX are designated as RDF capable.

EMC can change the state of these devices in the bin files. Alternatively, it can be changed by Citigroup staff using symconfigure. The syntax for the symconfigure command file is:

```
convert rdf dev <start-range>:<end-range> to dynamic;
```

- 8) Each device that will be migrated with concurrent SRDF needs to be allocated a pair of RFs.
- 9) If Symmetix Optimizer is being used, it should be disabled. Concurrent SRDF will consume the last remaining mirror position on the source unit that Optimizer uses.
- 10) Connect the SRDF/A RAs to the IPS infrastructure at Sites A and C.

12.2.4 SAN management server changes

- 1) Ensure that the correct SRDF licenses have been purchased and are have been installed on the SAN management workstation. The source DMX will need an additional SRDF/A license to allow concurrent SRDF. The new DMX at Site C should be purchased with the Citigroup HA license. See Section 5.2.1 for further details on licensing requirements.
- 2) Ensure that all SAN management servers at Site A and Site C are running the latest GDSE certified version of Solutions Enabler.
- 3) Ensure that the standard entries required for SRDF/A are set correctly in `/etc/system`. See Section 5.3.1 for specific details.
- 4) Ensure that the entries required for SRDF/A are set correctly in `/var/symapi/config/options` and `/var/symapi/config/daemon_options` See Sections 5.1.4 and 5.1.5 for specific details.
- 5) Ensure that there are sufficient Gatekeepers allocated to each SAN management server to support the SRDF/A configuration required. See Section 5.1.6 for details.
- 6) Set up SRDF/A monitoring should be set up as documented in Section 5.8.

12.3 Application migration planning

This section covers application-level migration planning issues that relate to storage. The following comments and tasks apply to each application.

12.3.1 General comments

- 1) Applications can and should be migrated independently of each other, to avoid an issue with one migration impacting another.
- 2) All volumes owned by each server should be documented. The VCMDB LUN masking DB should be used. Note that it is not possible to query a VCMDB remotely across an SRDF link, and local management servers should be used to get this information.
- 3) The BAU operational processes required during the interim concurrent SRDF phase are complex, and hence the amount of time that concurrent SRDF is used for a given application should be minimized (i.e. not left on for many weeks). It is recommended that no disk should be added/removed from an application once the migration process has begun (ideally including the planning phases) to minimize the impact on operational processes. The SAN management team must take responsibility for ensuring that this does not happen.
- 4) Any new applications added to the Symmetrix once migrations have begun should be implemented immediately as 2-site SRDF/A.
- 5) It is recommended that large applications be moved first, as subsequent bulk-copy migrations will impact those applications already migrated and using the SRDF/A links.

12.3.2 Existing BCVs on source Symmetrix

- 1) Confirm that TimeFinder/Mirror BCVs are not being used for the application on source DMX (Site A). It is not possible to run concurrent SRDF and establish TimeFinder/Mirror BCVs at Site A at the same time. Any BCVs being used will need to be split off before concurrent SRDF is setup. It will not be possible to establish them until after the migration has completed.

If it is essential to use BCVs at the source site when using the concurrent SRDF configuration, it will be necessary to use clone emulation mode. See the GDSE TimeFinder/Clone Solutions Guide for further details.

12.3.3 RDF groups

- 1) Determine which data servers will share the same SRDF/A RDF group. Usually there will be one RDF group per data server or application, depending on the required granularity of failover. Remember that it is not possible to operate on a subset of an SRDF/A RDF group e.g. all devices in the RDF group must be failed over together.
- 2) Having determined the total number of RDF groups needed, allocate an RDF group number to each data server or application for the SRDF/A leg.
- 3) Determine which RDF groups will be allocated to each RF pair. See Section 5.3.3 for Best Practice for the allocation of RDF groups to RF pairs..
- 4) Create the SRDF/A RDF groups. See Section 5.3.4.

5) Ensure the mapping of SRDF/A PiT BCVs to their R2 STDs is defined and documented (see Section 5.4) for each data server / application (i.e for each RDF group).

12.3.4 Device groups definitions

There are a number of problems with device group definitions in a concurrent SRDF configuration. These are discussed in detail in Section 7.3, but the main issues are:

- It is not possible to have remote BCV devices defined in a device group down two different RDF groups
- GNS remote mirroring is not supported with concurrent SRDF
- A device can only be in one device group (prior to SE 6.3). This is an issue when subset failover issue is required on the SRDF/S leg.
- BCV control issues with devices only being allowed in one Device Group (prior to SE 6.3)

Hence, in the two 2-Site SRDF/A “swing” scenario being discussed here, the following solution is recommended to minimize the impact of the above constraints imposed by running concurrent SRDF.

1) BCV volumes at SRDF/A target site should not be added to the source site Device Groups (DGs) until after the SRDF/S leg is removed. When running in concurrent SRDF mode, the PiT copies will be controlled from the SRDF/A Target site.

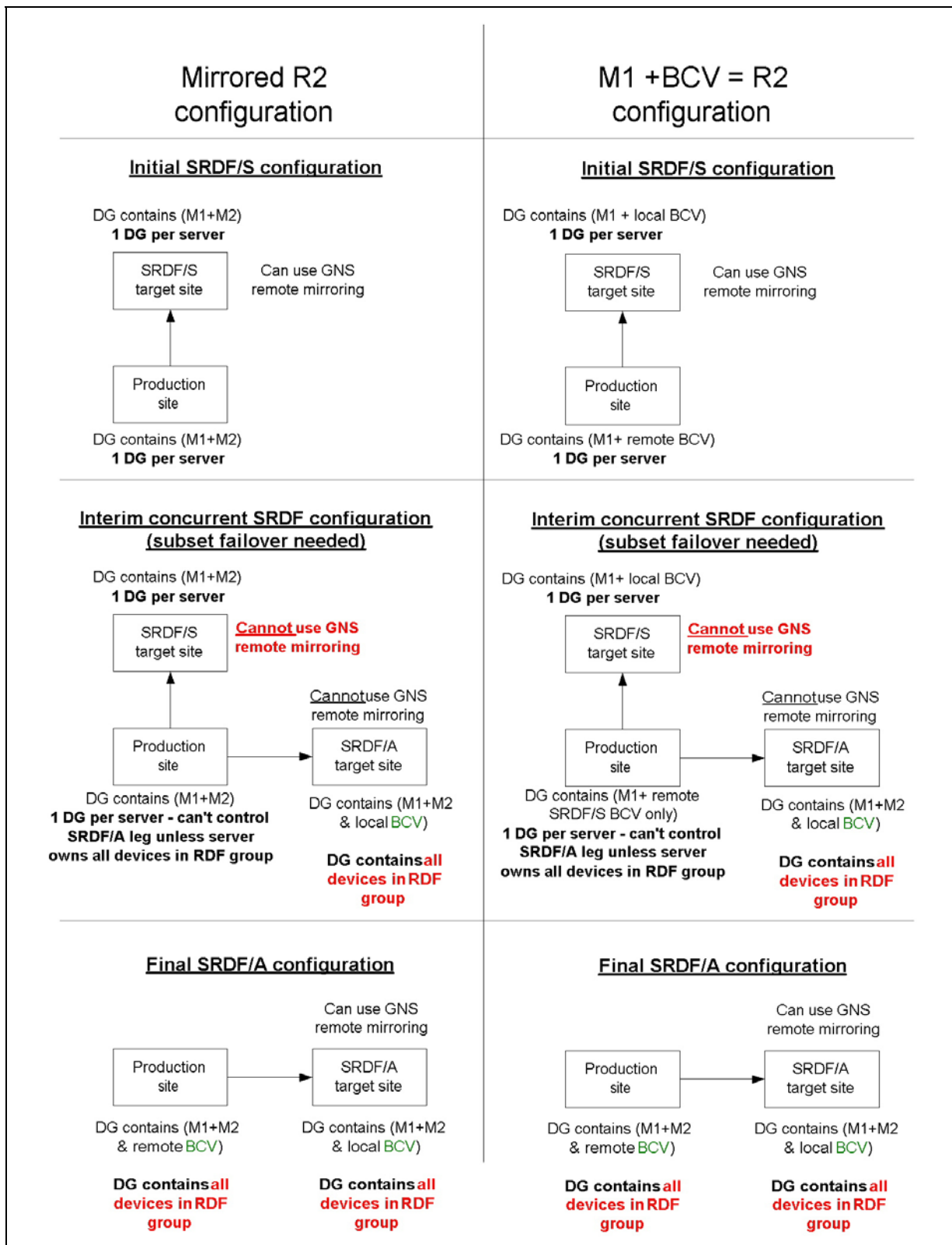
2) When running in concurrent SRDF mode, the source site device groups should remain configured with 1 DG per server, rather than 1 DG per RDF group. Hence, SRDF/A should be controlled from the SRDF/A target site when running in Concurrent mode. This allows server-level failover down the SRDF/S from the source site.

3) If GNS remote mirroring is used, changes to device groups will not be propagated to the remote sites when running in concurrent SRDF mode. During the period of the migrations, it is recommended that either:

- The SYMAPI_GNS_MIRRORED_GROUP_CONTROL option in the SYMAPI options file can be set to enable. With this option enabled, the mirror at Site C can be modified (breaking the gns mirror relationship). See the EMC document “**Using SYMCLI and GNS to Propagate Group Definitions to Multiple Hosts**” for further details of this approach.
- Alternatively, GNS at Site C be temporarily disabled for the SRDF/A target Symmetrix using the `storgnsd:gns_symavoid=<Symm-serial-number>` option in the `daemon_options` file. The Site C device groups for this Symmetrix will need to be setup in the local symapi database until after the migration has completed.

4) Do not implement multi-session consistency until after the SRDF/S leg has been decommissioned.

These recommendations are summarized in the following diagram. The left-hand column is for the scenario where the Sync target site R2 is an M1+M2. The right-hand side is for scenarios where the Sync target site R2 is an M1 + BCV.



12.4 Application COB migration

The actual migration of the COB function is achieved using concurrent SRDF to provide a copy of the data at the remote SRDF/A site.

12.4.1 Setting up Concurrent SRDF

During the application migration planning phase, each data server/application will have been allocated an RDF group for the SRDF/A leg. Similarly, each RDF group will have been allocated to a redundant pair of RFs on both the local and remote SRDF/A DMX.

For each application/server with its own RDF group, the following steps are needed to set up concurrent SRDF. In this example, the Server FLEX01 is using RDF group 4 for SRDF/A on RFs 3A and 14B on both the local Symmetrix (0822) and remote SRDF/A target Symmetrix (1497). The SRDF/S RDF group is 1. If there are multiple servers per RDF group, the steps will differ. Such changes are indicated below.

1) Create Asynchronous Leg RDF group

SRDF/A RDF groups should have their link limbo timer set to 60 seconds. The RDF group must be spread over two RAs for redundancy.

```
# symrdf addgrp -label FLEX01 -rdfig 4 -sid 0822 -dir 3A -link_limbo 60
-remote_rdfg 4 -remote_sid 1497 -remote_dir 3A -rem_link_limbo 60
```

```
# symrdf modifygrp -label FLEX01 -rdfig 4 -sid 0822 -dir 14B -
remote_dir 14B
```

2) Populate RDF group with LUNs and establish SRDF

An R1 $\leftrightarrow$ R2 device pairing file will be need to be prepared. See Section 5.3.5 for further details. In this example, the pairing file is pairsfile_FLEX01.

Do not use -g when populating the second SRDF leg as a device group should already exist for the SRDF/S leg.

```
# symrdf -file pairsfile_FLEX01 -sid 0822 -rdfig 4 createpair -type
rdf1 -establish -nop
```

Note that this command can take a very long time to execute (typically 1 second per RDF volume pair has been seen previously). Hence, sufficient time must be allowed for this when planning this activity.

3) Check concurrent SRDF set up

Use the “-rdfig all” option to ensure that both RDF legs are shown.

```
# symrdf -g FLEX01 query -rdfig all
```

```
Device Group (DG) Name      : FLEX01
DG's Type                  : RDF1
DG's Symmetrix ID          : 000187700822
```

```

Remote Symmetrix ID      : 000187721497
RDF (RA) Group Number   : 4
Remote Symmetrix ID      : 000187721135
RDF (RA) Group Number   : 1

```

Source (R1) View					Target (R2) View					MODES	
-----					-----					-----	
Standard	ST				LI	ST					
Logical	A				N	A					
Device	T	R1 Inv	R2 Inv		K	T	R1 Inv	R2 Inv		RDF Pair	
Dev	E	Tracks	Tracks		S	Dev	E	Tracks	Tracks	MDA	STATE
-----					-----					-----	
DEV001	0001	RW	0	0	RW	0001	WD	0	0	S..	SyncInProg
		RW	0	0	RW	0001	WD	0	0	S..	Synchronized
DEV002	0005	RW	0	0	RW	0005	WD	0	0	S..	SyncInProg
		RW	0	0	RW	0005	WD	0	0	S..	Synchronized
DEV003	0009	RW	0	0	RW	0009	WD	0	0	S..	SyncInProg
		RW	0	0	RW	0009	WD	0	0	S..	Synchronized
DEV004	000D	RW	0	0	RW	000D	WD	0	0	S..	SyncInProg
		RW	0	0	RW	000D	WD	0	0	S..	Synchronized
Total	-----				-----						
Track(s)		0	0			0	0				
MB(s)		0.0	0.0			0.0	0.0				

Legend for MODES:

```

M(mode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
D(omino)             : X = Enabled, . = Disabled
A(daptive Copy)      : D = Disk Mode, W = WP Mode, . = ACp off

```

This shows that each source device has two RDF Pair states. In this case, one is synchronized and the other is synchronizing. Unfortunately, the output does not clearly show which RDG group each entry is associated with.

Each RDF group can be separately interrogated. When in a concurrent SRDF configuration it is necessary to specify which RDF group the command applies to. In this example, for the Async leg the command is:

```
# symrdf -f FLEX01 query -rdfg 4
```

For the Sync leg:

```
# symrdf -g FLEX01 query -rdfg 1
```

If there is more than one server in the SRDF/A RDF group, then the existing "single server" device group set up for SRDF/S replication will not include all the devices in the SRDF/A RDF group. In that case, it will be necessary to use the device file used to setup SRDF to interrogate status instead.

For the Async leg:

```
# symrdf -file pairsfile_FLEX01 query -sid 0822 -rdfg 4
```

4) Enable SRDF/A replication on the Async leg

The enable command enables consistency between devices within the device group.

```
# symrdf -g FLEX01 -rdfg 4 set mode async -noprompt
```

```
# symrdf -g FLEX01 -rdfg 4 enable
```

or if using a device file for SRDF/A leg

```
# symrdf -file pairsfile_FLEX01 -sid 0822 -rdfg 4 set mode async -nop
# symrdf -file pairsfile_FLEX01 -sid 0822 -rdfg 4 enable
```

5) Define device group at SRDF/A target site

5a) Define device group at SRDF/A target site for SRDF/A Control

As mentioned earlier, GNS remote mirroring of device groups that contain Concurrent SRDF device is not supported by EMC (they do not get replicated). Also, as already mentioned, the device group needed for SRDF/A control may not match that needed to control the SRDF/S leg as already defined at the source site.

Hence, if using GNS, it will be necessary to first disable GNS mirroring using one of the methods suggested in section 12.3.4, and then manually define the device group at the SRDF/A target site (Site C). The author is aware that the pre-GNS procedures for maintaining remote device groups differ enormously from site to site (manual entry, export/edit/import of device group definitions, scripts that process a device list, etc). Sites should continue to use these processes for concurrent SRDF device group management.

5b) Add BCVs into device group at remote SRDF/A site. Establish BCVs.

BCVs can be added to groups either using existing local tools used at your site for BCV management. Alternatively, this can be achieved manually using the syntax in the example below.

On the management server at the SRDF/A target site, the BCVs in the target DMX should be associated as local BCVs.

Example

SRDF/A RDF group 4 contains 4 standard volumes. The device group associated with this RDF group is FLEX01. The PIT BCVs to be associated with the R2 volumes are 02AD, 02B1, 02B5, 02B9.

At the SRDF/A target site:

```
# symbcv -g FLEX01 -sid 1497 associate dev 02AD BCV02AD
# symbcv -g FLEX01 -sid 1497 associate dev 02B1 BCV02B1
# symbcv -g FLEX01 -sid 1497 associate dev 02B5 BCV02B5
# symbcv -g FLEX01 -sid 1497 associate dev 02B9 BCV02B9

# symmir -g FLEX01 establish -full -opt -nop
```

12.4.2 Changes to operational procedures when temporarily running in concurrent SRDF mode

It is recommended that for a given application/data server, concurrent SRDF not be used for an extended period due to increased management complexity. However, it is acknowledged that

certain management tasks may need to take place when running in this mode. The main tasks are listed below, with reference to relevant documentation within this Solution Guide.

It is strongly recommended that new LUNs not be added or removed whilst running in concurrent mode. Such allocations should be made prior to the migration or after the migration has completed.

Note that when running in concurrent SRDF mode during the migration, failover should be down the synchronous leg only.

12.4.2.1 Synchronous Leg Failover

It may be necessary to execute an SRDF failover of the synchronous site whilst in concurrent mode. Before the synchronous leg can be failed over, it is necessary to first suspend the asynchronous leg from the SRDF/A target site.

At SRDF/A target site:

```
# symrdf -g <device group> disable -rdfg <Async leg RDF group number>
# symrdf -g <device group> suspend -rdfg <Async leg RDF group number>
```

For example, using the FLEX01 example shown above

```
# symrdf -g FLEX01 disable -rdfg 4 -nop
# symrdf -g FLEX01 suspend -rdfg 4 -nop
```

The asynchronous leg can then be failed over as per your usual processes, either from the source of the SRDF/S target site, **with one exception**:

If you are invoking failover from the source site, you must add the qualifier

-rdfg <sync leg RDF group number>

to your usual failover command.

For example, using the FLEX01 scenario again:

At either the source or the SRDF/S target site:

```
# symrdf -g FLEX01 failover -rdfg 1 -nop
```

It is recommended that the asynchronous leg replication not be restarted (for any devices left in the SRDF/A RDF group which were not failed over) until after failback has occurred.

12.4.2.2 Synchronous Leg Failback

The failback process reverses the previous steps. Again, if the failback command is to be used at the source site, the RDF group must be specified. The FLEX01 example is used again here.

At source or SRDF/S target site:

```
# symrdf -g FLEX01 update -rdfg 1 -nop [Optional step]
```

```
# symrdf -g FLEX01 failback -rdfg 1 -nop
```

At SRDF/A target site:

```
# symrdf -g FLEX01 resume -rdfg 4 -nop
# symrdf -g FLEX01 enable -rdfg 4 -nop
```

12.5 Post-migration steps

First there must be a test of data availability by the application owner. As a minimum, the COB application server at the DR site must be able to see the disks, but ideally application owners should confirm that the data is good.

Once the application owner is happy to switch to using the SRDF/A site as their new COB/DR site, the concurrent SRDF replication must be changed to 2-site SRDF/A.

The main steps for each application are:

- 1) Cease SRDF/S replication
 - Suspend SRDF/S replication
 - Remove devices from SRDF/S device group
- 2) Modify device group(s) at source site (if necessary) to map one-to-one with the SRDF/A RDF group for server/application
- 3) Implement Multi-Session Consistency (if required)
- 4) Update local procedures to accommodate SRDF/A operational requirements
- 5) Physical clean up

12.5.1 Cease SRDF/S replication

Note that once SRDF/S replication is suspended, there may be a jump in I/O performance, as the SRDF/S “throttle” is removed. The SRDF/A replication does not need to be stopped during this process.

1) Suspend SRDF/S replication

At source or SRDF/S target site:

```
# symrdf -g <device group> suspend -rdfg <Sync leg RDF group number>
```

e.g.

```
# symrdf -g FLEX01 suspend -rdfg 1
```

2) Delete devices in FLEX01 from SRDF/S RDF group

It would be prudent to wait a week or so after suspension of the SRDF/S replication before deleting the devices from the SRDF/S RDF group.

This is a destructive command which destroys SRDF relationships. Make sure that you have the SRDF/S group in the command and not the SRDF/A RDF group!

```
# symrdf -g <device group> -rdfg <Sync leg RDF group number>
deletepair
```

e.g.

```
# symrdf -g FLEX01 -rdfg 1 deletepair
```

12.5.2 Modify source site device groups

A single Device Group should be configured at the source site that contains all devices in the relevant SRDF/A RDF group, if this is not already the case. There are many different ways to achieve this result, depending on local device group management practices. You should use your existing local processes for managing device groups.

Remote BCVs (at Site C) also need to be configured at the source site to allow full SRDF/A control to take place from the source site. Again, there are many different ways to achieve this result, depending on local device group management practices. One method is to use a local version of the script add_PIT_BCV (documented in Section 5.4.4), which uses a BCV pairing file to add the remote BCVs to the device group and establish the remote BCVs with the R2 standards. Any remote BCV definitions that relate to BCVs at site B should be removed from the device group first.

If you have disabled GNS remote mirroring prior to the migration process, this should be re-enabled. If mirrored group control was enabled, any device groups modified at Site C should first be deleted at Site C before mirrored group control is disabled. In this way, GNS will subsequently recreated the mirrored group at Site C from Site A and maintain group mirroring going forward.

12.5.3 Implement Multi-Session consistency (if required)

Details on how to implement multi-session consistency between RDF groups are given in Section 6.

12.5.4 Update local procedures

Local Operations documentation will need to be updated. As a minimum, the following changes in process/procedure will need to be addressed:

Process or procedure	Reference
Process for adding a new application requiring SRDF/A replication.	Section 8.1
Process for removing an application requiring SRDF/A replication.	
Process for provisioning additional storage to an existing SRDF/A application.	
Process for removing storage from an existing SRDF/A application.	
Process for stopping and starting SRDF/A replication	Section 8.2
Process for restarting SRDF/A after extended network outage	Section 8.3
Process for application failover/failback	Section 8.4
Process for SRDF personality swapping	Section 8.7
Multi-session consistency operational processes	Section 8.8

Changing replication mode	Section 8.9
Interactive monitoring of SRDF/A configuration and replication	Section 8.10
Using the SRDF/A PIT BCVs for COB testing	Section 9.2
Initial Master Program Load (IMPL) procedures	Section 9.4

12.5.5 Physical cleanup

Once all SRDF/S relationships have been removed, the former synchronous target Symmetrix can be decommissioned. Any RDF groups to this Symmetrix should be removed from the source unit, by EMC if static, or using symrdf deletpair if dynamic.

Details of physical cleanup (cabling removal etc) will vary from site to site and cannot be detailed here.

13 Appendix D – Selection of Point-in-Time technology for SRDF/A resynchronization protection

13.1 Background information

13.1.1 TimeFinder Modes

The following table shows different characteristics of each of the TimeFinder modes. Key points are highlighted in **bold**.

	“Classic” TimeFinder/Mirror	TimeFinder/Clone	TimeFinder/Mirror Emulation mode *
PiT copy is a true mirror, providing protection when established with R2?	Yes	No! (R2 must be mirrored)	No! (R2 must be mirrored)
Compatible with SRDF/Star?	Yes, but with restrictions	Yes	Yes
Compatible with EMC’s OS Automated SRDF restart?	Yes	No	Yes
Requires BCV devices?	Yes	No	Yes
Can use RAID/5?	No!	Yes	Yes
R2 Performance impacted if larger/slower disks are used for SRDF/A PiT copy?	Yes!	Some risk	Some risk
License required (Citigroup VPA contains <u>both</u> licenses)	TimeFinder/Mirror	TimeFinder/Clone	TimeFinder/Clone

* In this mode, the symmir command is used, but is transparently translated to clone operations. Note that if BCVs are on RAID-5 devices, TF/M emulation mode is automatically used.

The key points that this table identifies are:

- Where an R2 is comprised of an unmirrored R2 and a BCV, the BCVs cannot be RAID-5, as RAID-5 BCVs are not true mirrors. **Note that for SRDF/A, R2=M1+BCV is not supported; this configuration should only be used with SRDF/S.**
- For SRDF/A, TimeFinder/Mirror BCVs should not be used on slower disks than the R2s to which they are established.
- “Classic” TimeFinder/Mirror has limitations when used in SRDF/Star configurations due to lack of mirror positions.

13.1.2 Use of RAID-5 to protect RAID1+0 standards

The original Best Practice for SRDF/A PiT resync copies was only to use TimeFinder/Mirror with RAID-0 (stripped) BCVs, with BCVs on the same type (size and rotational speed) of disk as the R2 devices. The disadvantage of that configuration is that the BCVs are unprotected.

Use of RAID-5 (protected) BCVs has a number of advantages and disadvantages:

Pros

- Protected BCVs
- No mirror positions consumed when “established” with R2 as clone technology used under the covers. This is important as a mirrored R2 with a “classic” BCV established impacts use features such as dynamic hot spares and symptimizer.

Cons

- RAID5 (7+1) cannot be used to protect 34GB 4-way striped mirrors as the number of splits per spindle will be exceeded.
- RAID5 (3+1) can only be used to protect 34GB 4-way striped mirrors on 73GB or 146GB disks as the number of splits per spindle will be exceeded on larger disks.
- If RAID (3+1) is used in a frame, then RAID(7+1) cannot be deployed on the same array.
- Higher cost than unprotected BCVs (33% more for RAID (3+1) on like-sized disks).

Hence, it is not practical to protect our existing RAID1+0 Stripped Meta with RAID-5 BCVs.

13.1.3 Symptimizer / hot spare issue

Mirrored R2s have no spare mirror positions when a BCV is established. This would have impact prevent Optimizer being used with these volumes, although use of Optimizer with SRDF configurations is not certified within Citigroup at the time of writing. There are four obvious solutions to this issue:

1. Use “Classic” TimeFinder/Mirror, but only establish BCVs for PiT copy protection when an SRDF/A resynchronization is required. A resynchronization should not be a common event, but the need to update the BCV at this time (rather than have it up-to-date at all times), will impact end-to-end time taken to resynchronize SRDF/A.
2. Use TimeFinder/Mirror Clone emulation. No mirror positions consumed.
3. Use TimeFinder/Clone. No mirror positions consumed.
4. Do not use Optimizer, or split off the BCVs when disk relocation takes place.

13.1.3.1 Hot spares background

Beginning with later 5671 code, permanent member sparing was introduced. This eliminated the need for a free mirror position in certain situations, but in others, a dynamic spare is still used in conjunction with a permanent spare. **Note, however, that Citigroup currently recommends the use of Dynamic sparing (see DMX Solutions Guide), which requires a free mirror position.**

For example, if after removing the failing disk from the array there will still be at least one valid mirror (local or remote), the permanent sparing will not need the dynamic spare and will run from beginning to end. It will replace the failing drive and copy the valid data from an existing mirror. A free mirror position is not required. If, however, removing the disk would result in a situation where there is no valid mirror, the data must be copied from the failing disk to a dynamic spare. In that situation, there must be a free mirror position for the dynamic spare. After the data is copied from the failing disk to the dynamic spare, the permanent sparing script will be run and the dynamic spare will be permanently replaced.

If hot sparing (with dynamic spares) is required at the R2 site, TimeFinder/Mirror Clone emulation should be used to ensure that a free mirror position is available. However, use of hot spares is not mandatory in an SRDF environment due to the very low chance of data loss before EMC can repair any failed disk, as further multiple failures would need to occur.

13.2 Best Practice Recommendation

In order:

- to minimize diversity of SRDF/A PiT copy solutions between the different four supported SRDF deployment scenarios (2-site SRDF/S, 2-site SRDF/A, 3-site concurrent SRDF and 3-site SRDF/Star)
- to be compatible with current EMC documentation on SRDF PiT copy protection (SRDF/A and SRDF/Star)
- to offer a simple, generic solution without lots of configuration-specific caveats. e.g.
 - o avoid usage of RAID(3+1) conflicting with a wish to use RAID(7+1)
 - o avoid a solution that is sensitive to the size of the underlying physical disk
 - o adopt a solution that is compatible across the SRDF range of products as we use them
- to minimize cost

the recommended Best Practice is as follows:

13.2.1 SRDF/S – RAID1+0 R2s

In SRDF/S deployments where R2 = M1+BCV is required for COB testing, only use “Classic” TimeFinder/Mirror (with RAID-0 BCV devices on disks at least as fast as the R2 standard devices). Can be used with DMX2 or DMX3.

13.2.2 2-site SRDF/A & 3-Site concurrent SRDF – RAID1+0 R2s

There are two options for RAID1+0 4-way striped meta R2s. For both, RAID-0 BCVs should be of the same disk type (capacity and rotational speed) as the R2s which they will be protecting.

13.2.2.1 Existing Standard (Preferred option)

Use “Classic” TimeFinder/Mirror (with RAID-0 BCVs) on disks at least as fast as the R2 standard devices remains the preferred standard at the present time. This approach:

- maintains similar management processes with existing SRDF/S deployments where $R2 = M1 + BCV$
- is compatible with EMC’s Best Practice, documentation and forthcoming OS Automated SRDF restart scripts
- is compatible with existing Citigroup SRDF/A PiT Best Practice.

13.2.2.2 New additional Standard (Alternative option)

Use TimeFinder/Mirror Clone Emulation (with RAID-0 BCVs). This approach:

- Has all of the above advantages
- Does not consume a mirror position, enabling usage of features such as dynamic sparing at the R2 end at all times

The only caveats with the Clone emulation mode are:

- Symmetrix must have a TimeFinder/Clone license
- Newer technology. Not as well proven as “Classic” TimeFinder/Mirror.

As a consequence of these caveats, usage of TimeFinder/Mirror Clone Emulation is not being mandated over “Classic” TimeFinder/Mirror, which remains the preferred option at present.

13.2.3 3-site SRDF/Star – RAID1+0 R2s

Use of TimeFinder/Mirror Clone emulation mode is preferred in this configuration, with either RAID-0 BCVs as switch operations require a free mirror position, and will fail if a BCV is established at the R2 end (see SRDF/Star Solutions Guide for further details).

“Classic” TimeFinder/Mirror can be used for SRDF resync. PiT copies, but any BCV copies **MUST** be split off before a switch can take place. As this rule applies to both SRDF/A resync BCVs AND user BCVs, use of “Classic” Timefinder/Mirror by users should not be permitted in an SRDF/Star environment.

13.2.4 All SRDF configurations with RAID5 R2s

Use RAID-5 BCVs with TimeFinder/Mirror, with identical configuration as the standards to which they are protecting. The DMX3 standard for RAID5 uses 34.53GB hypervolumes. Clone

Emulation mode for TimeFinder/Mirror will automatically be invoked when RAID5 BCVs are used.

13.3 “FAQs”

13.3.1 Why is the use of RAID-5 BCVs not being recommended for RAID1+0 R2s?

Although RAID-5 BCVs superficially look like an attractive option, there are some significant limitations that impact its adoption as a standard for protecting RAID1+0 R2s:

- Can only be used with 146GB or small disks (to avoid splits-per-spindle issue)
- Can only be used with RAID(3+1), again to avoid splits-per-spindle issue. This would prevent use of RAID(7+1) in the same frame.
- Not supported in Citigroup on DMX2

These three issues make RAID5 a poor choice as a standard for this role.

13.3.2 Why is the use of symclone not being recommended?

At the present time is not being recommended as:

- All EMC documentation, scripts and processes still refer to symmir
- Not compatible with existing Citigroup scripts, procedures and documentation. Significant effort to modify all these.
- Symclone must be used with care with SRDF/A R2s. These issues do not occur with symmir (classic or emulation mode). Use of TimeFinder/Mirror Clone emulation enforces the correct steps to safely “clone” from an R2.

13.3.3 Why is the use of larger disks not being recommended for PiT copies?

Larger (slower) disks should not be used when “Classic” TimeFinder/Mirror is used, as anything that causes the R2 to be slower than the R1 when running SRDF/A is potentially a problem. This has been confirmed by EMC (Andrew Lubeck – Principle Corporate Systems Engineer). Hence, use of large disks precludes the option to revert to “Classic” TimeFinder/Mirror.

With clones, slower physical disks are unlikely to present a problem, but there is some small amount of risk. The danger is that the slower drives could accumulate write pendings, which would consume cache on the R2 and could cause SRDF/A to drop.

As insufficient testing has been undertaken at the present time of using larger disks with TimeFinder/Mirror Clone Emulation, and the impact on SRDF/A replication is not certain, this is not currently recommended as the default solution. When further testing has been completed, the Best Practice will be revised as appropriate.

13.3.4 What are the downsides of sticking with “Classic” TimeFinder/Mirror rather than using clone emulation mode?

The only disadvantage is the consumption of an additional mirror position when the BCV is established. The impact on this is:

- Symptomiser would not be able to run against these volumes.
- Use of hot spares may be impacted in some circumstances (see earlier description).
- In SRDF/Star configurations, it is essential that BCVs established with R2s be split off before attempting a site switch. This is Best Practice in any case.

13.4 Mapping SRDF/A point-in-time copies to R2s

Care must be taken to optimize the R2 / RBCV pairing as far as the device layout is concerned on the back end (DA and Spindles) to minimize contention. If you have R2 / RBCV pairing which are contentious (say the devices share the same spindle or DA), this could create a drag on the DA. This in theory could throttle your SRDF/A performance and elongate the cycle as the DA is what is responsible for the SRDFA restore operation and the destage of apply cycle data to the R2.

Clone copy (TimeFinder/Clone or TimeFinder/Mirror Clone Emulation) is a DA operation and hence contentious clone pairs could also throttle a DA, and anything which could overload a DA could affect the SRDF/A cycle. A "few" contentious pairs should not bring a DA to its knees; however, it is hard to predict how many contentious pairs is the threshold. So much of this depends on device type (RAID5 involves even more disks and DA resources) and active workload.

As a rule of thumb - contentious source/target device pairs is a recipe for trouble. It doesn't matter what type of Timefinder technology you use.