



Global Distributed Storage Engineering
Citi Architecture & Technology Engineering

Cascaded SRDF and other 5773 SRDF features

GDSE-08004-TR

Evaluation and Test Results
Version 1.0

VERSION CONTROL

Version and Date of Issue	Author	Comment
V1.0 9th April 2008	Chris Clayton	First draft

CONTENTS

Legal Notice	6
Contact	6
Latest Versions	6
1 EXECUTIVE SUMMARY	7
2 NEW SRDF FEATURES IN 5773	8
2.1 Cascaded SRDF & Cascaded Star	8
2.1.1 Description	8
2.1.2 Key findings	10
2.1.3 Recommendations	14
2.2 Moving dynamic RDF pair devices	14
2.2.1 Description	14
2.2.2 Key findings	14
2.2.3 Recommendations	15
2.3 SRDF/A R2 timestamp representation	15
2.3.1 Description	15
2.3.2 Key findings	15
2.3.3 Recommendation	16
2.4 Minimum cycle time for SRDF/A	16
2.4.1 Description	16
2.4.2 Key findings	16
2.4.3 Recommendation	16
3 APPENDIX A - “NATIVE” CASCADED SRDF FAILOVER SCENARIOS	17
3.1 Planned switches	17
3.1.1 A --> B --> C to C --> B --> A and 1b) reversal	17
3.1.2 Fast change from A --> B --> C to production at C	19
3.2 Unplanned switches (Site A Symm down)	20
3.2.1 Change from A --> B --> C to C --> B	20
3.2.2 Change from C --> B to A --> B --> C	20
3.2.3 Fast change from A --> B --> C to production at C	21
4 APPENDIX B - OTHER CASCADED SRDF OPERATIONAL TASKS	22
4.1 Adding to and removing disks from a Cascaded SRDF configuration	22
4.1.1 Using movepair	22
4.1.2 Not using movepair	22

4.2	Other Cascaded SRDF issues	23
4.2.1	Symrecover does not support for hop2 BCVs	23
4.2.2	Cascaded SRDF remote mirroring	23
5	APPENDIX C - CASCADED STAR FAILOVER SCENARIOS	25
5.1	Planned failover to Site C (with ongoing C → B async replication)	26
5.2	Planned failover to Site C (with ongoing C → A async replication)	26
5.2.1	Note on criticality of command order	27
5.2.2	Comparing runtime of cascaded and concurrent planned switches	28
5.2.3	Fast change from A --> B --> C to production at C	29
5.3	Planned Reconfiguration from Cascaded to Concurrent (and back)	29
5.3.1	When halt of workload site possible	29
5.3.2	Without halting workload site	30
5.4	Transient Fault recovery	30
5.4.1	Loss of synchronous link in cascaded configuration	30
5.4.2	Loss of asynchronous link in cascaded configuration	31
5.5	Unplanned switch to Site C (Site A Symm goes down)	31
5.5.1	Change from A → B → C to C → B	31
5.5.2	Comparing runtime of cascaded and concurrent unplanned switches	32
5.5.3	Comparing runtime of SE6.3 and SE6.5 concurrent unplanned switches	33
5.5.4	Fast change from A --> B --> C to production at C	34
5.6	Other Cascaded Star issues	34
5.6.1	Microcode requirements	34
5.6.2	Setup	34
5.6.3	Star protection	34
5.6.4	Disabling Star	35
5.6.5	Connect fails with “Disabling RDF Consistency protection” error	35
6	APPENDIX D - OTHER NOTABLE ISSUES ENCOUNTERED DURING BETA	36
6.1	Moving Dynamic RDF device pairs	36
6.1.1	Device groups not updated when pairs move	36
6.2	Minimum SRDF/A cycle time	36
6.3	GNS remote mirroring issue	37
7	APPENDIX E – ENHANCEMENT REQUESTS SUBMITTED TO EMC	38
7.1	Further detail on GNS remote mirroring support request (274630)	38
8	APPENDIX B – CONSOLIDATED TEST RESULTS	40

8.1	5773 New features for 2-Site SRDF/A	40
8.2	Cascaded SRDF – Base testing	43
8.3	Cascaded SRDF with Star Technologies – Base testing	45
8.4	Additional Citi Defined Testing	47

Legal Notice

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

CITI, CITIBANK, CITICORP and CITIGROUP 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.citigroup.net/gdse/products/>

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

1 Executive Summary

Enginuity 5773 combined with Solutions Enabler 6.5 offers several new SRDF features:

- 1) **Cascaded SRDF:** Cascaded SRDF is a new 3-site disaster recovery configuration where data from a primary site is synchronously replicated to a secondary site, and then asynchronously replicated to a tertiary site.

Benefits of Cascaded SRDF include it being the simplest (least sophisticated) 3-Site solution that offers the potential of zero data loss at Site C should Site A fail. It is also the most scalable SRDF-based 3-site solution. However, when used outside of SRDF/Star control it has a number of significant recoverability restrictions.

Cascaded SRDF may also be of value for temporary usage to aid with migrations.

It is recommended that Cascaded SRDF only initially be certified to be run under SRDF/Star. Use of “native” cascaded SRDF (not under Star control) should only be considered for certification if demand for 3-site solutions in Citi outstrips the scalability of SRDF/Star.

- 2) **Cascaded Star:** An enhancement to Cascaded SRDF that adds:

- Differential resynchronization capabilities of the surviving sites
- Automation of site switch tasks via the symstar command
- Dynamic reconfiguration to a concurrent configuration to deal with various failure scenarios

Hence, Cascaded Star addresses the recoverability limitations of “native” Cascaded SRDF.

It is recommended that certification of cascaded SRDF initially be limited to dealing with certain failure scenarios with Concurrent Star configurations, rather than full certification of Cascaded Star as a normal operating configuration.

- 3) **Moving RDF pair devices.** Provides the ability to move dynamic RDF pair devices between RDF groups without a full resynchronization.

It is strongly recommended that Citi Best Practice be updated to use this capability to reduce the overall time taken to allocation SRDF/A storage to hosts.

- 4) **SRDF/A R2 timestamp.** An actual timestamp, indicating the point-in-time that the SRDF/A R2 device represents can now be obtained both before and after an SRDF/A failure has occurred, and from either the source or target site. This can be provided to application owners after a disaster to help with quantifying data loss in a disaster scenario.

It is strongly recommended that Citi Best Practice be updated to use this new capability.

- 5) **Minimum Cycle time for SRDF/A.** This has now been reduced from 5 seconds to 1 second.

It is not recommended that this new minimum cycle time be adopted.

2 New SRDF features in 5773

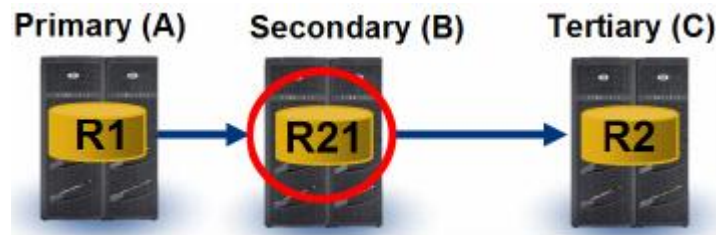
2.1 Cascaded SRDF & Cascaded Star

2.1.1 Description

Cascaded SRDF is a new three site disaster recovery configuration where data from a primary site is synchronously replicated to a secondary site, and then asynchronously replicated to a tertiary site.

Prior to Enginuity 5773, an SRDF device could be a primary device (R1 device) or a secondary device (R2 device); however it could not be in the dual role simultaneously. Cascaded SRDF introduces a new SRDF R21 device. The R21 device will assume dual roles of primary (R1) and secondary (R2) device types simultaneously. Data received by this device as a secondary can automatically be transferred by this device as a primary.

The basic cascaded SRDF configuration consists of a primary site (site A) replicating synchronously to a secondary site (site B) with SRDF/S, and then replicating the same data asynchronously to a tertiary site (site C) with SRDF/A.

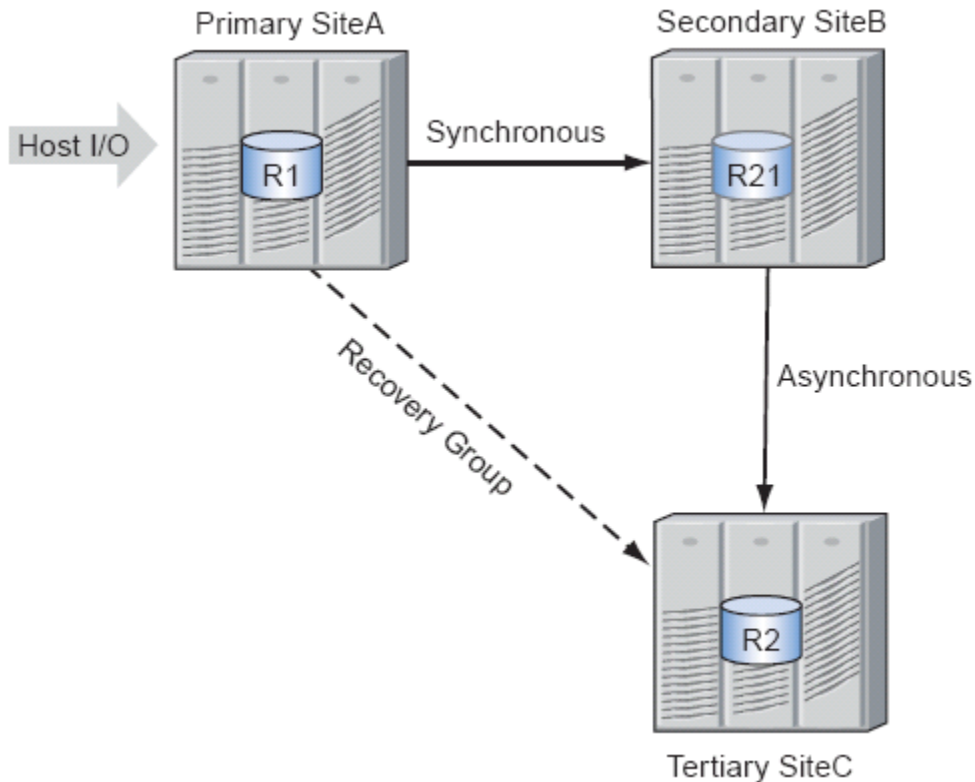


Cascaded SRDF devices are supported only on the Symmetrix DMX-3 and DMX-4 hardware platforms with Enginuity 5773 and above; however, only the Symmetrix that contains the R21 device is required to be Enginuity 5773. The Enginuity level for the Symmetrix containing the R1 and the R2 can be any level that EMC supports being connected to Enginuity 5773.

Cascaded Star provides the following significant benefits over “native” cascaded SRDF (i.e. a cascaded SRDF setup not controlled using Star):

1. Failover automation for Cascaded SRDF. Failover of Cascaded SRDF can be automated using home-grown scripts.
2. The ability to reconfigure a cascaded SRDF configuration into a concurrent SRDF (Star) configuration to allow replication directly from Site A to Site C should replication via Site B fail.
3. The ability to reconfigure a cascaded SRDF configuration into a concurrent SRDF (Star) configuration to allow on-going replication directly from Site C to Site A for a planned switch to Site C (e.g. after a server failure at Site A).
4. Differential resynchronization capabilities between the surviving sites after loss of a site.

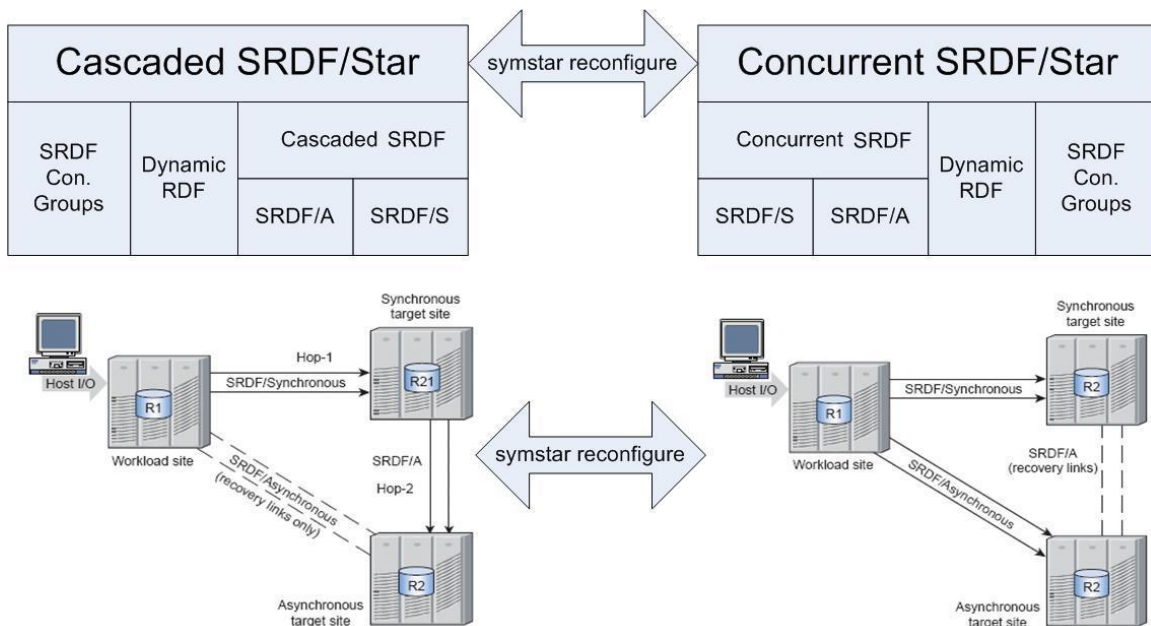
5. With Cascaded SRDF, asynchronous replication cache usage load is moved off of the production system.



Cascaded Star requires 5773 microcode at Sites A and B. “Native” Cascaded SRDF only needs 5773 at Site B.

The following diagram shows:

1. How Cascaded and Concurrent Star both are dependent on other SRDF technologies.
2. How the symstar reconfigure command can be used to move between the cascaded and concurrent configurations. This is vital in a Cascaded configuration as:
 - a. Loss of Site B (Synchronous Target), or links to or from Site B ($A \rightarrow B$ or $B \rightarrow C$) would lead to loss of DR protection. A reconfiguration to a Concurrent configuration to allow replication directly from A to C is necessary.
 - b. In a planned failover to Site C, replication back to Site A would require reconfiguration to a Concurrent configuration to allow direct SRDF/A replication from $C \rightarrow A$, as $C \rightarrow B \rightarrow A$ replication does not result in a consistent image at Site A.



2.1.2 Key findings

2.1.2.1 Comparison of different 3-Site configurations

The following table compares the characteristics of each of four different 3-Site DR solutions now available from EMC, and based on Citi testing and evaluation.

- Items in **green** are positive or non-issue items.
- Items in **orange** are acceptable but not as good as the **green** items
- Items in **orange italics** are acceptable would require further Citi certification work
- Items in **bold orange** are considered weak (but not showstopping) features of the product.
- Items in **large bold red** are considered “showstoppers”.

	Concurrent		Cascaded (new)	
	“Native” Concurrent SRDF	Concurrent SRDF/Star	“Native” Cascaded SRDF	Cascaded SRDF/Star
Minimum microcode	5771	5771	5773 (at Site B)	5773 (at Sites A & B)
Certification status in Citi	Certified, but not	Certified today	Not Certified yet	Not Certified yet

Cascaded SRDF and other 5773 features

	strategic solution			
Requires use of RDF daemon	No	Yes	No	Yes
Scalability	High	<u>Low. Limited by RDF daemon and SDDF session scalability</u>	High	<u>Low. Limited by RDF daemon and SDDF session scalability</u>
Failover automation from EMC?	<u>None</u>	Yes, using symstar	<u>None</u>	Yes, using symstar
Speed of failover to Site C	<u>Slow</u>	Moderate with 5771/5772 but fast with 5773/SE6.5	Fast (5773/SE6.5)	Fast (5773/SE6.5)
Ability to provide Best Case (non-rolling disaster) zero-data loss at Site C in the event of the loss of Site A with minimal intervention	<u>No</u>	Requires switch operation to Site C, keeping Site B data	Data automatically replicates to Site C. However, a symrdf failover would be needed to make it available to hosts.	Requires switch operation to Site C, keeping Site B data. Data does not automatically replicate from Site B to Site C as SRDF/A switching daemons run at Site A and would not be available. Data automatically replicates to Site C for application server failures at Site A.
Ability to deal with loss of Site B without compromising DR protection	Yes. Direct replication to Site C	Yes. Direct replication to Site C	<u>No</u>	Yes, via symstar reconfigure to a concurrent configuration.
Ease of planned switch to C with C → A replication (e.g. COB test)	<u>Requires Citi scripts</u>	OK	<u>Cannot provide a consistent image at Site A</u>	Yes, via symstar reconfigure to a concurrent configuration and symstar switch
Ease of unplanned switch to C with C → A (e.g. loss of server)	<u>Requires Citi scripts</u>	OK	<u>Cannot provide a consistent image at Site A</u>	Yes, via symstar reconfigure to a concurrent configuration and symstar switch
Ease of unplanned switch to C with C → B	OK	OK	Requires Citi scripts	Yes via symstar switch

(e.g. Loss of Site A)				
Ease of unplanned reconfiguration to A → C (e.g. Loss of Site B)	Not needed	Not needed	Cannot be done	Yes, via symstar reconfigure to a concurrent configuration
Platform for “dataless” bunker site?	No	No	Future development	Future development

The key findings for the new configurations only (“Native” Cascaded SRDF and Cascaded Star) are discussed below.

2.1.2.2 Key findings - “Native” Cascaded SRDF (no Star control)

- **Positive: Simplicity.** “Native” Cascaded SRDF is the simplest, entry-level 3-Site solution, that offers the potential of zero data loss should Site A fail. It is a replacement for the older “multi-hop” solution.
- **Positive: Scalability.** “Native” Cascaded SRDF is more scalable than SRDF/Star (at time or writing). “Native” Cascaded SRDF scalability is limited by 2-Site SRDF/A limits, although Site B needs 2 RDF groups per cascaded configuration limiting the total number for cascaded configurations nominally to 64 replication groups per set of 3 arrays. Cascaded Star is limited to 32 replication groups per set of 3 arrays.
- **Major Negative: Flexibility.** With “native” Cascaded SRDF, a failure of any of loss of Site B, the A – B link or the B – C link will stop replication to Site C. With “Native” Cascaded SRDF, this would cease replication to our COB/DR site (Site C).

2.1.2.3 Key findings - Cascaded Star

Cascaded Star refers to a cascaded SRDF configuration under Star control.

- **Positive: Flexibility:** As indicated above, “Native” Cascaded SRDF has a number of undesirable restrictions. These can be addressed by running cascaded SRDF under Star control (i.e. in a Cascaded Star configuration).
 - i. The ability to reconfigure a cascaded SRDF configuration into a concurrent SRDF (Star) configuration to allow replication directly from Site A to Site C should replication via Site B fail is only available when a cascaded configuration is under Star control (via symstar command).
 - ii. The ability to reconfigure a cascaded SRDF configuration into a concurrent SRDF (Star) configuration to allow on-going replication directly from Site C to Site A for a planned switch to Site C (e.g. after a server failure at Site A) is only available when a cascaded configuration is under Star control (via symstar command).
 - iii. Differential resynchronization capabilities between of the surviving sites after loss of a site only available when a cascaded configuration is Star protected.

- **Positive: Automation.** Failover automation for Cascaded SRDF is available via symstar when under Star control.
- **Negative: Scalability.** Scalability of a cascaded SRDF configuration is reduced when running under Star control. This is primarily due to the scalability of the RDF daemons, as the number of Star sessions increases, and the number of SDDF sessions that a Symmetrix can support (SDDF is the Symmetrix Differential Data Facility that is used for incremental updates in SRDF).
- **Negative: RDF daemon.** As with all SRDF/Star options, SRDF/A leg cycle switching is controlled by RDF daemons that run on external servers. These daemons must be running for replication to take place and physical servers for these daemons must be provided (an issue if a single pair of RDF daemons cannot deal with the cycle switching load of the SRDF/Star installed base). “Native” cascaded SRDF offers 3-site nominal “zero-data loss” without the use of RDF daemons.

2.1.2.4 Key findings - Generic Cascaded SRDF comments

The following points apply to both “native” cascaded SRDF and Cascaded Star.

- **Positive: Loading.** With Cascaded SRDF, asynchronous replication cache usage load is moved off of the production system.
- **Positive: Migrations.** Cascaded SRDF may be of use to help with migrations. Only the “middle” site would need to be running 5773. These scenarios were not explored as part of the Beta testing, but have been identified for future consideration.
- **Positive: Cost (future).** Cascaded configuration is platform for the development of “diskless bunker” in the future. No official timeframes for development are available from EMC yet.
- **Minor negative: Cost.** Cascaded SRDF needs an additional licence.
- **Cascaded vs Concurrent:** When running in a Cascaded SRDF configuration, there are several scenarios that require one to reconfigure to a Concurrent configuration:
 - I. Planned failover to Site C, with consistent replication back to Site A (presumed common)
 - II. Loss of links between Site A and Site B
 - III. Loss of links between Site B and Site C (assuming it cannot be routed at the network level via Site A)
 - IV. Loss of Site B (Bunker)

Hence, if one adopts Cascaded SRDF, one will need to reconfigure to use Concurrent to work around the above issues and hence will need to run under Star control. There are fewer circumstances when one must change to a Cascaded SRDF configuration when running Concurrent SRDF, but they do exist:

- I. Loss of links between Site A and Site C (assuming it cannot be routed at the network level via Site B)

2.1.3 Recommendations

- “Native” Cascaded SRDF has a number of restrictions relating to dealing with failures and any Citi certification of cascaded SRDF configurations should be for use under Star control.
- There is a slight preference in Open Systems for an SRDF/Star configuration to be Concurrent rather than Cascaded for the following reasons:
 - Adding full support for Cascaded configurations to Citi’s existing certified Star solution that is based on Concurrent configurations will even further complicate the our SRDF/Star solution for Operations, with only modest benefit.
 - Planned switched to Site C, with consistent replication back to Site A (presumed common) require a concurrent SRDF configuration, or a reconfiguration from cascaded to concurrent. Hence, operating in Concurrent modemakes the normal configuration is “C → A ready”.
 - With the concurrent configuration, there are less failure scenarios where one would require reconfiguration
- The process for reconfiguring to cascaded configuration should certified and added to the SRDF/Star Solutions Guide to deal with the failure scenario when A →C replication is not possible.
- If demand within Citi for 3-site solutions begins to outstrip the scalability of Star (limited by SDDF sessions and RDF daemon scalability), then certification of “native” Cascaded SRDF should be considered.

2.2 Moving dynamic RDF pair devices

2.2.1 Description

Previously, moving a dynamic RDF device pair from one RDF group to another required the deletion of the existing pair followed by the creation of the same pair in the new RDF group. While successful in moving the device pair, this process resulted in a full resynchronization of the data being performed.

Enginuity version 5773 allows this move to occur without the deletion of the existing dynamic RDF pair, thus avoiding a full resynchronization.

2.2.2 Key findings

- With SRDF/A, we usually have one RDF group per application. To add volumes to an application requires a full resync of the new volumes at the time they are added to the RDF

group. This process causes the remote copy of the application data to appear not to be consistent throughout the entire period it takes for the new volumes to resynchronize. A point-in-time copy needs to be made prior to the resynchronization, which will age during the resync.

- The new movepair feature will allow us to pre-synchronize the volumes to be added, hugely reducing the time that the application is not being fully protected by SRDF/A.

2.2.3 Recommendations

- We will definitely want to certify the use of movepair for adding disk to SRDF/A groups and update Citi Best Practice documentation.

2.3 SRDF/A R2 timestamp representation

2.3.1 Description

When using SRDF/A, it is important to constantly monitor adherence to the recovery point objective (RPO). Previously, when querying an SRDF/A enabled RDF group, the “time that R2 is behind R1” for the R2 data was only displayed while the SRDF/A session was active. It was also only available when querying from a host connected to the production array.

With Enginuity version 5773, the microcode running on the Symmetrix DMX will maintain the time that the R2 is behind the R1 and return this information to Solutions Enabler SYMCLI commands. This value will then be available to the API whether the SRDF/A session is active or inactive. The information will be stored on both the R1 and R2 Symmetrix DMX and so will also be available when querying the SRDF/A devices from a host connected to the R2 Symmetrix DMX. SYMCLI will also calculate and display an actual timestamp, indicating the time at which the “apply” SRDF/A delta set transitioned from the “capture” delta set to the “transmit” delta set.

```
# symrdf -g RDFA query -rdfa
Device Group (DG) Name : RDFA
...
RDFA Session Status : Inactive
...
Time that R2 is behind R1 : 00:02:49
R2 Image Capture Time : Thu Dec 6 10:25:00 2007
```

2.3.2 Key findings

- Today it is not possible to determine after a failure the point-in-time that the application data represents. With 5773, we will be able to communicate to the business the capture time (to within ~1 second if NTP is used on the Solutions Enabler host) of the data on the R2 device.
- The value of this to any given application owner is uncertain, but it is an improvement on the situation today where we can give only a limited indication of the amount of data that might have been lost.

2.3.3 Recommendation

- We should add the use of this feature to Citi Best Practice documentation.

2.4 Minimum cycle time for SRDF/A

2.4.1 Description

In 5773, this is reduced from 5 seconds to 1 second (3 seconds is using Multi-Session Consistency).

2.4.2 Key findings

- Limited benefit to Citi. This is mostly a marketing benefit to show that SRDF/A can have an RPO of 1-2 seconds as HDS claim with HUR. In practice, there are disadvantages to setting a very short minimum cycle time for SRDF/A.
- You lose the write-folding benefits that result in 20-30% reduction in bandwidth requirements you get when the minimum cycle time is set to 30 sec.
- In testing, we found that the actual cycle time (not minimum time) extended to several seconds (depending on load) as it takes a finite amount of time to replicate the transmit delta to the remote site.

2.4.3 Recommendation

- No intention to recommend very low cycle times as Best Practice

3 Appendix A - “Native” Cascaded SRDF Failover scenarios

The following assumes that Site B is a bunker and that you would never actually failover to that site. In all of the following , Site A is production (under normal circumstances) and Site C is the failover site.

Planned switches to site C (Site A Symm still up)

- Change from A --> B --> C to C --> B --> A (reversal of replication direction)
- Change from C --> B --> A to A --> B --> C (reversal of the above change i.e. back to normal operations)
- Fast change from A --> B --> C to production at C (quickest process to get applications up at Site C when Site A server/cluster fails)

Unplanned switches (Site A Symm goes down)

- Change from A --> B --> C to C --> B (provides ongoing replication)
- Change from C --> B to A --> B --> C (reversal of the above change. This is strictly a planned switch after Site A is available again)
- Fast change from A --> B --> C to production at C (quickest process to get applications up at Site C)

All of the above switch types were tested. The following points were noted.

3.1 Planned switches

3.1.1 A --> B --> C to C --> B --> A and 1b) reversal

Note that a C --> B --> A configuration does not provide a consistent image at Site A, as the B → A leg must in in adaptive copy. Hence, this is actually of little use to Citi.

This can be achieved all from Site A, but this requires one to use device files rather than the device groups for certain steps. For example, the following script, run at Site A was used to swap replication direction, and then swap it back.

```
#!/bin/sh -x
#
# Swap over
#
symrdf -g beta11 query; symrdf -g beta11 query -hop2
symrdf -g beta11 suspend -nop
symrdf -g beta11 suspend -hop2 -nop
```

```

#
# Have to set 919-717 hop to acp_disk ready for swap as hop2 cannot be sync
#
symrdf -g beta11 set mode acp_disk -nop
#
# Have to failover and swap hop1 first to avoid a R1 → R22 ← R1 config)
# This will instead give us R2 ← R11 → R2, which is legal
#
symrdf -g beta11 failover -force -nop
symrdf -g beta11 swap -nop
#
# Cannot use query -hop2 as no longer a cascaded configuration (R2 ← R11 → R2)
#
symrdf -f pairsfile_casc_717_3110_beta12 -sid 717 failover -rdfg 12 -force -nop
symrdf -f pairsfile_casc_717_3110_beta12 -sid 717 swap -rdfg 12 -nop
symrdf -g beta11 query; symrdf -g beta11 query -hop2
#
# Now we have a cascaded setup going opposite direction to original setup
#
symrdf -g beta11 establish -nop
symrdf -g beta11 query; symrdf -g beta11 query -hop2
symrdf -g beta11 establish -hop2 -nop
symrdf -g beta11 query; symrdf -g beta11 query -hop2
#
# Swap back
#
echo " *****"
echo " *** Swapping back...."
echo " *****"
symrdf -g beta11 suspend -nop
symrdf -g beta11 suspend -hop2 -nop
symrdf -g beta11 failover -hop2 -nop
symrdf -f pairsfile_casc_717_3110_beta12 -sid 717 swap -rdfg 12 -nop
symrdf -g beta11 failover -force -nop
symrdf -g beta11 set mode sync -nop
symrdf -g beta11 swap -nop
symrdf -g beta11 establish -nop
symrdf -g beta11 query; symrdf -g beta11 query -hop2
symrdf -g beta11 establish -hop2 -nop
symrdf -g beta11 query; symrdf -g beta11 query -hop2

```

The following issues were noted:

- 1) After first swap is executed, configuration is no longer cascade and hence use of -hop2 commands no longer works. Hence, one is forced to use a device file to complete the process.
- 2) Some operations require use of -force (e.g. failover commands).
- 3) If a "resume -force" is done instead of the "establish used in the above script, the RDF pair states become invalid.

EMC's response was:

The best practices in failing over to site C and back to A are still being examined. It would not be possible to run the entire scenario (failover and failback) from a single host connected to either site A or C and continue to work against the same device group. However, it should be possible to perform the initial failover and swaps from a host connected to site C, and then perform the failback from a host connected to site A. From a host connected to site B, however, it would be possible to run the failover/failback/swap commands against a single

device group as the swap command will alter the type of the device group from R21 to R1 and vice versa.

Hence, further testing found that the device group exclusively (no device files) could be used throughout the swap process if:

- Change from A → B → C to C → B → A is controlled from Site C
- Change from C → B → A to A → B → C is controlled from Site A

The following tested scripts show the steps required for device group only failovers.

Site C script for change from A → B → C to C → B → A

The commands are to run as Site C. Note that the device group has to be set up manually at Site C as GNS remote mirroring is not supported in a cascaded environment.

```
symrdf -g betall suspend -hop2 -nop
symrdf -g betall suspend -nop
symrdf -g betall failover -hop2 -nop
symrdf -g betall failover -force -nop
symrdf -g betall set mode acp_disk -hop2 -nop
symrdf -g betall swap -hop2 -nop
symrdf -g betall swap -nop
symrdf -g betall establish -hop2 -nop
symrdf -g betall establish -nop
symrdf -g betall query -hop2
symrdf -g betall query
```

Site A script for change from C → B → A to A → B → C is controlled from Site A

```
symrdf -g betall suspend -hop2 -nop
symrdf -g betall suspend -nop
symrdf -g betall failover -hop2 -nop
symrdf -g betall failover -force -nop
symrdf -g betall swap -hop2 -nop
symrdf -g betall swap -nop
symrdf -g betall set mode sync -nop
symrdf -g betall establish -hop2 -nop
symrdf -g betall establish -nop
symrdf -g betall query
symrdf -g betall query -hop2
```

3.1.2 Fast change from A --> B --> C to production at C

This is documented as the fastest possible failover process. No ongoing replication is set up.

The commands are to run as Site C. Note that the device group has to be set up manually at Site C as GNS remote mirroring is not supported in a cascaded environment.

```
symrdf -g betall suspend -hop2
symrdf -g betall suspend
symrdf -g betall failover -force -nop
```

3.2 Unplanned switches (Site A Symm down)

3.2.1 Change from A --> B --> C to C --> B

In this scenario, the A – B leg will be partitioned. It will not be possible to execute a swap on the B – C leg immediately as this would result in an illegal configuration.

Site A R1 → Site B R22 ← Site C R1

It is necessary to first execute a half swap on the A – B link. It is also necessary to set the A – B link (B end at least) to acp copy mode else the swap on the B – C link fails with mode errors.

The following process was found to work. The order is critical to work around the cascaded SRDF restrictions. See comments in script regarding the issue.

Script `casc_unplanned_fo_sitec_with_C_to_B`

```
# This script does a failover to Site C for an unplanned failover (assume Site A not
available)
#
symrdf -g betall query; symrdf -g betall query -hop2
#
symrdf -g betall suspend -nop
symrdf -g betall failover -force -nop
#
# Have to make the Site B half of the pair an R1 so that I can change the mode of the link
# This can only be done for an R1
#
symrdf -g betall half_swap -hop2 -nop
#
# We need to get the A - B link out of sync else the B - C swap will not work as far hop
# must not be sync
#
# As the configuration is not no longer concurrent, a pairsfile has to be used
#
symrdf -f pairsfile_casc_717_919_betall -sid 717 -rdrg 11 set mode acp_disk -nop
#
# Now swap the B - C link
#
symrdf -g betall swap -nop
symrdf -g betall establish -nop
#
symrdf -g betall query; symrdf -g betall query -hop2
```

3.2.2 Change from C --> B to A --> B --> C

This script reverses the change in the previous section.

Script `casc_unplanned_fo_sitec_with_C_to_B_backout`

```
#!/bin/sh -x
#
# This script does REVERSE of the script a failover to Site C for an unplanned failover
(assume Site A not
available)
#
symrdf -g betall query; symrdf -g betall query -hop2
#
symrdf -g betall suspend -nop
symrdf -g betall failover -nop
```

```
symrdf -g betall swap -nop
symrdf -g betall establish -nop
symrdf -g betall query
#
# Note that this half swap (from R1 ro R2) seems to set the mode back to sync for
# the Site B end of things
#
symrdf -f pairsfile_casc_717_919_betall -sid 717 -rdfg 11 half_swap -nop
symrdf -g betall query -hop2
symrdf -g betall resume -hop2 -nop
symrdf -g betall query; symrdf -g betall query -hop2
```

3.2.3 Fast change from A --> B --> C to production at C

This is documented as the fastest possible failover process. No ongoing replication is set up.

The commands are to run as Site C. Note that the device group has to be set up manually at Site C as GNS remote mirroring is not supported in a cascaded environment.

```
SiteC# symrdf -g betall failover
```

Failover is fast (63 and 59 seconds during two test runs)

If and when Site A is available again, failback is.

```
SiteC# symrdf -g betall failback
```

4 Appendix B - Other Cascaded SRDF operational tasks

4.1 Adding to and removing disks from a Cascaded SRDF configuration

4.1.1 Using movepair

This could not be tested at the time of writing due to restrictions on the version of Beta code being used.

In addition, the source Symmetrix was running 5772 and hence it is not possible to move new devices into both legs of a cascaded configuration. Addition of devices just to hop 2 (5773 → 5773) will be tested when permitted.

4.1.2 Not using movepair

This more crude process is independent of microcode level.

This example show devices being moved from another cascaded group beta10 (simply due to availability) and added the the beta11 cascaded group. Composite Group test1cg includes disks from both groups, and hence is used for this test rather than updating the definition of the beta11 device group. In a real deployment, the beta11 DG would be of course need to be updated to include the new disks.

```
#
# Suspend both legs
#
symrdf -g beta10 suspend -nop
symrdf -g beta10 suspend -hop2 -nop
symrdf -g beta11 suspend -nop
symrdf -g beta11 suspend -hop2 -nop
#
# Set hop2 to async mode as I could not add disk when suspended in async mode
#
symrdf -g beta11 set mode acp_disk -hop2 -nop
#
# Delete pairs
#
symrdf -f pairsfile_casc_919_717_beta10 -sid 919 -rdfg 10 deletepair -nop
symrdf -f pairsfile_casc_717_3110_beta14 -sid 717 -rdfg 14 deletepair -nop
#
# createpairs
#
symrdf createpair -f pairsfile_casc_919_717_beta10 -sid 919 -rdfg 11 -type R1 -establish -
nop
#
symrdf createpair -f pairsfile_casc_717_3110_beta14 -sid 717 -rdfg 12 -type R1 -rdf_mode
acp_disk -establish -nop
#
symrdf -cg test1cg establish -nop
symrdf -cg test1cg establish -hop2 -nop
#
# Set hop2 back into async mode
# Should wait until copy finished for performance reasons
#
symrdf -cg test1cg verify -synchronized -i 10
symrdf -cg test1cg verify -synchronized -hop2 -i 10
symrdf -cg test1cg set mode async -hop2 -nop
```

```
#
#
symrdf -cg test1cg query -nop
symrdf -cg test1cg query -hop2 -nop
```

The following were points were noted.

- Target RDF groups (both hop1 and hop2) needs to be suspended
- Hop2 of target cascaded setup set to acp_disk as unable to add disk to it when target hop2 is in suspended async mode.
- The createpair needs to define the target RDF group -rdf_mode as acp_disk
- Hop2 needs to be set back to async mode when all the disks are synchronized.

4.2 Other Cascaded SRDF issues

4.2.1 Symrecover does not support for hop2 BCVs

Symrecover has not been updated to support hop2 BCVs.

EMC response:

Response from engineering is that hop2 BCV support will not be included with symrecover in Solutions Enabler 6.5, however, it is likely to be included in a future release of Solutions Enabler.

4.2.2 Cascaded SRDF remote mirroring

Citi have provided to EMC the following requirements for Cascaded SRDF remote mirroring:

The driver for this is to:

- a) Reduce steps needed to manage DGs/CGs
- b) Reduce errors in device groups when managed by hand
- c) Ensure people don't forget to set up the remote site DG/CG!

In a cascaded setup, I would want a DG/CG set up in Site A with R1s and Hop2BCVs setup (against the Site C R2) to be "mirrored" so that:

- 1) The RDF elements now refer to the B - C hop. This would now be hop1 in terms of commands issued from Site C and would of course be an RDF2 type
- 2) The BCVs are now local to the Site C Symm.

To be very explicit, if for example the Site A definition of a CG with hop2 BCVs was

```
CGTYPE_RDF1
S 000190100919 041 DEV003
S 000190100919 045 DEV004
S 000190100919 039 DEV001
S 000190100919 03D DEV002
D 000190100919 11 12 059 2BCV001
D 000190100919 11 12 05D 2BCV002
D 000190100919 11 12 061 2BCV003
```

D 000190100919 11 12 065 2BCV004

Then at Site C definition I would want is:

CGTYPE RDF2

S 000190103110 041 DEV003
S 000190103110 045 DEV004
S 000190103110 039 DEV001
S 000190103110 03D DEV002
B 000190103110 059 BCV001
B 000190103110 05D BCV002
B 000190103110 061 BCV003
B 000190103110 065 BCV004

5 Appendix C - Cascaded Star Failover Scenarios

The following assumes that Site B is a bunker and that you would never actually failover to that site. In all of the following, Site A is production (under normal circumstances) and Site C is the failover site.

1) Planned switches to Site C (Site A Symm still up)

- Change from A --> B --> C to C --> B (reversal of replication direction)
- Change from C --> B to A --> B --> C (reversal of the above change i.e. back to normal operations)
- Change from A --> B --> C to C --> A (reversal of replication direction)
- Change from C --> A to A --> B --> C (reversal of the above change i.e. back to normal operations)
- Fast change from A --> B --> C to production at C (quickest process to get applications up at Site C when Site A server/cluster fails)

2) Planned reconfiguration (Prod remains at Site A)

- When halt of workload site possible
 - From Cascaded to Concurrent
 - From Concurrent to Cascaded
- Without halting workload site
 - From Cascaded to Concurrent
 - From Concurrent to Cascaded

3) Transient Fault Recovery (Prod remains at Site A)

- Loss of Synchronous link in Cascaded configuration
 - Change from A --> B --> C to A --> C (provides ongoing replication)
 - Change from A --> C to A --> B --> C (reversal of the above change. This is strictly a planned switch after Site A is available again)
- Loss of Asynchronous link in Cascaded configuration (A --> B --> C to A --> C)
 - Change from A --> B --> C to A --> C (provides ongoing replication)

- Change from A --> C to A --> B --> C (reversal of the above change. This is strictly a planned switch after Site A is available again)

4) Unplanned switches (Site A Symm goes down)

- Change from A --> B --> C to C --> B (provides ongoing replication)
- Change from C --> B to A --> B --> C (reversal of the above change. This is strictly a planned switch after Site A is available again)
- Fast change from A --> B --> C to production at C (quickest process to get applications up at Site C)

All of the above switch types were tested. The following points were noted.

5.1 Planned failover to Site C (with ongoing C → B async replication)

The following results in Async replication to Site B and is of limited use in Citi

Also Cascaded RDF Star with workload at Site C is not supported by EMC

```
SiteA# symstar -cg nova halt -nop 135 sec
SiteC# symstar -cg nova switch -site Dallas -nop 10 sec
SiteC# symstar -cg nova connect -site NewJersey -nop 11 sec
SiteC# symstar -cg nova connect -site Manhattan -nop 14 sec
SiteC# symstar -cg nova protect -site NewJersey -nop 68 sec
```

Note that one does not protect Manhattan in this configuration. ACP_DISK replication from NewJersey to Manhattan will be taking place to stop Manhattan from getting too far behind. Sync replication on hop2 is not supported.

Dallas Site C ---- Async → **NewJersey Site B** --- acp_disk → **Manhattan Site A**

Failback process is:

```
SiteC# symstar -cg nova halt -nop
SiteA# symstar -cg nova switch -site Manhattan -nop
SiteA# symstar -cg nova connect -site NewJersey -nop
SiteA# symstar -cg nova connect -site Dallas -nop
SiteA# symstar -cg nova protect -site NewJersey -nop
SiteA# symstar -cg nova protect -site Dallas -nop
SiteA# symstar -cg nova enable -nop
SiteA# symstar -cg nova query
```

5.2 Planned failover to Site C (with ongoing C → A async replication)

This type of failover requires a reconfiguration from cascaded to concurrent Star.

This reconfiguration results in a configuration that gives consistent data at Site A when running out of Site C.

```
SiteA# symstar -cg nova halt -nop
SiteA# symstar -cg nova reconfigure -site Dallas -path Manhattan:Dallas -nop

SiteC# symstar -cg nova switch -site Dallas
SiteC# symstar -cg nova connect -site Manhattan
SiteC# symstar -cg nova connect -site NewJersey
SiteC# symstar -cg nova protect -site Manhattan
```

Note: The reconfiguration must be run from the workload site. Also it must be done BEFORE the switch, otherwise the reconfiguration needed to switch back to Site A will require a full resync. Of the SiteB – Site C leg.

At this point we have concurrent configuration

Dallas Site C ---- ACP → NewJersey Site B
Dallas Site C ---- Async → Manhattan Site A

The process to switch back to A → B → C is:

```
SiteC# symstar -cg nova halt -nop
SiteC# symstar -cg nova reconfigure -site Manhattan -path NewJersey:Manhattan -nop

SiteA# symstar -cg nova switch -site Manhattan
SiteA# symstar -cg nova connect -site NewJersey
SiteA# symstar -cg nova connect -site Dallas
SiteA# symstar -cg nova protect -site NewJersey
SiteA# symstar -cg nova protect -site Dallas
SiteA# symstar -cg nova enable
```

Note that at the time of writing, the “protect dallas” command always fails to reach RDF consistency due to CG corruption (PR-0087), but the process is in principle correct.

5.2.1 Note on criticality of command order

The order in the above process is critical. If you do execute the first reconfiguration AFTER the switch to Site C, a problem then comes to when I try to get back to A->B->C, as one is forced to do a full resync.

```
SiteC# symstar -cg nova halt -nop
SiteC# symstar -cg nova switch -site Dallas -nop

SiteC# symstar -cg nova reconfigure -site Manhattan -path Dallas:Manhattan -nop -full
```

Without the full, the last command gives error "star differential resync is not possible, cannot proceed unless -full is specified"

The helpful error message implies that this it is expected that a full resync will be needed. The correct sequence is to do the reconfiguration at the workload site BEFORE the switch.

5.2.2 Comparing runtime of cascaded and concurrent planned switches

The runtime of the planned switch processes between A and C are compared for Cascaded and Concurrent configurations.

Cascaded			Concurrent		
Site	Action	Runtime	Site	Action	Runtime
A	Halt	125	A	Halt	149
A	Reconfigure	14			
C	Switch C	35	C	Switch C	36
C	Connect B	19	C	Connect B	27
C	Protect B	77	C	Protect B	83
C	Connect A	24	C	Connect A	17
	TOTAL	294		TOTAL	312
C	Halt	129	C	Halt	132
C	Reconfigure	16			
A	Switch A	35	A	Switch A	33
A	Connect B	12	A	Connect B	17
A	Connect C	14	A	Connect C	23
A	Protect B	10	A	Protect B	12
A	Protect C	71	A	Protect C	80
A	Enable	194	A	Enable	141
	TOTAL	481		TOTAL	438

Planned switch runtime comparison between Cascaded and Concurrent Star

The following table compares runtime of concurrent planned switch operation between SE6.2, SE6.3 & and SE6.5.

Step	SE6.2 5771	SE6.3 5771	SE6.5 5773
Halt at Site A	2m 54s	2m 01s	2m 21s
Switch to Site C	5m 49s	3m 51s	2m 42s
TOTAL	8m 43s	5m 52s	5m 03s
Halt at Site C	2m 19s	2m 5s	2m 12s
Switch to Site A	12m 34s	5m 30s	5m 6s
TOTAL	14m 53s	7m 35s	7m 18s

Planned switch runtime comparison between SE/Microcode versions for Concurrent Star

In conclusion, SE6.5 shows some improvements in runtime compared with SE6.3, but nothing dramatic.

5.2.3 Fast change from A --> B --> C to production at C

This process is the quickest process to get applications up at Site C when Site A server/cluster fails. It is the same process as the planned switch, but without connecting and protecting the other sites. Hence, it just covers the halt, reconfigure and switch steps. The reconfigure should be included so that Site C can be protected later once the application is up and running.

From the table in the previous section, it can be seen that:

- For cascaded, the halt, reconfigure & switch steps take: 200 seconds
- For concurrent, the halt & switch steps take: 185 seconds

5.3 Planned Reconfiguration from Cascaded to Concurrent (and back)

5.3.1 When halt of workload site possible

This requires a reconfiguration from cascaded to concurrent Star, with workload remaining at Site A.

Initial configuration is:

Manhattan Site A ---- Sync → **NewJersey Site B** ---- Async → **Dallas Site C**

```
SiteA# symstar -cg nova halt -nop
SiteA# symstar -cg nova reconfigure -site Dallas -path Manhattan:Dallas -nop
SiteA# symstar -cg nova halt -reset -nop

SiteA# symstar -cg nova connect -site NewJersey -nop
SiteA# symstar -cg nova connect -site Dallas -nop
SiteA# symstar -cg nova protect -site NewJersey -nop
SiteA# symstar -cg nova protect -site Dallas -nop
SiteA# symstar -cg nova enable -nop
SiteA# symstar -cg nova query
```

At this point we have concurrent configuration

Manhattan Site A ---- Sync → **NewJersey Site B**
Manhattan Site A ---- Async → **Dallas Site C**

The process to configure back to cascaded is:

```
SiteA# symstar -cg nova halt -nop
SiteA# symstar -cg nova reconfigure -site Dallas -path NewJersey:Dallas -nop
SiteA# symstar -cg nova halt -reset -nop

SiteA# symstar -cg nova connect -site NewJersey -nop
SiteA# symstar -cg nova connect -site Dallas -nop
SiteA# symstar -cg nova protect -site NewJersey -nop
SiteA# symstar -cg nova protect -site Dallas -nop
SiteA# symstar -cg nova enable -nop
SiteA# symstar -cg nova query
```

Manhattan Site A ---- Sync → **NewJersey Site B** ---- Async → **Dallas Site C**

5.3.2 Without halting workload site

In this case, one need to inject an artifical error to suspend the RDF links to the Async site and then follow the process for transient fault management with reconfiguration. This process takes the system out of Star protected state and hence one must then protect it again.

The `-reset` is needed on the reconfigure command to transition the site state from PathFail to Disconnected prior to starting the Reconfiguration operation.

Action	CLI command	Runtime
Concurrent --> Cascaded		
Disconnect Site C	SiteA# sysmtar -cg nova disconnect -trip -site Dallas -nop	39s
Reconfigure to concurrent	Sitea# sysmtar -cg nova reconfigure -reset -site Dallas -path Manhattan:Dallas -nop	58s
Connect Site C again	SiteA# symstar -cg nova connect -site Dallas -nop	10s
Protect Site C again	SiteA# symstar -cg nova protect -site Dallas -nop	1m 21s
Enable Star	SiteA# symstar -cg nova enable -nop	2m 21s
Concurrent --> Cascaded		
Disconnect Site C	SiteA# sysmtar -cg nova disconnect -trip -site Dallas -nop	39s
Reconfigure to concurrent	Sitea# sysmtar -cg nova reconfigure -reset -site Dallas -path NewJersey:Dallas -nop	47s
Connect Site C again	SiteA# symstar -cg nova connect -site Dallas -nop	8s
Protect Site C again	SiteA# symstar -cg nova protect -site Dallas -nop	2m 12s
Enable Star	SiteA# symstar -cg nova enable -nop	2m 10s

Process and runtimes for planned reconfiguration without halting application

5.4 Transient Fault recovery

5.4.1 Loss of synchronous link in cascaded configuration

This is the process to recover from a transient failure on the Sync leg. SRDF/S does not have transmit idle and so would partition instead. On recovery of link, the following process (excluding the disconnect -trip) should be used.

0) Simulate a link failure to the Synchronous Link to Site B (NewJersey). Not part of recovery process!

```
SiteA# symstar -cg <cg_name> disconnect -trip -site NewJersey -nop
```

1) Reset Site B to the "Disconnected" state

```
SiteA# symstar -cg <cg name> reset -site NewJersey -nop
```

2) Unprotect B → C before protecting A → C to ensure that we don't end up with an acp – async configuration

```
SiteA# symstar -cg <cg name> unprotect -site Dallas -nop
```

3) Set up Star protection again

```
SiteA# symstar -cg <cg name> connect -site NewJersey -nop
SiteA# symstar -cg <cg name> protect -site NewJersey -nop
SiteA# symstar -cg <cg_name> protect -site Dallas
SiteA# symstar -cg <cg name> enable -nop
```

5.4.2 Loss of asynchronous link in cascaded configuration

This requires an unplanned reconfigure to concurrent SRDF to get A → C replication.

0) Simulate a link failure to the Asynchronous Link to Site C (Dallas). Not part of recovery process!

```
SiteA# symstar -cg <cg_name> disconnect -trip -site Dallas -nop
```

1) Reconfigure to concurrent to get direct replication from A → C

```
SiteA# symstar -cg <cg name> reconfigure -reset -site Dallas -path Manhattan:Dallas -nop
```

2) Set up Star protection again

```
SiteA# symstar -cg <cg name> connect -site Dallas -nop
SiteA# symstar -cg <cg name> protect -site Dallas -nop
SiteA# symstar -cg <cg name> enable -nop
```

5.5 Unplanned switch to Site C (Site A Symm goes down)

5.5.1 Change from A → B → C to C → B

The process is almost identical for Cascaded as it is for concurrent. The only different step is the first; Cascaded requires a disconnect -trip, whereas concurrent needs a symrdf suspend – immediate, to override transmit idle.

The full process is:

Steps for switching operations to Site C (Dallas) when Site A (Manhattan) is unavailable, keeping data from Site B (NewJersey).

```
SiteC# symstar -cg nova disconnect -trip -site Dallas -nop
SiteC# symstar -cg nova clean -site Dallas -nop
SiteC# symstar -cg nova switch -site Dallas -keep NewJersey -nop
SiteC# symstar -cg nova protect -site NewJersey -nop
```

When Site A is available again

```
SiteC# symstar -cg nova connect -site Manhattan -nop
```

Steps for switching operations back to Site A

```
SiteC# symstar -cg nova halt -nop

SiteA# symstar -cg nova switch -site Manhattan -nop
SiteA# symstar -cg nova connect -site NewJersey -nop
SiteA# symstar -cg nova connect -site Dallas -nop
SiteA# symstar -cg nova protect -site NewJersey -nop
SiteA# symstar -cg nova protect -site Dallas -nop
SiteA# symstar -cg nova enable -nop
SiteA# symstar -cg nova query
```

5.5.2 Comparing runtime of cascaded and concurrent unplanned switches

The runtime of the unplanned switch processes between A and C are compared for Cascaded and Concurrent configurations.

Note that in these tests there was no data to move between Sites B and C and hence these represent the overheads of Star state changes only.

Cascaded (SE6.5)			Concurrent (SE6.5)		
Site	Action	Runtime	Site	Action	Runtime
C	Disconnect -trip -site SiteC	38*	C	Symrdf suspend -immediate	6
C	Clean	4	C	Clean	4
C	Switch C keep B	32	C	Switch C keep B	29
	Sub-total	74		Sub-total	39
C	Protect B	69	C	Protect B	68
	GRAND TOTAL	143		GRAND TOTAL	107
	When Site A is back			When Site A is back	
C	Connect A	30	C	Connect A	50
C	Halt	115	C	Halt	120
A	Switch A	33	A	Switch A	34

A	Connect B	12		A	Connect B	15
A	Connect C	14		A	Connect C	26
A	Protect B	68		A	Protect B	74
A	Protect C	132		A	Protect C	139
A	Enable	135		A	Enable	139
	TOTAL	539			TOTAL	597

* This operation includes a cleanup window that lasts a minimum of 30 sec.

Unplanned switch runtime comparison between Cascaded and Concurrent Star

This table implies us that concurrent is faster than cascaded for failover. However, this is misleading as with cascaded, the “missing” data from Site C will have been copied down from Site B to Site C, and hence there will be no time needed to copy this data AFTER the switch, as one must do with Concurrent.

	Cascaded	Concurrent
Minimum time taken before apps can use storage at Site C (<i>excludes data copy time</i>)*	74 sec	39 sec
Time taken until Site C is protected by Site B	143 sec	107 sec
Time taken to switch operations back to Site A	539 sec	597 sec

* For Cascaded configuration, there will be no data copy time after the switch.

Unplanned switch runtime summary: Cascaded vs Concurrent Star

5.5.3 Comparing runtime of SE6.3 and SE6.5 concurrent unplanned switches

The following results show that switch times with SE6.5 are much faster than previous versions.

	End-to-end runtime			
Step	SE6.2 (sec)	SE6.3 (sec)	SE6.5 (sec)	SE6.5/6.3 ratio
Switch to Async, keeping Sync data & protect B	457	257	107	0.42
Halt and revert to workload site	656	580	597	1.03

Note: For SE6.5, this figure includes 6 seconds for the suspend –immediate needed when Transmit Idle is in use. This feature was not available during the SE6.3 testing.

Unplanned switch runtime comparison between SE/ucode versions for Concurrent Star

Cascaded SRDF and other 5773 features

5.5.4 Fast change from A --> B --> C to production at C

This process is the quickest process to get applications up at Site C when Site A fails. It is the same process as the unplanned switch described earlier, but without connecting and protecting the site B. Hence, it just covers the halt, reconfigure and switch steps. The reconfigure should be included so that Site C can be protected later once the application is up and running.

From the table in the pervious section section, it can be seen that for cascaded, the halt, reconfigure & switch steps take 74 seconds.

5.6 Other Cascaded Star issues

Testing revealed the following issues not previously known (to author).

5.6.1 Microcode requirements

The microcode requirements are a little different than regular Cascaded SRDF.

For regular Cascaded SRDF, only Site B needs to be running 5773 (to support the R21 device).

For Cascaded Star, sites A and B will need to be running 5773 code. Site C can be running 5x71 or higher. The reason that 5773 is required on sites A and B is because, in the event of a switch from site A to B, an R21 device will need to be configured in the site A symm. As the site C symm would not be configured with R21s at any time it does not require 5773.

5.6.2 Setup

For Cascaded Star, the options files must also have the entry:

```
SYMCLI_STAR_ALLOW_CASCADED_CONFIGURATION = Yes
```

```
SiteA# symstar -cg nova setup -options options
```

Then FTP star definition file to Site C

```
SiteC# symstar -cg nova buildcg -site Dallas
```

5.6.3 Star protection

```
SiteA# symstar -cg nova connect -site NewJersey -nop
SiteA# symstar -cg nova connect -site Dallas -nop
SiteA# symstar -cg nova query
SiteA# symstar -cg nova protect -site NewJersey -nop
SiteA# symstar -cg nova protect -site Dallas -nop
SiteA# symstar -cg nova query
SiteA# symstar -cg nova enable -nop
SiteA# symstar -cg nova query
```

5.6.4 Disabling Star

```
SiteA# symstar -cg nova disable -nop
SiteA# symstar -cg nova query

#
# New Jersey cannot be unprotected until Dallas has been unprotected
#

SiteA# symstar -cg nova unprotect -site Dallas -nop
SiteA# symstar -cg nova unprotect -site NewJersey -nop
SiteA# symstar -cg nova query
SiteA# symstar -cg nova disconnect -site NewJersey -nop
SiteA# symstar -cg nova disconnect -site Dallas -nop
SiteA# symstar -cg nova query
```

5.6.5 Connect fails with “Disabling RDF Consistency protection” error

This was seen a couple of times during testing. The cause was a lock# 386 (Usage "RDFA area") on the SiteA symm (3110). Once the lock was cleared, the connect would work correctly. EMC claim this will be fixed in the GA code.

6 Appendix D - Other notable issues encountered during Beta

6.1 Moving Dynamic RDF device pairs

6.1.1 Device groups not updated when pairs move

Testing showed that after a device in RDF group 14 is moved to RDF group 15, the information in the device group (beta14) is not updated in GNS to reflect change in RDF group being used for remote BCV, resulting in spurious "syndg show" output giving information on both RDF groups 14 and 15.

Also

```
# symmir -g beta14 query -rdf
```

will fail with error "An internal error occurred during a SYMAPI operation. Please report to EMC"

EMC response (Donal DeBurca):

```
The movepair command is not designed to automatically re-associate
remote BCVs within in the device group. This is in line with
previous behaviour when performing a deletepair/createpair
operation to move devices to a new RDF group.
```

```
A future code drop will include a more meaningful error message for
TimeFinder operations in this situation, rather than 'internal
error' being returned.
```

```
I suggest this issue be set to an enhancement request, such that if
a movepair command is run against a device group that there be an
option to reassociated remote BCVs that are defined within the
device group.
```

Additional EMC verbal response (Donal DeBurca):

```
This may be considered for a future SE version but will not be
available in SE6.5 as the necessary hooks are not available. Users
need to update device groups by hand using disassociate and
associate commands.
```

6.2 Minimum SRDF/A cycle time

Minimum cycle time can now be set to 1 seconds (3 sec of MSC). The actual cycle time varies from 1 sec (no I/O) to 4 sec (moderate I/O) and 6-11 sec for high I/O i.e. cycle time limited by data transfer time.

Minimum cycle time can now be set as low as 1 seconds, whereas minimum interval for symstat (and other) commands is still at 5 seconds. This impacts our ability to observe the flow of data through the SRDF/A replication process.

EMC response:

The response from engineering is that there are currently no plans to reduce the minimum interval allowed on symrdf query or symstat commands, nor are there likely to be in the future. The reason is that a small interval query places a large amount of syscall traffic on the symm in order to collect information to be returned to the query. Allowing intervals less than 5 seconds could cause significant impact on the symm.

6.3 GNS remote mirroring issue

Tested with Drop 3.

GNS remote mirroring was originally enabled on server sangs7 (which has Symm 3110 as a local system). Attempts to switch it off have failed. These attempts included:

Changing the entry in daemon_options file to

```
storgnsd:gns_remote_mirror = disable
```

and restarting storgnsd. Log file for this daemon shows that on restart it accepted the option, but remote mirroring is still taking place.

Also a stordaeon list -v command shows that the host thinks that GNS Mirroring is switched off.

```
# stordaeon list -v
[Other entries deleted]

[*] storgnsd                EMC Solutions Enabler GNS Daemon
-----
Daemon State                : Running
Daemon Start Time           : Mon Feb 25 03:58:05 2008
Version                     : T6.5-863 (0.261)
Auto-Restart by Watchdog    : Enabled

Total Number of Connections : 27
Number of Active Connections : 0
Total Number of Requests    : 106

RDF Remote Group Mirroring  : Disabled
GNS Data Backup (Shadow Files): Disabled
```

Workaround is to use storgnsd: gns_symavoid at the remote site, but this is just a workaround.

See Appendix E for a description of Citi's enhancement request to EMC to address this issue.

7 Appendix E – Enhancement requests submitted to EMC

This section only covers enhancement requests. Bug fixes reports are logged in feedbackcentral and are not repeated here.

Feedback central ID	Title	Description	Priority	EMC tracking number (if any)
PR-0012	Symdbg show giving spurious output after movepair operation	After device in RDF group 14 is moved to RDF group 15, the information in the device group (beta14) is not updated in GNS to reflect change in RDF group being used for remote BCV, resulting in spurious "symdbg show" output giving information on both RDF groups 14 and 15.	2	272827
PR-0051	Symrecover does not support hop2 BCVs	The symrecover command has not yet been updated to support hop2 BCVs so that you can automate resyncs of the SRDF/A leg from Site A after suspensions.	2	273834
PR-0067	No GNS remote mirroring support for Cascaded SRDF	We understand that GNS remote mirroring is not supported with cascaded SRDF, but would like to request that it is supported at Site C at some point.	1	274630
PR-0122	Enhanced configuration status in symstar query	It is of course possible with symstar query to determine in you are running in concurrent or cascaded mode by looking at the 2nd Target site and seeing if the source site is Site A or Site B. However, it would be helpful to operations teams if symstar query had a line that stated if the configuration were cascaded or concurrent. This is a low priority request, but probably very easy to implement.	3	275892

7.1 Further detail on GNS remote mirroring support request (274630)

Citi have provided to EMC the following requirements for Cascaded SRDF remote mirroring:

The driver for this is to:

- a) Reduce steps needed to manage DGs/CGs
- b) Reduce errors in device groups when managed by hand
- c) Ensure people don't forget to set up the remote site DG/CG!

In a cascaded setup, I would want a DG/CG set up in Site A with R1s and Hop2BCVs setup (against the Site C R2) to be "mirrored" so that:

1) The RDF elements now refer to the B - C hop. This would now be hop1 in terms of commands issued from Site C and would of course be an RDF2 type

2) The BCVs are now local to the Site C Symm.

To be very explicit, if for example the Site A definition of a CG with hop2 BCVs was

```
CGTYPE RDF1
S 000190100919 041 DEV003
S 000190100919 045 DEV004
S 000190100919 039 DEV001
S 000190100919 03D DEV002
D 000190100919 11 12 059 2BCV001
D 000190100919 11 12 05D 2BCV002
D 000190100919 11 12 061 2BCV003
D 000190100919 11 12 065 2BCV004
```

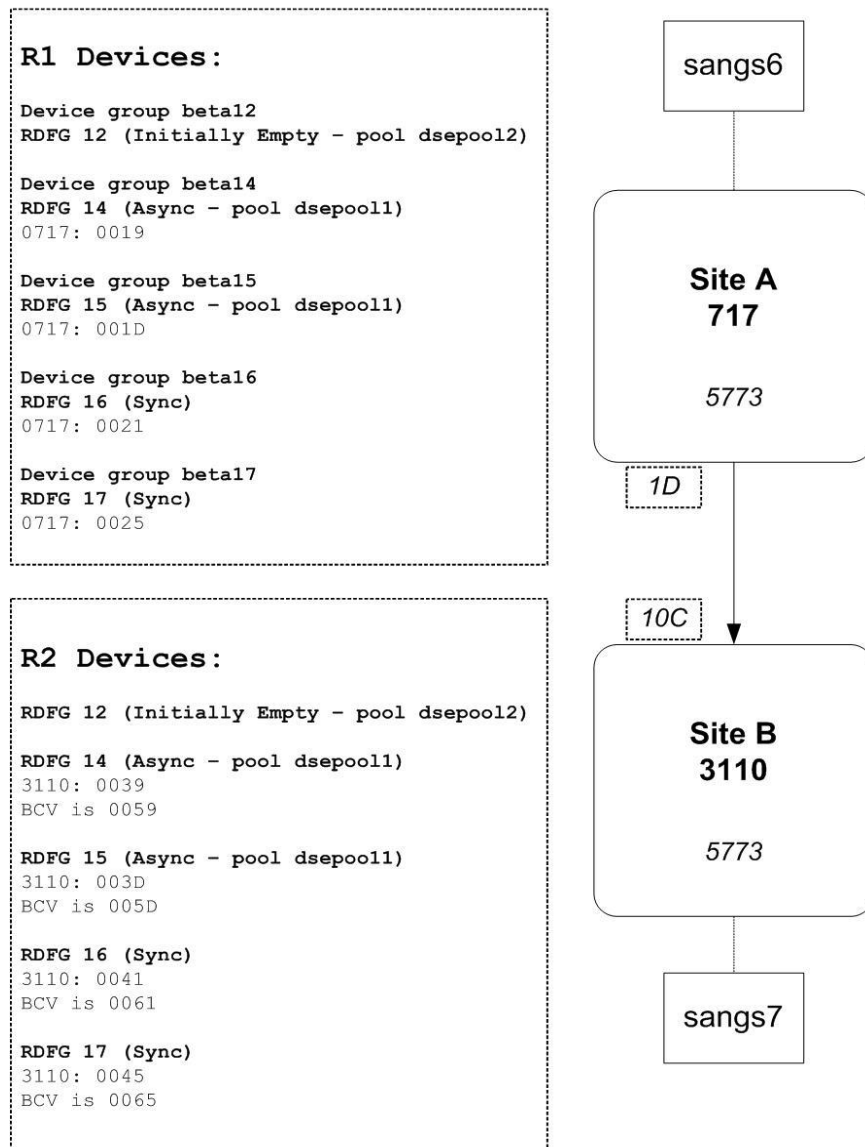
Then at Site C definition I would want is:

```
CGTYPE RDF2
S 000190103110 041 DEV003
S 000190103110 045 DEV004
S 000190103110 039 DEV001
S 000190103110 03D DEV002
B 000190103110 059 BCV001
B 000190103110 05D BCV002
B 000190103110 061 BCV003
B 000190103110 065 BCV004
```

8 Appendix B – Consolidated Test Results

8.1 5773 New features for 2-Site SRDF/A

RAID1 – 5773 2 Site

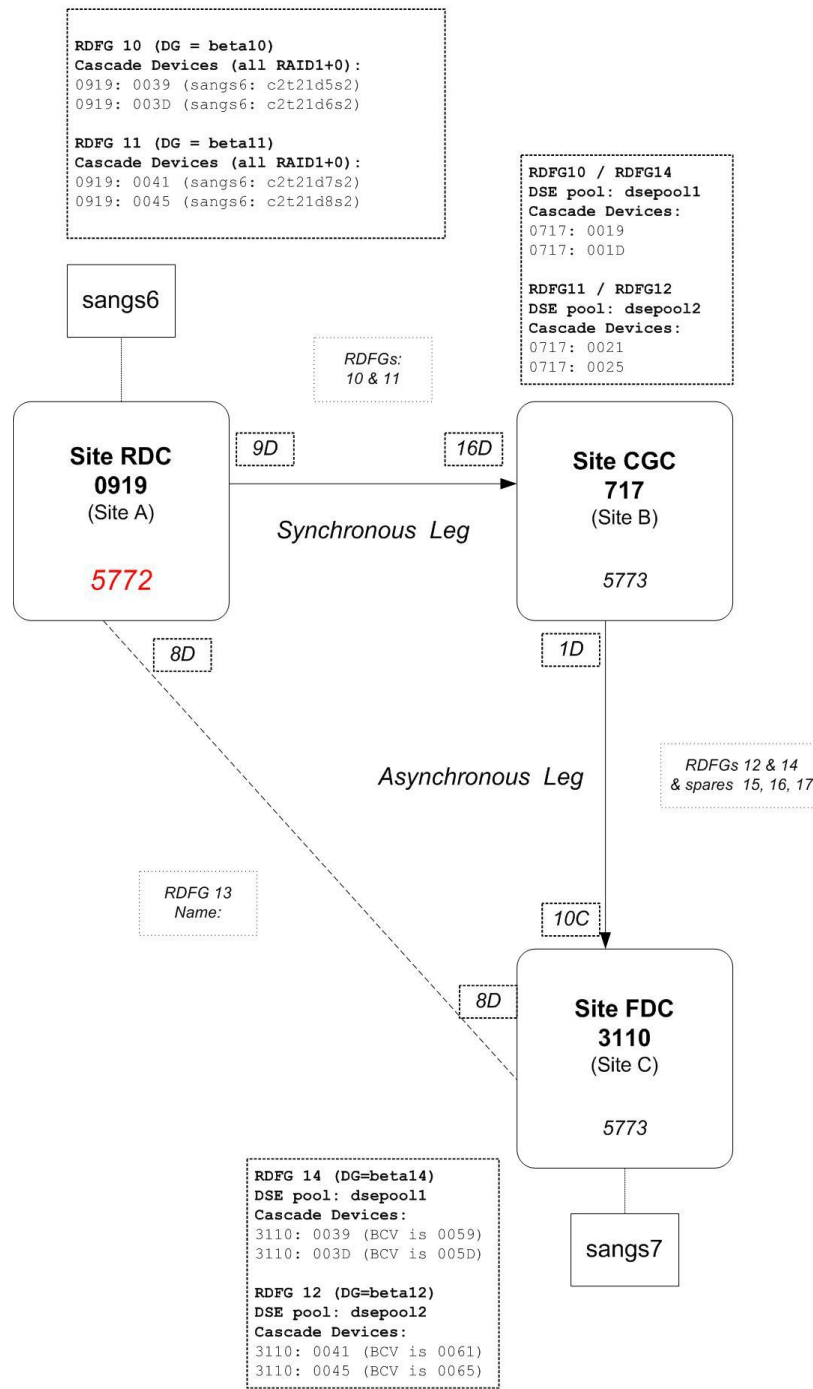


Reclaim IDE						
Section: Two Site - Basic SRDF Implementation (BRDF)						
Step	Task	Equivalent SYMCLI Command	Date Performed	Pass/Fail	OPT Number	Comments / Results
Basic SRDF Implementation						
From Site A Control Server - Build Dynamic RDF Groups						
BRDF.1.1	Verify RDF Connectivity between Site A and Site B Symmetrix Arrays	symcfg list -ra all -switched -sid <Site A SID>	29/01/2008	Pass		
BRDF.1.2	Verify the following parameters are enabled on Site A and Site B Symmetrix Arrays	symcfg list -v grep "Dynamic RDF Configuration State"	29/01/2008	Pass		
BRDF.1.3	If required - Enable Symmetrix Arrays at Site A and Site B for Dynamic SRDF	symconfigure -sid ### commit -cmd "set symmetrix dynamic_rdfenable;"	N/A	Not Performed		
BRDF.1.4	Create Dynamic RDF Group (label "beta14", number 14) between Site A and Site B Symms	symrdf addgrp -label beta14 -sid ### -rdgf 14 -dir ## -remote_sid ### -remote_rdgf 14 -remote_dir ##	29/01/2008	Pass		
BRDF.1.5	Create Dynamic RDF Group (label "beta15", number 15) between Site A and Site B Symms	symrdf addgrp -label beta15 -sid ### -rdgf 15 -dir ## -remote_sid ### -remote_rdgf 15 -remote_dir ##	29/01/2008	Pass		
BRDF.1.6	Create Dynamic RDF Group (label "beta16", number 16) between Site A and Site B Symms	symrdf addgrp -label beta16 -sid ### -rdgf 16 -dir ## -remote_sid ### -remote_rdgf 16 -remote_dir ##	29/01/2008	Pass		
BRDF.1.7	Create Dynamic RDF Group (label "beta17", number 17) between Site A and Site B Symms	symrdf addgrp -label beta17 -sid ### -rdgf 17 -dir ## -remote_sid ### -remote_rdgf 17 -remote_dir ##	29/01/2008	Pass		
BRDF.1.8	Create Dynamic RDF Group (label "beta12", number 12) between Site A and Site B Symms	symrdf addgrp -label beta12 -sid ### -rdgf 12 -dir ## -remote_sid ### -remote_rdgf 12 -remote_dir ##	29/01/2008	Pass		
BRDF.1.9	Verify dynamic rdf group added successfully	symcfg list -ra all -switched -sid <Site A SID>	29/01/2008	Pass		
If Required - Enable Transmit lde on non 5773 arrays						
BRDF.2.1	If required, enable Transmit lde on both Site A and Site B Symmetrix Arrays. Note: For 5773 arrays, Transmit lde is enabled by default		N/A	Not Performed		
BRDF.2.1.1	Check the state of Transmit lde on both the Source and Target arrays	symcfg list -rdgf all -rdfa	29/01/2008	Pass		
BRDF.2.1.2	Change the Transmit lde state to ENABLED on both Site A and Site B arrays for RDGF 14	symconfigure -sid ### commit -cmd "set rdf group 14 rdfa_transmit_lde = ENABLE;"	N/A	Not Performed		Transmit lde enable by default when RDF group created
BRDF.2.1.3	Change the Transmit lde state to ENABLED on both Site A and Site B arrays for RDGF 15	symconfigure -sid ### commit -cmd "set rdf group 15 rdfa_transmit_lde = ENABLE;"	N/A	Not Performed		Transmit lde enable by default when RDF group created
BRDF.2.1.4	Change the Transmit lde state to ENABLED on both Site A and Site B arrays for RDGF 16	symconfigure -sid ### commit -cmd "set rdf group 16 rdfa_transmit_lde = ENABLE;"	N/A	Not Performed		Transmit lde enable by default when RDF group created
BRDF.2.1.5	Change the Transmit lde state to ENABLED on both Site A and Site B arrays for RDGF 17	symconfigure -sid ### commit -cmd "set rdf group 17 rdfa_transmit_lde = ENABLE;"	N/A	Not Performed		Transmit lde enable by default when RDF group created
BRDF.2.1.6	Change the Transmit lde state to ENABLED on both Site A and Site B arrays for RDGF 12	symconfigure -sid ### commit -cmd "set rdf group 12 rdfa_transmit_lde = ENABLE;"	N/A	Not Performed		Transmit lde enable by default when RDF group created
BRDF.2.2	Using SYMCLI - Examine properties of all new dynamic rdf group. Ensure that Transm lde is enabled. Compare results to SMC	symcfg list -rdgf ## -sid ### -rdfa	29/01/2008	Not Performed		
Configure DSE Devices and Pools for Site A and Site B						
BRDF.3.1	If required, disable 15 save device from the default pool		N/A	Not Performed		
BRDF.3.1.1	Examine Default Pool Devices	symcfg list -sid ### -pool DEFAULT_POOL -snap	29/01/2008	Pass		Nothing in that pool
BRDF.3.1.2	Disable 15 save devices in the Default Pool	symconfigure -sid ### commit -cmd "disable dev <start end> in pool DEFAULT_POOL, type=snap;"	N/A	Not Performed		Nothing in that pool
BRDF.3.1.3	Remove 15 save devices from the Default Pool	symconfigure -sid ### commit -cmd "remove dev <start end> from pool DEFAULT_POOL, type=snap;"	N/A	Not Performed		
BRDF.3.2	Create DSE device pools		29/01/2008	Pass		
BRDF.3.2.1	Create DSE Pool "despool1"	symconfigure -sid <sid> commit -cmd "create pool despool1, type=rdfa_dae;"	29/01/2008	Pass		
BRDF.3.2.2	Create DSE Pool "despool2"	symconfigure -sid <sid> commit -cmd "create pool despool2, type=rdfa_dae;"	29/01/2008	Pass		
BRDF.3.2.3	If Testing Cascading STAR - Create DSE Pool "recoptool"	symconfigure -sid <sid> commit -cmd "create pool recoptool, type=rdfa_dae;"	N/A	Not Performed		Will reuse existing DSE pool
BRDF.3.2.4	Optional - Create DSE Pool "despool3"	symconfigure -sid <sid> commit -cmd "create pool despool3, type=rdfa_dae;"	N/A	Not Performed		
BRDF.3.3	Assign 5 Save Devices to each DSE pool		29/01/2008	Pass		
BRDF.3.3.4	Assign 16 SAVE devices to DSE device pool "despool1" and enable the devices.	symconfigure -sid <sid> commit -cmd "add dev <SymDevName:SymDevName> to pool <pool_name> type=rdfa_dae	29/01/2008	Pass		
BRDF.3.3.5	Assign 16 SAVE devices to DSE device pool "despool2" and enable the devices.	symconfigure -sid <sid> commit -cmd "add dev <SymDevName:SymDevName> to pool <pool_name> type=rdfa_dae	29/01/2008	Pass		
BRDF.3.3.6	Fltting Cascading STAR - Assign 5 SAVE devices to DSE device pool "recoptool" and enable the devices.	symconfigure -sid <sid> commit -cmd "add dev <SymDevName:SymDevName> to pool <pool_name> type=rdfa_dae	N/A	Not Performed		Will reuse existing DSE pool
BRDF.3.3.7	Optional - Assign 5 SAVE devices to DSE device pool "despool3" and enable the devices.	symconfigure -sid <sid> commit -cmd "add dev <SymDevName:SymDevName> to pool <pool_name> type=rdfa_dae	N/A	Not Performed		Will reuse existing DSE pool
BRDF.3.4	Enable DSE / Bind RDF Groups to DSE Pools		29/01/2008	Pass		
BRDF.3.4.1	Enable DSE for RDGF 14. Bind to Pool "despool1". Set a DSE threshold of 20% and enable autostart	symconfigure -sid <sid> commit -cmd "set rdf group 14 rdfa_dae_pool=despool1 emulation=fa rdfa_dae_threshold=20	29/01/2008	Pass		Executed on both source and target Symm
BRDF.3.4.2	Enable DSE for RDGF 15. Bind to Pool "despool1". Set a DSE threshold of 20% and enable autostart	symconfigure -sid <sid> commit -cmd "set rdf group 15 rdfa_dae_pool=despool1 emulation=fa rdfa_dae_threshold=20	29/01/2008	Pass		Executed on both source and target Symm
BRDF.3.4.3	Enable DSE for RDGF 12. Bind to Pool "despool2". Set a DSE threshold of 20% and enable autostart	symconfigure -sid <sid> commit -cmd "set rdf group 12 rdfa_dae_pool=despool2 emulation=fa rdfa_dae_threshold=20	29/01/2008	Pass		Executed on both source and target Symm
BRDF.3.4.4	Optional - Enable DSE for RDGF xx. Bind to Pool "despool3". Set a DSE threshold of 20% and enable autostart	symconfigure -sid <sid> commit -cmd "set rdf group xx rdfa_dae_pool=despool3 emulation=fa rdfa_dae_threshold=20	N/A	Not Performed		
BRDF.3.4.5	Optional - Enable DSE for RDGF xx. Bind to Pool "despool3". Set a DSE threshold of 20% and enable autostart	symconfigure -sid <sid> commit -cmd "set rdf group xx rdfa_dae_pool=despool3 emulation=fa rdfa_dae_threshold=20	N/A	Not Performed		
BRDF.3.5	Examine DSE Devices and Pools		29/01/2008	Pass		
BRDF.3.5.1	Examine the DSE devices	symcfg show -pool <pool_name> -rdfa_dae	29/01/2008	Pass		Shows devices in pools. Session status is inactive for all at this stage.
BRDF.3.5.2	Examine the DSE pools, and threshold setting.	symcfg list -pool -rdfa_dae	29/01/2008	Pass		
BRDF.3.5.3	Examine the DSE threshold settings.	symcfg list -rdgf ## -rdfa	29/01/2008	Pass		All set to 20%, as expected
BRDF.3.6	Repeat above steps for Site B Target Array - These steps can be performed from Site A Control Server		29/01/2008	Pass		
Create Dynamic RDF Pairs, set the mode to ACP_disk, and perform full establish						
BRDF.4.1	Verify STD devices in device groups are enabled for dynamic rdf - if using reservations, use reservation to filter device selection list or use meta filter	symdev list -sid ### -range <startend> -dynamic -reserved	30/01/2008	Pass		No reserved devices so reserved dropped
BRDF.4.2	If required - convert devices in device groups to dynamic rdf enabled devices	symconfigure -sid ### commit -cmd "set dev <startend> attribute=dyn_rdf;"	N/A	Not Performed		Not required
BRDF.4.3	Verify that the are dynamic rdf capable devices on the Site B symmetrix	symdev list -sid <Site B SID> -dynamic	30/01/2008	Pass		
BRDF.4.4	If required - convert candidate devices in Site B symm to dynamic rdf enabled devices	symconfigure -sid <Site B SID> commit -cmd "set dev <startend> attribute=dyn_rdf;"	N/A	Not Performed		
BRDF.4.5	Create Dynamic SRDF Pairs by changing STDs in beta14 to RDF Configuration. Use RDF Group 14	symrdf createpair -f <device file beta14> -sid <Site A SID> -rdgf 14 -type RI -rdf mode asp_disk -establish -g beta14	30/01/2008	Pass		
BRDF.4.6	Create Dynamic SRDF Pairs by changing STDs in beta15 to RDF Configuration. Use RDF Group 15	symrdf createpair -f <device file beta15> -sid <Site A SID> -rdgf 15 -type RI -rdf mode asp_disk -establish -g beta15	30/01/2008	Pass		
BRDF.4.7	Create Dynamic SRDF Pairs by changing STDs in beta16 to RDF Configuration. Use RDF Group 16	symrdf createpair -f <device file beta16> -sid <Site A SID> -rdgf 16 -type RI -rdf mode asp_disk -establish -g beta16	30/01/2008	Pass		
BRDF.4.8	Create Dynamic SRDF Pairs by changing STDs in beta17 to RDF Configuration. Use RDF Group 17	symrdf createpair -f <device file beta17> -sid <Site A SID> -rdgf 17 -type RI -rdf mode asp_disk -establish -g beta17	30/01/2008	Pass		
BRDF.4.9	Using SYMCLI - monitor SRDF the establish for the device groups	symrdf -g <dg name> query -i 5	30/01/2008	Pass		

BRDF-5	Change SROF Mode to Asynchronous for testIdg and testDg				
BRDF 5.1	Using SYMCLI monitor the full SROF establish for all device groups - Confirm ACP_Disk mode	<code>symconf -g <dg name> query</code>	30/01/2009	Pass	
BRDF 5.1.1	Ensure that the mode for beta14 is ACP_Disk (C D)	<code>symconf -g beta14 query</code>	30/01/2009	Pass	
BRDF 5.1.2	Ensure that the mode for beta15 is ACP_Disk (C D)	<code>symconf -g beta15 query</code>	30/01/2009	Pass	
BRDF 5.2	Change SROF Mode from ACP_Disk to Async for beta14 and beta15 once invalid tracks are less than 30000	<code>symconf -g <dg name> set mode async -nop</code>	30/01/2009	Pass	
BRDF 5.3	Enable SROF consistency protection for testIdg and testDg	<code>symconf -g <dg name> enable -nop</code>	30/01/2009	Pass	
BRDF 5.4	Using SYMCLI - monitor SROF the establish for the device groups testIdg and testDg	<code>symconf -g <dg name> query -rdfa -i 5</code>	30/01/2009	Pass	
BRDF-6	Optional - Change SROF Mode to Synchronous for testDg and testIdg				
BRDF 6.1	Using SYMCLI monitor the full SROF establish for all device groups - Confirm ACP_Disk mode	<code>symconf -g <dg name> query</code>	30/01/2009	Pass	
BRDF 6.2	Change SROF Mode from ACP_Disk to Sync for beta16 and beta17	<code>symconf -g <dg name> set mode sync -nop</code>	30/01/2009	Pass	
BRDF 6.3	Using SYMCLI - monitor SROF the establish for the device groups	<code>symconf -g <dg name> query -i 5</code>	30/01/2009	Pass	
BRDF 6.4	Optional - Change SROF/A Session Priority for testIdg (RDFG 16) to 10	<code>symconfig -add \$\$\$ commands -cmd "set xa group 37, session_priority=10"</code>	30/01/2009	Pass	Executed at both source and target sites
BRDF 6.5	Optional - using SYMCLI - examine session priority in symcli query output - Compare to SMC	<code>symconf -g testIdg query -rdfa</code>	30/01/2009	Pass	SMC not tested
BRDF-7	Verify DSE is active for testIdg and testDg				
BRDF 7.1	Examine the RDF state of beta14 - Verify that DSE is active	<code>symconf -g beta14 query -rdfa</code>	30/01/2009	Pass	
BRDF 7.2	Examine the RDF state of beta15 - Verify that DSE is active	<code>symconf -g beta15 query -rdfa</code>	30/01/2009	Pass	
BRDF 7.3	If necessary, Enable DSE for both beta14 and beta15	<code>symconf -g <dg name> activate -rdfa_dse -nop</code>	N/A	Not Performed	N/A
BRDF 7.4	Optional - DSE Monitoring		31/01/2009	Pass	
BRDF 7.4.1	Generate I/O the SROF R1 devices in testIdg and testDg		31/01/2009	Pass	
BRDF 7.4.2	Watch the DSE information in query output and ensure that it is using a significant amount of cache, but not spilling over. Adjust the workload as necessary.	<code>symconf -g <dg name> query -rdfa -i 10</code>	31/01/2009	Pass	Using
BRDF 7.4.3	Create a second workload simulating a burst or bursts of I/O activity		31/01/2009	Pass	Disabled network link instead as easier to invoke DSE that way
BRDF 7.4.4	Monitor beta14 & beta15 for tracks spilling over on both sides of the link	<code>symconf -g <dg name> query -rdfa -i 10</code>	31/01/2009	Pass	Also used symcli -type cache as I can monitor both at once this way
BRDF 7.4.5	Check the total percent full of the SAVV devices in the DSE pool	<code>symconf -add <cid> monitor -exp <PoolName> [-i Interval] [-o Count]</code>	31/01/2009	Pass	Watch percentage full creep up in pool display!
BRDF 7.4.6	When the burst activity stops, monitor until SROF/A recovers and is no longer spilling over.	<code>symconf -g <dg name> query -rdfa, # symconf -add <cid> monitor -exp <PoolName> [-i Interval] [-o Count]</code>	31/01/2009	Pass	Re-enable network and watch percentage of dse pool usage decrease to zero
BRDF-8	Add remote BCVs from Site B Symmetric to device groups and Capture a Gold Copy				
BRDF 8.1	Add 1 remote RBCV devices from Site B Symm to device group beta14		30/01/2009	Pass	ALL BCVs SET UP USING CLONE EMULATION SYMCLI CLONE EMULATION-ENABLED
BRDF 8.1.1	List the BCV devices present in the Site B Symmetric	<code>symdev list -aid <Site B SID> -meta -bvv -noopen</code>	30/01/2009	Pass	Excluded - report as I only have mirrored BCVs in the box
BRDF 8.1.2	Add RBCV devices to device group beta14	<code>symdev -g beta14 associateall dev -range <start:end> -aid <Site A SID> -rdt -nop</code>	30/01/2009	Pass	Added BCV 0605 in 3110. Note that you have to specify Site A when using -rdt
BRDF 8.1.3	Re-examine beta14	<code>symdg show beta14</code>	30/01/2009	Pass	
BRDF 8.2	Add 1 remote RBCV devices from Site B Symm to device group beta15		30/01/2009	Pass	
BRDF 8.2.1	List the BCV devices present in the Site B Symmetric	<code>symdev list -aid <Site B SID> -meta -bvv -noopen</code>	30/01/2009	Pass	Excluded - report as I only have mirrored BCVs in the box
BRDF 8.2.2	Add RBCV devices to device group beta15	<code>symdev -g beta15 associateall dev -range <start:end> -aid <Site A SID> -rdt -nop</code>	30/01/2009	Pass	Added BCV 0602 in 3110. Note that you have to specify Site A when using -rdt
BRDF 8.2.3	Re-examine beta15	<code>symdg show beta15</code>	30/01/2009	Pass	
BRDF 8.3	Add 1 remote RBCV devices from Site B Symm to device group beta16		30/01/2009	Pass	
BRDF 8.3.1	List the BCV devices present in the Site B Symmetric	<code>symdev list -aid <Site B SID> -meta -bvv -noopen</code>	30/01/2009	Pass	Excluded - report as I only have mirrored BCVs in the box
BRDF 8.3.2	Add RBCV devices to device group beta16	<code>symdev -g beta16 associateall dev -range <start:end> -aid <Site A SID> -rdt -nop</code>	30/01/2009	Pass	Added BCV 0606 in 3110. Note that you have to specify Site A when using -rdt
BRDF 8.3.3	Re-examine beta16	<code>symdg show beta16</code>	30/01/2009	Pass	
BRDF 8.4	Add 1 remote RBCV devices from Site B Symm to device group beta17		30/01/2009	Pass	
BRDF 8.4.1	List the BCV devices present in the Site B Symmetric	<code>symdev list -aid <Site B SID> -meta -bvv -noopen</code>	30/01/2009	Pass	Excluded - report as I only have mirrored BCVs in the box
BRDF 8.4.2	Add RBCV devices to device group beta17	<code>symdev -g beta17 associateall dev -range <start:end> -aid <Site A SID> -rdt -nop</code>	30/01/2009	Pass	Added BCV 0605 in 3110. Note that you have to specify Site A when using -rdt
BRDF 8.4.3	Re-examine beta17	<code>symdg show beta17</code>	30/01/2009	Pass	
BRDF 8.5	Perform a full establish on the remote R2 / RBCV pairs for all device groups		30/01/2009	Pass	
BRDF 8.6	Monitor the Timeslider full establish for all device groups	<code>symmir -g <dg name> query -i 5 -rdt</code>	30/01/2009	Pass	
BRDF 8.7	Split the R2/RBCV pairs using the -consistent option for each device group once the pairs are establish	<code>symmir -g <dg name> split -consistent -rdt -nop</code>	30/01/2009	Pass	
BRDF 8.8	Monitor the Timeslider split for all device groups	<code>symmir -g <dg name> query -i 5 -rdt -bg -percent</code>	30/01/2009	Pass	
BRDF 8.10	Optional: Once the R2/RBCV pairs are split mount the RBCV for device groups testIdg and testDg to Test Host C		N/A	Not Performed	
BRDF-9	Basic SROF Operations				
BRDF-9	Optional - SMC SROF Monitoring				
BRDF-10	Move SROF/A Device Pairs between RDF Groups				
BRDF-11	Build a composite group and examine MSC Consistency				
BRDF 11.1	If required, provision extra gatekeepers to Site A control server so that it has a minimum of 8	<code>symmk -add <SymID> -non chba vnm add devx <start:end> -aid <FA #> -p <port #></code>	N/A	Not Performed	Already have 8 GKs available. Will use these
BRDF 11.2	On Site A control server, verify that the storadm daemon is running	<code>storadmmon list</code>	31/01/2009	Pass	
BRDF 11.3	On Site A control server, examine SYMAP/Options file and make sure that the parameter SYMAP_USE_RDFD is ENABLED	<code>more /var/symapi/options grep SYMAP_USE_RDFD</code>	31/01/2009	Pass	
BRDF 11.4	If required, start storadm on SYMAP server	<code>storadmmon start storadm</code>	31/01/2009	Pass	Not required but I restarted it anyway
BRDF 11.5	Optional - Establish R2/RBCV pairs at Site B and Capture a gold copy for beta14 and beta15	<code>symmir -g <dg name> establish -rdt -nop</code> <code>symmir -g <dg name> split -consistent -rdt -nop</code>	31/01/2009	Pass	Note Beta14 is in RDFG 14 and Beta15 in RDFG 15 for this test (i.e. I used mover to move back to original groups before starting this test. Both RDF groups set back to async mode, as they get reset to sync when empty of any luns and then luns are readded.
BRDF 11.6	Disable RDF/A consistency protection for beta14 and beta15	<code>symconf -g <dg name> disable</code>	31/01/2009	Pass	
BRDF 11.7	Build RCP1 type composite group testIdg using devices in beta14 and beta15 - create using -rdt_consistency option		31/01/2009	Pass	
BRDF 11.7.1	Add STD and RBCV devices from beta14 to testIdg using the symdg dgIdg option. Enable testIdg for rdt consistency protection	<code>symdg dgIdg beta14 testing -rdt_consistency</code>	31/01/2009	Pass	
BRDF 11.7.2	Add STD and RBCV devices from beta15 to testIdg using the symdg dgIdg -rename option.	<code>symdg dgIdg beta15 testing -rename</code>	31/01/2009	Pass	
BRDF 11.7.3	Verify that testIdg was created successfully. Note the amount of STD devices and RBCVs	<code>symcg list</code>	31/01/2009	Pass	
BRDF 11.7.4	Examine testIdg details	<code>symcg show testing</code>	31/01/2009	Pass	
BRDF 11.8	Examine the RDF status for testIdg - verify that DSE is active for both RDF groups	<code>symconf -g testing query</code>	31/01/2009	Pass	DSE active
BRDF 11.9	Examine the Timeslider status for the R2/RBCV pairs for testIdg	<code>symmir -g testing query -rdt -rdsg all</code>	31/01/2009	Pass	All BCVs split off
BRDF 11.10	Enable MSC for testIdg	<code>symcg -g testing enable -nop</code>	31/01/2009	Pass	
BRDF 11.11	Examine the SROF Status for testIdg - Verify that the MDAC mode is "A_X"	<code>symconf -g testing query -rdfa -i 5</code>	31/01/2009	Pass	Timestamp noted. MDAC is A_X
BRDF-12	Perform an MSC Trip for Composite Group and Examine SROF/A Timestamp				
BRDF 12.1	On the Fabric Switch - Offline the port(s) the RF de(s) RDFG 14 is assigned to		31/01/2009	Partial Pass	Both RDF groups are on SAME port so MSC trip test is compromised
BRDF 12.2	After a few moments - Query the RDF status and examine the SROF/A Timestamp	<code>symconf -g testing query -rdfa</code>	31/01/2009	Pass	
BRDF 12.2.1	Verify that the RDF Pair State for both RDFGs in testIdg is "Transient Idle"		31/01/2009	Pass	RDF pair state is Transient
BRDF 12.2.2	View the "time lag" of the R2 displayed by Time that R2 is behind R1		31/01/2009	Pass	Time R2 is behind R1 is increasing
BRDF 12.2.3	View the timestamp of the R2 data displayed by R2 Image Capture Time		31/01/2009	Pass	R2 timestamp static at 08:50:20
BRDF 12.2.4	Verify that the time lag and timestamp compared to each other		31/01/2009	Pass	At 09:00:21, time stamps was 08:50:21 and "Time R2 is behind R1" is 10:00". Hence, this is spot on to the second. Subsequent tests showed the same result
BRDF 12.3	On the Fabric Switch - Online the port(s) the RF de(s) RDFG 14 is assigned to		31/01/2009	Pass	
BRDF 12.4	After a few moments - Query the RDF status and examine the SROF/A Timestamp	<code>symconf -g testing query -rdfa</code>	31/01/2009	Pass	
BRDF 12.4.1	Verify that the RDF Pair State for both RDFGs in testIdg is "Consistent or Suspended"		31/01/2009	Pass	Transient life was makes do reverts to consistent once the network link returns
BRDF 12.4.2	View the "time lag" of the R2 displayed by Time that R2 is behind R1 - Verify that it is decreasing		31/01/2009	Pass	Initially increasing waiting for first cycle switch since outage. Then it reverts to the usual 30 -> 60 -> 30 cycle
BRDF 12.4.3	View the timestamp of the R2 data displayed by R2 Image Capture Time		31/01/2009	Pass	Update whenever a cycle switch occurs

8.2 Cascaded SRDF – Base testing

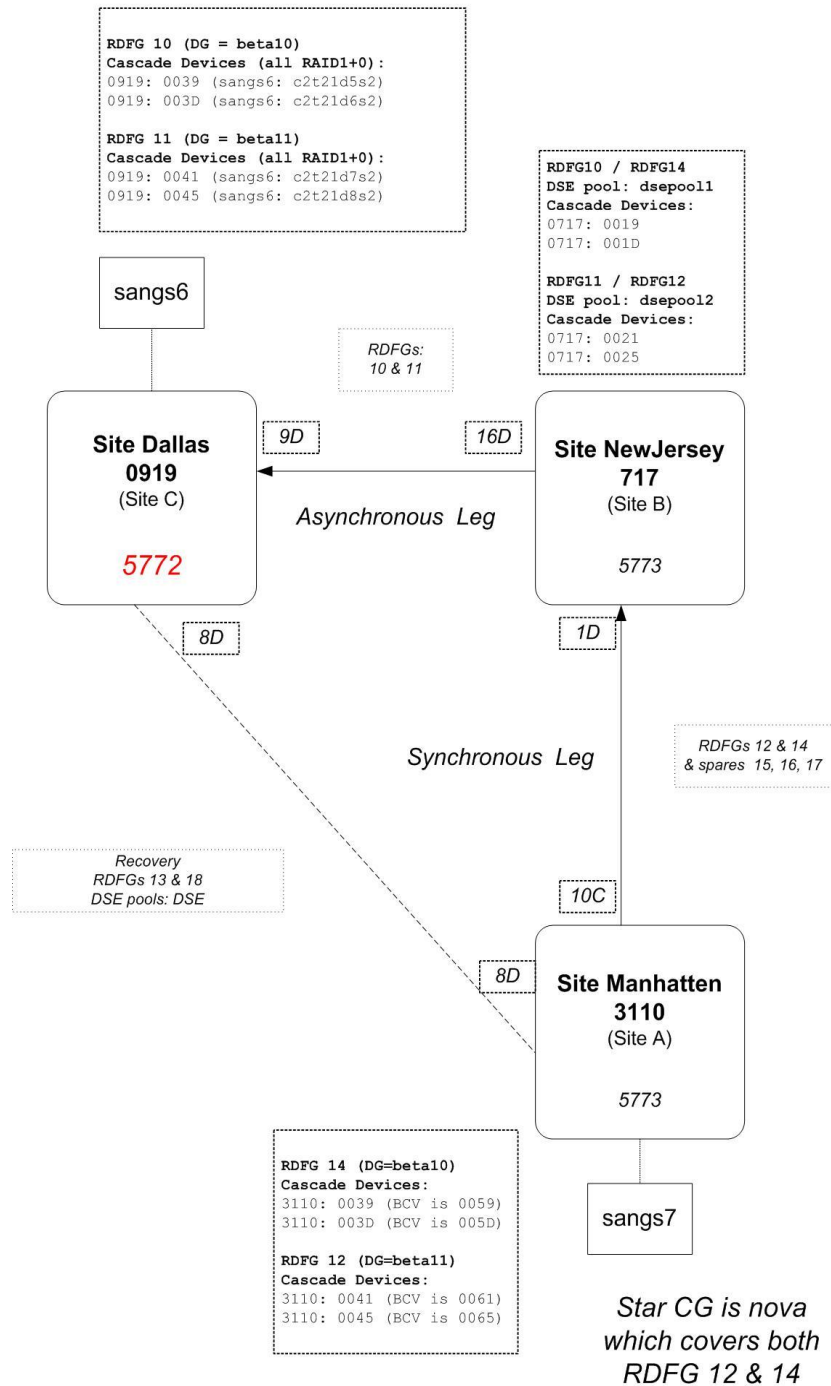
RAID1 – Cascaded SRDF



From Site A Control Server - Enable RDF Consistency Protection for both legs of test1cg					
Change RDF mode into Asynchronous for the hop 2 leg (Site B -> Site C) for test1cg	symrdf -cg test1cg set mode async -hop2 -nop	04/02/2009	Pass		
Enable ECA Consistency for the Site A -> Site B leg of test1cg	symcg -cg test1cg enable -nop	04/02/2009	Pass		
Enable MSC Consistency Protection for the hop 2 leg for test1cg	symcg -cg test1cg enable -hop2 -nop	04/02/2009	Pass		
Examine the RDF States and Modes for both legs of test1cg	symrdf -cg test1cg query -detail symrdf -cg test1cg query -rdfa -hop2 symrdf -cg test1cg query -detail -hop2	04/02/2009	Pass		
Examine the details of test1cg	symcg show test1cg	04/02/2009	Pass		
If necessary activate DSE for test1cg	symrdf -og test1cg activate -rdfa_dse -hop2 -nop	04/02/2009	Pass		
Optional : Invoke DSE Spillover on hop2 for test1cg					
Verify that DSE is active on both Site B and Site C	symrdf -og test1cg query -detail -hop2	04/02/2009	Pass		
If necessary activate DSE for test1cg	symrdf -og test1cg activate -rdfa_dse -hop2 -nop	04/02/2009	Pass		
Generate I/O to the Devices in the SRDF Group		04/02/2009	Pass		
Generate I/O the SRDF R1 devices in test1cg		04/02/2009	Pass		
Watch the DSE information in query output and insure that it is using a significant amount of cache, but not spilling over. Adjust the workload as necessary.	symrdf -og test1cg query -rdfa -i 5 -hop2	04/02/2009	Pass		
Create a second workload simulating a burst or bursts of I/O activity		04/02/2009	Pass		Blocked a port between Site B and Site C instead as easier to force an overflow.
Monitor SRDF and Spillover Devices		04/02/2009	Pass		
Monitor test1cg for tracks spilling over on both sides of the link.	symrdf -og test1cg query -rdfa -i 10 -hop2	04/02/2009	Pass		
Check the total percent full of the SAVE devices in the DSE pool	symrdf -sid <sid> monitor -avp <PoolName> [-i Interval] [-o Count]	04/02/2009	Pass		DSE pool usage topped out as with only 2 IUNs per pool
When the burst activity stops, monitor until SRDF/A recovers and is no longer spilling over.	symrdf -og <cg_name> query -hop2 -rdfa symrdf -sid <sid> monitor -avp <PoolName> [-i Interval] [-o Count]	04/02/2009	Pass		
From Site A Control Server - Add remote BCVs from Site C to test1cg and Capture a Gold Copy					
If not done previously, on all Site A, Site B, and Site C Control Servers, set the SYMCLL_CLONE_EMULATION Variable	set SYMCLL_CLONE_EMULATION=ENABLED	05/02/2009	Pass		SYMCLL_CLONE_EMULATION=ENABLED export SYMCLL_CLONE_EMULATION
Add 4 4way Meta remote RBCV devices from Site C Symm to test1cg	symbcv -cg test1cg associateall dev -range <start-end> -sid <Site A SID> -rdg 11 -remote_rdg 12 -hop2 -nop	05/02/2009	Pass		symcg show test1cg shows devices added as Hop2 BCV devices with logical names
List the Meta BCV devices present in the Site C Symmetrix	symdev list -sid <Site C SID> -meta -bcv -noport	05/02/2009	Pass		Simply showed the 8 BCVs on Summ 3110
Add 6 2-way Meta (6 member) RBCV devices to RDG 35 of test1cg	symbcv -cg test1cg associateall dev -range <start-end> -sid <Site A SID> -rdg 35 -remote_rdg 45 -hop2 -nop	N/A	Not Performed		Does not fit our configuration
Re-examine test1cg	symcg show test1cg	N/A	Not Performed		Does not fit our configuration
Add 6 RAID6 Meta (6 member) remote RBCV devices from Site C Symm to test1cg		N/A	Not Performed		Does not fit our configuration
List the RAID6 Meta BCV devices present in the Site C Symmetrix	symdev list -sid <Site C SID> -meta -bcv -noport -raid6	N/A	Not Performed		Does not fit our configuration
Add 6 RAID6 Meta (6 member) RBCV devices to test1cg	symbcv -cg test1cg associateall dev -range <start-end> -sid <Site A SID> -rdg 39 -remote_rdg 46 -hop2 -nop	N/A	Not Performed		Does not fit our configuration
Re-examine test1cg	symcg show test1cg	N/A	Not Performed		Does not fit our configuration
Perform a full establish on the Site C R2 / RBCV pairs for test1cg	symmir -cg test1cg establish -full -rdg all -hop2 -nop	05/02/2009	Pass		
Monitor the Timefinder full establish for all device groups	symmir -cg test1cg query -rdg all -hop2 -i 5	05/02/2009	Pass		All BCVs observed to be syncing
Split the Site C R2/RBCV pairs using the -consistent option for each device group once the pairs are establish	symmir -cg test1cg split -consistent -rdg all -hop2 -nop	05/02/2009	Pass		
Monitor the Timefinder split for test1cg	symmir -cg test1cg query -i 5 -rdg all -hop2 -bg -percent	05/02/2009	Pass		Split happened so fast that 5 sec monitoring loop missed any <100% status of split
Once the R2/RBCV pairs are split mount the RBCV for device groups test1dg and test2dg to Test Host C		N/A	Not Performed		We do not have the consistency benchmark checking scripts
Using consistency benchmark checking scripts - verify that the image on RBCV devices is consistent		N/A	Not Performed		We do not have the consistency benchmark checking scripts
Once consistency is verified, re-establish R2/RBCV pairs on Site C	symmir -cg test1cg establish -rdg all -hop2 -nop	05/02/2009	Pass		
Perform an MSC Trip and Recovery for test1cg. Examine the SRDF/A Timestamp					
Perform a RDF Suspend with the force flag to trip MSC. Use original beta11 device group and rdg 12	symrdf -g beta11 suspend -immediate -hop2 -rdg 12 -force -nop	05/02/2009	Pass		Suspend initiated at 03:43:48
Examine the RDF State and timestamps of test1cg	symrdf -cg test1cg query -hop2 -detail -i 5	05/02/2009	Pass		
Verify that the RDF Pair State for both hop2 RDFGs in test1cg is "Suspended"		05/02/2009	Pass		Both groups went into suspended mode
Verify that the "time lag" of the R2 displayed by Time that R2 is behind R1 is increasing		05/02/2009	Pass		"Time behind" steadily increasing
Verify that the timestamp of the R2 data displayed by 'R2 Image Capture Time' is staying constant		05/02/2009	Pass		R2 Timestamp was 03:43:39

8.3 Cascaded SRDF with Star Technologies – Base testing

Cascaded Star



Page 46 of 51

8.4 Additional Citi Defined Testing

Return to TOC						
Customer Defined Tests						
Step	Task	Command / Outputs	Date Performed	Pass/Fail	OPT Number	Comments / Results
1.1	movepair stress test					
1.1.1	Move all test devices into RDFG 12	symrdf -g beta14 movepair -new_rdfg 12 -nop symrdf -g beta15 movepair -new_rdfg 12 -nop symrdf -g beta16 movepair -new_rdfg 12 -nop symrdf -g beta17 movepair -new_rdfg 12 -nop symrdf -f pairfile_beta_all -rdfg 12 -sid 717 movepair -new_rdfg 14 -nop symrdf -f pairfile_beta_all -rdfg 14 -sid 717 movepair -new_rdfg 15 -nop symrdf -f pairfile_beta_all -rdfg 15 -sid 717 movepair -new_rdfg 16 -nop symrdf -f pairfile_beta_all -rdfg 16 -sid 717 movepair -new_rdfg 17 -nop symrdf -f pairfile_beta_all -rdfg 17 -sid 717 movepair -new_rdfg 12 -nop	01/02/2008	Pass		
1.1.2	Move all devices between RDFGs		01/02/2008	Pass		
1.1.3	Repeat above test 32 times		01/02/2008	Pass		
1.1.4	Move devices back to original groups		01/02/2008	Pass		
1.2	RDF mode check			Not Performed		
1.2.1	Move a device into an empty group	symrdf -g beta14 movepair -new_rdfg 12 -nop	30/01/2008	Pass		
1.2.2	Check RDF mode		30/01/2008	Pass		Mode is ACP_DISK
1.2.3	Move a device into an existing async group	symrdf -g beta14 movepair -new_rdfg 15 -nop	30/01/2008	Pass		Target async group must be in suspended state New device added moves into async suspended state Show information about both RDFG 15 (which devices are now a member of) and RDFG 14 (which the remote BGV is defined in the device group as being down)
1.2.4	Check remote mirror can still be managed	symrdf show beta14	30/01/2008	Partial Pass	PR-0012	Command failed with an internal error
1.2.5	Check if remote mirror can be controlled	symrdf -g beta14 query -rdf	30/01/2008	Failure	PR-0012	
1.2.6	Reset configuration	symrdf -g beta14 movepair -new_rdfg 14 -nop	30/01/2008	Pass		
1.3	Move pair during an establish					
1.3.1	Start establish	symrdf -g beta14 establish -full	31/01/2008	Pass		
1.3.2	Force a suspension	symrdf -g beta14 suspend -symforce -force	31/01/2008	Pass		
1.3.3	Check invalids	symrdf list -sid 717	31/01/2008	Pass		R2 Invalid Tracks 228607 on RDFG14
1.3.4	Move device to RDFG 15	symrdf -g beta14 movepair -new_rdfg 15	31/01/2008	Pass		
1.3.5	Check invalids	symrdf list -sid 717	31/01/2008	Pass		R2 Invalid Tracks 228607 on RDFG15
1.3.6	Resume establish	symrdf -f log15 -sid 717 resume -rdfg 15	31/01/2008	Pass		Mode is async (as RDFG 15 was already async)
1.4	Change minimum cycle time to 1 sec					
1.4.1	Set minimum cycle time of 2 groups to 1 sec	symconfigure -f change_min_cycle_time.cmd -sid 717 commit - set ra group 14, minimum_cycle_time=1; - set ra group 15, minimum_cycle_time=1;	01/02/2008	Pass		
1.4.2	Start I/O		01/02/2008	Pass		The actual cycle time varies from 1 sec (no I/O) to 4 sec (moderate I/O) and 6-11 sec for high I/O i.e. cycle time limited by data transfer time
1.4.3	Monitor at cycle time	sysstat -type cycle -rdtype rdta -sid 717 -t 1 symconfigure -f change_min_cycle_time.cmd -sid 717 commit - set ra group 14, minimum_cycle_time=30; - set ra group 15, minimum_cycle_time=30;	01/02/2008	Failure	PR-0029	Although minimum cycle switching can now be set to 1 sec, it is not possible to monitor more frequently than 5 sec
1.4.4	Reset cycle times		01/02/2008	Pass		
2.1	Set up Cascaded SRDF					
2.1.1	Create RDFgroups	# Create dynamic RDF group pairs # None of disks involved in this configuration to be in any RDF group at start of this process # symdev list -sid 919 -dynamic symdev list -sid 717 -dynamic symdev list -sid 3110 -dynamic # symrdf addgrp -label beta11 -sid 919 -rdfg 11 -dr 90 -remote_sid 717 -remote_rdfg 11 -remote_dir 160 -nop symrdf addgrp -label beta10 -sid 919 -rdfg 10 -dr 90 -remote_sid 717 -remote_rdfg 10 -remote_dir 160 -nop	04/01/2008	Pass		Alternative setup needed to test plan as we had to use Sites B and C as Sites A and B for the movepair testing, as the final SiteA Symm is on 5772 and this does not support movepair.
2.1.2	Create pairs	# beta10: 10 -> 10/14 -> 14 # symrdf createpair -f pairfile_casc_919_717_beta10 -sid 919 -rdfg 10 -type R1 -rdf_mode acc_disk -establish -g beta10 -nop symrdf createpair -f pairfile_casc_717_3110_beta14 -sid 717 -rdfg 14 -type R1 -rdf_mode acc_disk -establish -nop symrdf -g beta10 query symrdf -g beta10 query -hop2 # # beta11: 11 -> 11/12 -> 12 # symrdf createpair -f pairfile_casc_919_717_beta11 -sid 919 -rdfg 11 -type R1 -rdf_mode acc_disk -establish -g beta11 -nop symrdf createpair -f pairfile_casc_717_3110_beta12 -sid 717 -rdfg 12 -type R1 -rdf_mode acc_disk -establish -nop # symrdf -g beta11 query symrdf -g beta11 query -hop2	04/01/2008	Pass		
2.1.3	Ensure DSE and TI are set up on 2nd leg	symrdf list -rdf all -sid 717 -rdta symrdf list -pool -rdta -sid 717 -rdta symrdf ds2cp beta10 test1cp -rdf consistency symrdf ds2cp beta11 test1cp -rdf consistency	04/01/2008	Pass		
2.1.4	Create consistency group	symrdf -g beta10 set mode sync -nop symrdf -g beta11 set mode sync -nop	04/01/2008	Pass		
2.1.5	Set modes ready to join main test plan		04/01/2008	Pass		

Cascaded swap testing (A->B->C to C->B->A) controlled from Site A						Script casc_planned_swap_and_swapback (Run from Site A)
2.2						
2.2.1	Suspend both legs	symrdf -g betall suspend -nop symrdf -g betall suspend -hop2 -nop	11/02/2008	Pass		
2.2.2	set 919-717 hop to acp_disk ready for swap as hop2 cannot be sync	symrdf -g betall set_mode acp_disk -nop	11/02/2008	Pass		
2.2.3	Falover and swap hop1	symrdf -g betall fallover -force -nop symrdf -g betall swap -nop symrdf -f pairsfile_casc_717_9110_beta12 -aid 717 fallover -rdofg 12 -force -nop symrdf -f pairsfile_casc_717_9110_beta12 -aid 717 swap -rdofg 12 -nop	11/02/2008	Partial Pass	PR-0045	Have to fallover and swap hop1 first to avoid a R1 --> R22 <-- R1 config. This will instead give us R2 <-- R11 --> R2, which is legal Why is -force flag needed
2.2.4	Falover and swap hop2	symrdf -g betall fallover -force -nop symrdf -g betall swap -nop symrdf -f pairsfile_casc_717_9110_beta12 -aid 717 swap -rdofg 12 -nop	11/02/2008	Pass		Cannot use symrdf -g <DG> -hop2 as no longer a cascaded configuration (R2 <-- R11 --> R2)
2.2.5	Establish both legs	symrdf -g betall establish -nop symrdf -g betall establish -hop2 -nop symrdf -g betall suspend -nop symrdf -g betall suspend -hop2 -nop symrdf -g betall fallover -hop2 -nop symrdf -f pairsfile_casc_717_9110_beta12 -aid 717 swap -rdofg 12 -nop symrdf -g betall fallover -force -nop symrdf -g betall set_mode sync -nop symrdf -g betall swap -nop symrdf -g betall establish -nop symrdf -g betall establish -hop2 -nop symrdf -g betall query; symrdf -g	11/02/2008	Pass		Note that in testing it was found that if resume -force was used, then RDF pair state became invalid
2.2.6	Swap everything back	symrdf -g betall query; symrdf -g	11/02/2008	Partial Pass	PR-0045	Why is -force flag needed for fallover?
2.3	Cascade: Fast fallover at Site C (Site A up)					
2.3.1	Check if GNS remote mirroring works for Cascaded configurations	SiteA symdg show beta11	12/02/2008	Partial Pass	N/A	Docs state that RDF1 and RDF2 DGs and CGs with STD devices participating in a Cascaded RDF relationship will not be mirrored
2.3.2	Create DG at Site C	SiteC symdg import -f beta11.dg beta11	12/02/2008	Pass		
2.3.3	Suspend Sync leg from Site C	SiteC symrdf -g beta11 suspend -hop2	12/02/2008	Pass		
2.3.4	Suspend Async leg from Site C	SiteC symrdf -g beta11 suspend	12/02/2008	Pass		
2.3.5	Falover Async leg from Site C	SiteC symrdf -g beta11 fallover -force -nop	12/02/2008	Partial Pass	PR-0045	Why is -force flag needed
2.3.6				Not Performed		
2.4	Cascade: Fast fallover at Site C (Site A down)					
2.4.1	Block RA on Symm at Site A used for Sync replication	SiteC symrdf -g beta11 query -hop2	12/02/2008	Pass		Leg is PARTITIONED
2.4.2	Check status of Sync leg from Site C	SiteC symrdf -g beta11 query -hop2	12/02/2008	Pass		This compound command gracefully suspends SRDF/A replication. Fast fallover compared with Star (Run 1: 63 sec; Run 2: 59 sec)
2.4.3	Falover Async leg from Site C	SiteC symrdf -g beta11 fallover	12/02/2008	Pass		Leg now suspended
2.4.4	Unblock RA on Symm at Site A used for Sync replication	SiteC symrdf -g beta11 query -hop2	12/02/2008	Pass		Consistent again
2.4.5	Check status of Sync leg from Site C	SiteC symrdf -g beta11 query -hop2	12/02/2008	Pass		Synchronized again
2.4.6	Failback Async leg from Site C	SiteC symrdf -g beta11 failback	12/02/2008	Pass		
2.4.7	Resume the sync leg	SiteC symrdf -g beta11 resume -hop2	12/02/2008	Pass		
2.5	Cascade: Variant with ongoing Replication to Site B when Site A lost					
2.5.1				Not Performed		
	Variant with ongoing Replication to Site B when Site A lost					
	Site C Script used: casc_unplanned_to_sitec_with_C_to_B					This script does a fallover to Site C for an unplanned fallover (assume Site A not available)
2.5.1	Block RA on Symm at Site A used for Sync replication	N/A	21/02/2008	Pass		
2.5.2	Check status of Sync leg from Site C	SiteC symrdf -g beta11 query -hop2	21/02/2008	Pass		Leg is PARTITIONED
	Suspend Async leg	SiteC symrdf -g beta11 suspend	21/02/2008	Pass		
	Falover Async leg from Site C	SiteC symrdf -g beta11 fallover -force	21/02/2008	Pass		
	Half-swap partitioned A -> B link at Site B	SiteC symrdf -g beta11 half_swap -hop2 -nop	21/02/2008	Pass		Have to make the Site B half of the pair an R1 so that I can change the mode of the link. This can only be done for an R1 We need to get the A - B link out of sync else the B - C swap will not work as far hop must not be sync
	Change A - B link to acp_disk (or B end at least)	SiteC symrdf -f pairsfile_casc_717_919_beta11 -aid 717 -rdofg 11 set_mode acp_disk -nop	21/02/2008	Pass		As the configuration is not no longer concurrent, a pairsfile has to be used The device is not in a valid RDF state for this operation This is blocked as it would cause a R1 --> R22 <-- R1 configuration
	Swap Async leg from Site C	SiteC symrdf -g beta11 swap	21/02/2008	Pass	Not yet logged	Do I need to do a half-swap on the sync leg?
	Establish Async leg (C-> B) from Site C	SiteC symrdf -g beta11 establish	21/02/2008	Pass		
	Query B - C link	SiteC symrdf -g beta11 query	21/02/2008	Pass		In "consistent" state
	Site C Script used: casc_unplanned_to_sitec_with_C_to_B backout					
	Unblock RA on Symm at Site A used for Sync replication	SiteC symrdf -g beta11 query -hop2	21/02/2008	Pass		#this script does REVERSE of the script to fallover to Site C for an unplanned fallover (assume Site A not available)
	Check status of Sync leg from Site C	SiteC symrdf -g beta11 query -hop2	21/02/2008	Pass		
	Suspend Async leg	SiteC symrdf -g beta11 suspend	21/02/2008	Pass		
	Failback Async leg from Site C	SiteC symrdf -g beta11 fallover	21/02/2008	Pass		
	Swap Async leg from Site C	SiteC symrdf -g beta11 swap	21/02/2008	Pass		
	Establish Async leg (B-> C) from Site C	SiteC symrdf -g beta11 establish	21/02/2008	Pass		
	Check status of Async leg	SiteC symrdf -g beta11 query	21/02/2008	Pass		
	Execute half swap at Site B on A - B link	SiteC symrdf -f pairsfile_casc_717_919_beta11 -aid 717 -rdofg 11 half_swap -nop	21/02/2008	Pass		Note that this half swap (from R1 to R2) seems to set the mode back to sync for the Site B end of things
	Check sync leg no longer partitioned by instead suspended	SiteC symrdf -g beta11 query -hop2	21/02/2008	Pass		
	Resume sync leg	SiteC symrdf -g beta11 resume -hop2 -nop	21/02/2008	Pass		
	Check status of sync leg	SiteC symrdf -g beta11 query -hop2	21/02/2008	Pass		

2.6 Cascade: use of symmrecover in a cascaded environment						
2.6.1	Enable test1cg	symmrc -cg test1cg enable	12/02/2008	Pass		
2.6.2	Check hop2 BCVs are established	symmrc -cg test1cg query -rdg all -hop2 -i 5	12/02/2008	Pass		
2.6.3	Suspend hop2	symmrc -cg test1cg suspend -hop2	12/02/2008	Pass		
2.6.4	Use symmrecover to take PIT copy and resync async leg	symmrecover start -cg test1cg -mode async -options options	12/02/2008	Failure	PR-0050	Error: Specified group name [test1cg] with [4] standard devices only contains [0] [R2] BCV devices
2.9 Cascaded swap testing (A->B->C to C->B->A) controlled from Site C						Script casc_planned_swap_from_sitec (Run from Site C)
2.9.1	Suspend Sync leg	symmrcf -g betall suspend -hop2 -nop	12/02/2008	Pass		
2.9.2	Suspend Async leg	symmrcf -g betall suspend -nop	12/02/2008	Pass		
2.9.3	Failover sync leg	symmrcf -g betall failover -hop2 -nop	12/02/2008	Pass		
2.9.4	Failover async leg	symmrcf -g betall failover -force -nop	12/02/2008	Pass		
2.9.5	Set sync leg to acp disk prior to swap	symmrcf -g betall set mode acp_disk -hop2 -nop	12/02/2008	Pass		
2.9.6	Swap sync leg	symmrcf -g betall swap -hop2 -nop	12/02/2008	Pass		
2.9.7	Swap async leg	symmrcf -g betall swap -nop	12/02/2008	Pass		
2.9.8	Establish former sync leg (now acp disk)	symmrcf -g betall establish -hop2 -nop	12/02/2008	Pass		
2.9.9	Establish async leg	symmrcf -g betall establish -nop	12/02/2008	Pass		
2.9.10	Query former sync leg	symmrcf -g betall query -hop2	12/02/2008	Pass		
2.9.11	Query async leg	symmrcf -g betall query	12/02/2008	Pass		
2.10 Cascaded swap testing (C->B->A to A->B->C) controlled from Site A						Script casc_planned_swapback_from_sitea (Run from Site A)
2.10.1	Suspend Async leg	symmrcf -g betall suspend -hop2 -nop	12/02/2008 & 19/02/2008	Pass		
2.10.2	Suspend Sync leg	symmrcf -g betall suspend -nop	12/02/2008 & 19/02/2008	Pass		
2.10.3	Failover async leg	symmrcf -g betall failover -hop2 -nop	12/02/2008 & 19/02/2008	Pass		
2.10.4	Failover sync leg	symmrcf -g betall failover -force -nop	12/02/2008 & 19/02/2008	Pass		
2.10.5	Swap async leg	sleep 10	12/02/2008 & 19/02/2008	Pass		
2.10.6	Swap sync leg	symmrcf -g betall swap -nop	12/02/2008 & 19/02/2008	Pass	PR-0071	Issue fixed with later code drop
2.10.7	Set sync leg back to sync again	symmrcf -g betall set mode sync -nop	12/02/2008 & 19/02/2008	Pass		
2.10.8	Establish async leg	symmrcf -g betall establish -hop2 -nop	12/02/2008 & 19/02/2008	Pass		
2.10.9	Establish former async leg	symmrcf -g betall establish -nop	12/02/2008 & 19/02/2008	Pass		
2.10.10	Query sync leg	symmrcf -g betall query	12/02/2008 & 19/02/2008	Pass		
2.10.11	Query async leg	symmrcf -g betall query -hop2	12/02/2008 & 19/02/2008	Pass		
2.10 Cascaded: Adding/removing disks (non-movepair)						
To create free disks move into the "beta11" cascaded SRDF setup they must first be removed from the "beta10" cascaded setup. Note that cg test1cg covers all disks i.e. beta10+beta11 Scripts are "remove_from_beta10_to_beta11" and "reverse_remove"			19/02/2008	Pass		
2.10.1	Suspend all legs in both cascaded setups	symmrcf -g beta10 suspend -nop symmrcf -g beta10 suspend -hop2 -nop symmrcf -g beta11 suspend -nop symmrcf -g beta11 suspend -hop2 -nop	19/02/2008	Pass		
2.10.2	Set hop2 to async mode as I could not add disk when suspended in async mode	symmrcf -g beta11 set mode acp_disk -hop2 -nop	19/02/2008	Pass		
2.10.3	Delete "beta10" pairs	symmrcf -f pairsfile_casc_919_717_beta10 -aid 919 -rdg 10 deletepair -nop symmrcf -f pairsfile_casc_717_3110_beta14 -aid 717 -rdg 14 deletepair -nop	19/02/2008	Pass		
2.10.4	Execute createpair to add former beta10 disks to beta11 cascade	symmrcf createpair -f pairsfile_casc_919_717_beta10 -aid 919 -rdg 11 -type R1 -establish -nop symmrcf createpair -f pairsfile_casc_717_3110_beta14 -aid 717 -rdg 12 -type R1 -rdi mode acp_disk -establish -nop	19/02/2008	Pass		
2.10.5	Establish all disks, both links	symmrcf -cg test1cg establish -nop symmrcf -cg test1cg establish -hop2 -nop	19/02/2008	Pass		
2.10.6	Set hop2 back into async mode Should wait until copy finished for performance reasons	symmrcf -cg test1cg verify -synchronized -i 10 symmrcf -cg test1cg set mode async -hop2 -nop	19/02/2008	Pass		
2.10.7	Check mode and pair status all correct	symmrcf -cg test1cg query -nop symmrcf -cg test1cg query -hop2 -nop	19/02/2008	Pass		
2.10.8 REVERSE THE ABOVE CHANGE						
2.10.9	Suspend both legs	symmrcf -cg test1cg suspend -nop symmrcf -cg test1cg suspend -hop2 -nop	19/02/2008	Pass		
2.10.10	Delete pairs	symmrcf -f pairsfile_casc_919_717_beta10 -aid 919 -rdg 11 deletepair -nop symmrcf -f pairsfile_casc_717_3110_beta14 -aid 717 -rdg 12 deletepair -nop	19/02/2008	Pass		
2.10.11	createpairs	symmrcf createpair -f pairsfile_casc_919_717_beta10 -aid 919 -rdg 10 -type R1 -establish -nop symmrcf createpair -f pairsfile_casc_717_3110_beta14 -aid 717 -rdg 14 -type R1 -rdi mode acp_disk -establish -nop	19/02/2008	Pass		
2.10.12	Establish all disks, both links	symmrcf -g beta10 establish -nop symmrcf -g beta10 establish -hop2 -nop symmrcf -g beta11 establish -nop symmrcf -g beta11 establish -hop2 -nop	19/02/2008	Pass		
2.10.13	Set hop2 back into async mode Should wait until copy finished for performance reasons	symmrcf -g beta10 verify -synchronized -i 10 symmrcf -g beta10 verify -synchronized -hop2 -i 10 symmrcf -g beta10 set mode async -hop2 -nop	19/02/2008	Pass		
2.10.14	Check mode and pair status all correct	symmrcf -g beta10 query -nop symmrcf -g beta10 query -hop2 -nop symmrcf -g beta11 query -nop symmrcf -g beta11 query -hop2 -nop	19/02/2008	Pass		

2.12	Cascaded SRDF - Disabling Star					
2.12.1	Disable star	symstar -cg nova disable -nop	25/02/2008	Pass		The unprotect order is important (unlike with concurrent star) New Jersey cannot be unprotected until Dallas has been unprotected
2.12.2	Unprotect Site C first	symstar -cg nova unprotect -site Dallas -nop	25/02/2008	Pass		
2.12.3	Unprotect Site B	symstar -cg nova unprotect -site NewJersey -nop	25/02/2008	Pass		
2.12.4	Disconnect site B	symstar -cg nova disconnect -site NewJersey -nop	25/02/2008	Pass		
2.12.5	Disconnect Site C	symstar -cg nova disconnect -site Dallas -nop	25/02/2008	Pass		
2.12.6	Confirm status	symstar -cg nova query	25/02/2008	Pass		
2.13	Cascaded Star - Planned A->B->C to C->B (and return)					
2.13.1	The following results in Async replication to Site B		25/02/2008	Pass		
2.13.2	Switch to Site C with Async replication to Site B	SiteC# symstar -cg nova switch -site Dallas -nop 10 sec SiteC# symstar -cg nova connect -site NewJersey -nop 11 sec SiteC# symstar -cg nova connect -site Manhattan -nop 14 sec SiteC# symstar -cg nova protect -site NewJersey -nop 60 sec	25/02/2008	Pass		Note that one does not protect Manhattan in this configuration. ACP_DISK replication from NewJersey to Manhattan will be taking place to stop Manhattan from getting too far behind. Sync replication on hop2 is not supported. Dallas Site C — Async -> NewJersey Site B — acp_disk -> Manhattan Site A
2.13.3	Switch back to Site A	SiteA# symstar -cg nova halt -nop SiteA# symstar -cg nova switch -site Manhattan -nop SiteA# symstar -cg nova connect -site NewJersey -nop SiteA# symstar -cg nova connect -site Dallas -nop SiteA# symstar -cg nova protect -site NewJersey -nop SiteA# symstar -cg nova protect -site Dallas -nop SiteA# symstar -cg nova enable -nop SiteA# symstar -cg nova query	25/02/2008	Pass		
2.14	Cascaded Star - Planned reconfiguration from cascaded to concurrent (and back) via halt					
2.14.1		SiteA# symstar -cg nova halt -nop SiteA# symstar -cg nova reconfigure -site Dallas -path Manhattan.Dallas -nop SiteA# symstar -cg nova halt -reset -nop SiteA# symstar -cg nova connect -site NewJersey -nop SiteA# symstar -cg nova connect -site Dallas -nop SiteA# symstar -cg nova protect -site NewJersey -nop SiteA# symstar -cg nova protect -site Dallas -nop SiteA# symstar -cg nova enable -nop SiteA# symstar -cg nova query	25/02/2008	Pass		Initial config Manhattan Site A — Sync -> NewJersey Site B - Async -> Dallas Site C Final config Manhattan Site A — Sync -> NewJersey Site B Manhattan Site A — Async -> Dallas Site C
2.14.2	Reconfigure from Cascaded to concurrent	SiteA# symstar -cg nova halt -nop SiteA# symstar -cg nova reconfigure -site Dallas -path NewJersey.Dallas -nop SiteA# symstar -cg nova halt -reset -nop SiteA# symstar -cg nova connect -site NewJersey -nop SiteA# symstar -cg nova connect -site Dallas -nop SiteA# symstar -cg nova protect -site NewJersey -nop SiteA# symstar -cg nova protect -site Dallas -nop SiteA# symstar -cg nova enable -nop SiteA# symstar -cg nova query	25/02/2008	Pass		Initial config Manhattan Site A — Sync -> NewJersey Site B Manhattan Site A — Async -> Dallas Site C Final config Manhattan Site A — Sync -> NewJersey Site B — Async -> Dallas Site C
2.14.3	Reconfigure from Concurrent to cascaded	SiteA# symstar -cg nova halt -nop SiteA# symstar -cg nova reconfigure -site Dallas -path NewJersey.Dallas -nop SiteA# symstar -cg nova halt -reset -nop SiteA# symstar -cg nova connect -site NewJersey -nop SiteA# symstar -cg nova connect -site Dallas -nop SiteA# symstar -cg nova protect -site NewJersey -nop SiteA# symstar -cg nova protect -site Dallas -nop SiteA# symstar -cg nova enable -nop SiteA# symstar -cg nova query	25/02/2008	Pass		
2.15	Cascaded Star - Planned ABC to CA (and return)					
2.15.1		SiteA# symstar -cg nova halt -nop	03/03/2008	Pass		It is essential to do the reconfigure before the switch to avoid a full resync later in the process.
2.15.2		SiteA# symstar -cg nova reconfigure -site Dallas -path Manhattan.Dallas -nop	03/03/2008	Pass		
2.15.3		SiteC# symstar -cg nova switch -site Dallas	03/03/2008	Pass		
2.15.4		SiteC# symstar -cg nova connect -site Manhattan	03/03/2008	Pass		
2.15.5		SiteC# symstar -cg nova connect -site NewJersey	03/03/2008	Pass		
2.15.6		SiteC# symstar -cg nova Manhattan	03/03/2008	Pass		
2.15.7						
2.15.8		SiteC# symstar -cg nova halt -nop	03/03/2008	Pass		
2.15.9		SiteC# symstar -cg nova reconfigure -site Manhattan -path NewJersey.Manhattan	03/03/2008	Pass		
2.15.10		SiteA# symstar -cg nova switch -site Manhattan	03/03/2008	Pass		
2.15.11		SiteA# symstar -cg nova connect -site NewJersey	03/03/2008	Pass		
2.15.12		SiteA# symstar -cg nova connect -site Dallas	03/03/2008	Pass		
2.15.13		SiteA# symstar -cg nova protect -site NewJersey	03/03/2008	Pass		
2.15.14		SiteA# symstar -cg nova protect -site Dallas	03/03/2008 17/03/2008	Pass	PR-0087	

Fails to protect, unless I remove Star configuration, delete CG, create CG and rebuild stuff
Fixed with the 12/03/08 code drop

2.16	Cascaded Star - Planned reconfiguration WITHOUT symstar halt Cascaded -> Concurrent				
2.16.1	Disconnect Site C	SiteA# symstar -cg nova disconnect -trip -site Dallas -nop	07/03/2008	Pass	39 s
2.16.2	Reconfigure to concurrent	SiteA# symstar -cg nova reconfigure -reset -site Dallas -path Manhattan Dallas -nop	07/03/2008	Pass	58s -reset is needed to transition the site state from PathFail to Disconnected prior to starting the Reconfiguration operation
2.16.3	Connect Site C again	SiteA# symstar -cg nova connect -site Dallas -nop	07/03/2008	Pass	10s
2.16.4	Protect Site C again	SiteA# symstar -cg nova protect -site Dallas -nop	07/03/2008	Pass	1m 21s
2.16.5	Enable Star	SiteA# symstar -cg nova enable -nop	07/03/2008	Pass	2m 21s
	Concurrent -> Cascaded				
2.16.6	Disconnect Site C	SiteA# symstar -cg nova disconnect -trip -site Dallas -nop	07/03/2008	Pass	39s
2.16.7	Reconfigure to concurrent	SiteA# symstar -cg nova reconfigure -reset -site Dallas -path NewJersey Dallas -nop	07/03/2008	Pass	47s
2.16.8	Connect Site C again	SiteA# symstar -cg nova connect -site Dallas -nop	07/03/2008	Pass	8s
2.16.9	Protect Site C again	SiteA# symstar -cg nova protect -site Dallas -nop	07/03/2008	Pass	2m 12s
2.16.10	Enable Star	SiteA# symstar -cg nova enable -nop	07/03/2008	Pass	2m 10s
2.17	Cascaded Star - Unplanned ABC to CB (Site A down)				
2.17.1	Block ports to isolate Site A	SiteC# symstar -cg nova disconnect -trip -site Dallas -nop	17/03/2008	Pass	
2.17.2	Bring up at Site C, with C-B replication	SiteC# symstar -cg nova clean -site Dallas -nop	17/03/2008	Pass	
2.17.3		SiteC# symstar -cg nova switch -site Dallas -keep NewJersey -nop	17/03/2008	Pass	
2.17.4		SiteC# symstar -cg nova protect -site NewJersey -nop	17/03/2008	Pass	
2.17.5		SiteC# symstar -cg nova enable -nop	17/03/2008	Pass	
2.17.6	Unblock ports to make Site A accessible again	SiteC# symstar -cg nova halt -nop	17/03/2008	Pass	
2.17.7	Halt and move production back to Site A	SiteA# symstar -cg nova connect -site NewJersey -nop	17/03/2008	Pass	
2.17.8		SiteA# symstar -cg nova connect -site Dallas -nop	17/03/2008	Pass	
2.17.9		SiteA# symstar -cg nova protect -site Dallas -nop	17/03/2008	Pass	
2.17.10		SiteA# symstar -cg nova enable -nop	17/03/2008	Pass	
2.17.11		SiteA# symstar -cg nova query	17/03/2008	Pass	
2.17.12					
2.17.13					
2.17.14					