VMware, Inc.
World Headquarters
3401 Hillview Ave, Palo Alto, CA 94304
Tel: 1-877-486-9273
http://www.vmware.com/

Storage Replication Adapter 2.0

Interface Specification

VMware Confidential

Author: Maria Basmanova
Last Updated Date: 3/24/2015

Revision History

Date	Change Description
07/14/2009	• renamed discoverLuns to discoverDevices; • renamed Lun to Device throughout the document; • added prepareFailover command called during planned migration to make production devices read-only; • added HelpUrl tag to queryInfo response;
07/21/2009	• added checkTestFailoverStart and checkFailover commands to check for test or failover readiness;
07/23/2009	• simplified response for queryCapabilities command;
07/27/2009	• simplified response for queryErrorDefinitions command;
07/29/2009	• user-friendly names for arrays, consistency groups, and devices in discoverArrays and discoverDevices response;
07/30/2009	• modified queryRecoveryPoints response;
08/07/2009	• cleaned up discoverDevices response; • introduced TargetGroup tag to distinguish replication targets of consistency groups;
08/21/2009	• added support for replication software capable of taking quiesced snapshot of VMs before replication; affected commands: queryCapabilities, queryRecoveryPoints, testFailoverStart, failover;
08/21/2009	• replaced querySyncSchedule with more genertic queryReplicationSettings command which convers replication schedule as well as settings for VM snapshots taken before replication;
09/15/2009	• added datastore MoId as a device identity to allow SRA to report datastores in addition to raw storage devices (this could be useful to secondary storage replication solutions);
09/17/2009	• added more details to queryRecoveryPoints command; • clarified the requirement for failover to keep a copy of pre-failover replica data to ensure robust revert experience;
10/19/2009	• added list of supported locales to queryCapabilities command;
12/4/2009	• outlined specific limitations of the Cancun release of SRM;
1/14/2010	• added support for VmImageConsistency feature back in;
7/22/2010	• extended queryCapabilities response to add OneLunPerIScsiTarget flag indicating that underlying storage creates separate iSCSI target for each LUN;
10/19/2010	• described a scenario where prepareFailover command is executed after failover command; • fixed example of a pre-defined error with runtime parameters to add custom namespace to tags describing runtime parameters; • clarified that prepareFailover, failover, prepareReverseReplication, reverseReplication; prepareRestoreReplication, and restoreReplication commands must be idempotent;

	• changed Cancun to Glenlivet;
02/15/2011	• clarified Discovery of Replicated Storage workflow; • clarified the difference between Planned Migration and Disaster Recovery mode of Failover workflow; • outlined limitations of the Glenlivet release of SRM with regards to checkTestFailoverStart and checkFailover commands;
03/23/2011	• clarified VmImageConsistency and OneLunPerTarget SRA capabilities; • clarified that all devices in a consistency group must have the same type which should match the type of the consistency group itself; • clarified how to report source, snapshot and promoted devices which are not presented to any host in response to discoverDevices, testFailoverStart and failover commands; • added Installation section which describes how to install an SRA; • added End-to-End Workflows section to illustrate how SRA is used in end-to-end SRM workflows;
11/03/2011	• clarified usage of VmImageConsistency • recommended using queryReplicationSettings to avoid manual setup during failback
11/07/2011	• clarified supported locales
11/28/2011	• added information about moving to 64bit server
2/28/2012	• added notes about post disaster workflows for arrays that perform dynamic personality swap • added logging requirements
4/30/2012	• clarified that testFailoverStart, testFailoverStop commands must be idempotent;
11/28/2012	• fixed example for discoverDevices response • enhanced device discovery support in the middle of several concurrent workflows • recommended adding support for device name filter • removed clarification that testFailoverStart needs to be idempotent (not required for now)
03/26/2014	• require SRA to install all binaries under the designated SRA installation folder
06/03/2014	• require ReverseReplication and RestoreReplication to succeed in case of invalid replication settings

Introduction

This document defines the interface between VMware Site Recovery Manager (SRM) and replicated storage arrays. This interface, called the Storage Replication Adapter (SRA) interface, enables SRM to execute the following workflows:

- discovery of replicated storage;
- non-disruptive failover test using a writable copy of replicated data;
- emergency or planned failover;
- reverse replication after failover as part of failback;
- restore replication after failover as part of a production test.

The interface consists of the following commands.

- Discovery of SRA properties and capabilities:
  - queryInfo - queries the SRA for basic properties such as name and version
  - queryCapabilities - queries the SRA for supported models of storage arrays and supported SRM commands
  - queryErrorDefinitions - queries the SRA for pre-defined array specific errors
  - queryConnectionParameters - queries the SRA for parameters needed to connect to the array management system to perform array management operations
  - queryStrings – queries for strings localized for a given locale
- Discovery of array replication:
  - discoverArrays – discovers storage array pairs configured for replication
  - discoverDevices - discovers replicated devices on a given storage array
  - queryReplicationSettings - queries replication settings for a list of devices
  - queryRecoveryPoints* - queries for recovery points for a list of devices
  - queryDeviceDetails* – queries SRA-specific properties for a list of devices
- Array operations:
  - syncOnce – requests immediate replication
  - querySyncStatus – queries the status of a replication initiated by syncOnce
  - checkTestFailoverStart* – validates environment and target devices for test failover readiness
  - testFailoverStart – creates writable temporary copies of the target devices
  - testFailoverStop – deletes the temporary copies created by testFailoverStart
  - checkFailover* – validates environment and target devices for failover readiness
  - prepareFailover – makes source devices read-only and optionally takes a snapshot of the source devices in anticipation of a failover
  - failover – promotes target devices by stopping replication for those devices and making them writable
  - prepareReverseReplication and reverseReplication - reverses array replication so that the original target array becomes the source array and vice versa
  - prepareRestoreReplication and restoreReplication - restores array replication after failover.

This document defines the requests and responses for each command as an XML schema. The XML request is prepared by SRM and passed to the SRA. The SRA executes the specified command and returns the response as an XML file.

Each SRA must include a Perl script named command.pl. To execute a single command, SRM runs this script, passing the XML request through the standard input, as shown in this example.

```
perl command.pl < {XML request}
```

The contents of the XML request is different for each command. However, all XML requests contain the following elements:

- the command name
- command parameters if required

* Not supported by the Glenlivet release of VMware vCenter Site Recovery Manager

- a local path name where the response will be written
- a local path name where progress messages will be written during command execution
- a log verbosity level
- a local path name where log messages will be written

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>command name</Name>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>log verbosity level</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    ...command parameters (command-specific format)...
</Command>
```

An SRA is required to log messages (in English) to standard output at the specified log level. Log messages must be encoded as UTF-8.

In addition to logging to the standard output, an SRA may create log files in the directory specified by `<LogDirectory>`.

Installation

SRA installation must create a sub-directory under <SRM Install Directory>/storage/sra and place SRA's command.pl script and all other partner-authored bits called by command.pl in that directory. All SRA components containing SRA logic must reside in this directory or in a sub-directory of the SRA installation directory. Any logic components found residing outside the SRA installation directory structure will invalidate any SRA certification results.
<SRM Install Directory> can be retrieved from registry path HKEY_LOCAL_MACHINE\SOFTWARE\VMware, Inc.\VMware vCenter Site Recovery Manager\InstallPath. Version 5.1 of Site Recovery Manager is a 64 bit application. This is a change from SRM 5.0 which is a 32 bit application running on a 64 bit OS.

Localization

SRM supports localization of error messages and strings, such as UI controls and names for connection parameters, that are visible to users. Log messages are required to be in English and cannot be localized.

Supported locales are: en (English), de (German), fr (French), ja (Japanese) and zh-CN (Chinese - China).

The default locale is English. English versions of all localizable strings must be supplied as part of the SRA. Localizable strings in SRA response have the following format:

```
<Xxx stringId="Xxx123">Text in default locale (e.g. English)</Xxx>
```

The text content of the tag is required and must contain the string in English.

`stringId` is an optional argument. If specified, SRM will use it to look up the translation for the current user locale. If `stringId` is not specified SRM will use the text value of the tag.

SRM uses the queryStrings command to retrieve translations for localizable strings.

Security

SRA authors are encouraged to consider the security of usernames and passwords. These should not appear in output logs or error reports. In addition, they should never be written in plaintext to the local file system or to the Windows registry.

Error Reporting

In case of a failure SRA is required to report errors in the XML response like this:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    <Error code="1034"/>
</Response>
```

SRA can report warnings as well like this:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    ...command results...
    <Warnings>
        <Warning code="2001"/>
        <Warning code="2002"/>
    </Warnings>
</Response>
```

To report progress SRA should append XML messages to a file specified in the <StatusFile> element. The content of a progress update message is a <ProgressUpdate> element containing an integer in the range [0, 100]. SRM interprets this integer as percent complete:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ProgressUpdate xmlns="http://www.vmware.com/srm/sra/v2">15</ProgressUpdate>
```

Errors are classified into three categories:

- Pre-defined array-independent errors (defined by SRM)
- Pre-defined array-specific errors (defined by each SRA)
- Other errors (returned by the SRA)

Pre-defined errors support localization. SRM uses queryErrorDefinitions command to get error codes and descriptions in supported locales from the SRA. SRM caches the results of queryErrorDefinitions and uses this cache to translate an actual error returned from the SRA according to the user's locale.

If SRA implements pre-defined error codes it must ensure that all errors returned from all commands use these pre-defined codes.

Pre-defined errors have the following attributes:

- Error code – a unique integer identifying the error.

  Error codes with values less than 1000 are reserved for SRM. Error codes with values 1000 and higher can be defined by an SRA.

- Problem description – user-friendly localizable description of the problem.

- Suggested fix – user-friendly localizable suggestion on how to proceed in order to fix the problem.

Problem description and suggested fix strings may contain placeholders. A placeholder is a name in curly braces. That name defines an XML tag returned along with the error code. SRM will replace placeholder with the contents of the corresponding XML tag at runtime.

Note: Placeholders will be left intact if corresponding XML tag is not present in the actual error.

Example:

Error code: 1001
Problem description: Failed to connect to CLI at {CliAddress}
Suggested fix: Make sure the address {CliAddress} is correct and accessible

Format of the error definition:

```
<ErrorDefinition code="1001">
   <Description stringId="Description1001"
      >Failed to connect to CLI at {CliAddress}</Description>
   <FixHint stringId="FixHint1001"
      >Make sure the address {CliAddress} is correct and
       accessible</FixHint>
</ErrorDefinition>
...
<Strings locale="ja">
   <String id="Description1001">...</String>
   <String id="FixHint1001">...</String>
</Strings>
```

Format of the actual error:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2"
          xmlns:foo="http://www.foo.com/sra/v2">
    …
    <Error code="1001">
        <foo:CliAddress>10.20.30.40</foo:CliAddress>
    </Error>
    …
</Response>
```

Format of the actual error for which there is no pre-defined error:

```
<Error>
    <Description>Failed to connect to CLI at 10.20.30.40</Description>
    <FixHint>Make sure the address 10.20.30.40 is correct and
        accessible</FixHint>
</Error>
```

Unusual conditions that might indicate a problem but don't prevent an operation from succeeding can be reported as warnings. A single operation may produce multiple warnings but only one error. Warnings are defined in the same way as errors, with `<WarningDefinition>` elements grouped within a `<WarningDefinitions>` container element. Warnings are reported with `Warning` tags grouped under `Warnings` container tag.

Warning descriptions and fix hints may contain placeholders.

Example:

Format of a warning definition:

```
<WarningDefinition code="2001">
   <Description stringId="Description2001">...</Description>
   <FixHint stringId="FixHint2001">...</FixHint>
</WarningDefinition>
```

Format of an actual warning:

```
<Warning code="2001" />
```

or

```
<Warning>
    <Description>...</Description>
    <FixHint>...</FixHint>
</Warning>
```

Predefined array-independent errors

Errors:

100. Command {Command} not supported.
101. Locale {Locale} not supported.

Warnings:

500. Replication of device {Device} is in progress.

Discovery of SRA Properties and Capabilities

SRM uses the following commands to discover SRA properties and capabilities:

- queryInfo - queries the SRA for basic properties such as name, version and UUID
- queryCapabilities - queries the SRA for supported models of storage arrays and supported SRM commands
- queryErrorDefinitions - queries the SRA for pre-defined array-specific errors
- queryConnectionParameters - queries the SRA for parameters needed to connect to the array management system and perform array management operations
- queryStrings – queries for string translations for a given locale

The UUID returned by queryInfo is used by SRM to identify the SRA. This UUID must remain the same across all versions of an SRA.

Name, version and other SRA properties (excluding UUID) are not interpreted by SRM. This information is displayed unmodified to the user and is designed to help troubleshooting by identifying the version of the SRA being used.

Information about supported replication software and array models returned by queryCapabilities is not interpreted by SRM. This information is displayed unmodified to the user and is designed to help troubleshooting by identifying supported storage arrays and replication software.

The list of supported SRM commands returned by queryCapabilities is used by SRM to determine whether or not the SRA can enable SRM workflows. The following table lists SRM workflows and the commands required to support them.

Workflow	Description	Required Commands
Discovery	Discovery of replicated storage	discoverArrays,

		discoverDevices
Test	Non-disruptive failover test	checkTestFailoverStart, testFailoverStart, testFailoverStop *Optional:* queryRecoveryPoints*
Failover	Failover	checkFailover, failover *Optional*: syncOnce†, querySyncStatus†, queryRecoveryPoints, prepareFailover
Re-protect	Reverse replication after failover	reverseReplication *Optional*: prepareReverseReplication, syncOnce, querySyncStatus, queryReplicationSettings
Restore	Restore replication after failover	restoreReplication *Optional*: prepareRestoreReplication, syncOnce†, querySyncStatus†, queryReplicationSettings

Note: Support for syncOnce and querySyncStatus is not required for failover. If a user is attempting a failover and the SRA does not support these commands, SRM will pause recovery and prompt the user to manually replicate all outstanding changes.

Note: For arrays that enable synchronous replication, syncOnce and querySyncStatus can be implemented as no-ops. However, the SRA must be able to check replication status, make sure that replication is working, and return an error if it is not.

Discovery of Replicated Storage

SRM executes the following commands to discover replicated storage:

- discoverArrays – discover storage arrays configured for replication
- discoverDevices – discover devices on a given storage array replicated to the specified target array

SRM executes discoverArrays command to discover storage arrays at local site configured for replication. Each storage array is expected to have one or more replication peers. For each storage array, discoverArrays command must return ID of the local storage array and IDs of its replication peers.

SRM executes discoverArrays command at both sites and correlates the results to identify pairs of storage arrays replicating data between sites.

> Given array pair {00A, 00B} with array 00A located at site A and array 00B located at site B, discoverArrays command is expected to return the following:

at Site A	at Site B
`<Array id="00A">` `  <PeerArrays>`	`<Array id="00B">` `  <PeerArrays>`

* Not supported by the Glenlivet release of VMware vCenter Site Recovery Manager
† Required in the Glenlivet release of VMware vCenter Site Recovery Manager. Optional in future releases.

<PeerArray id="00B"/> </PeerArrays> </Array>	<PeerArray id="00A"/> </PeerArrays> </Array>

Discovered array pairs are presented to the user who selects one or more array pairs to use with SRM. For each selected array pair, SRM executes discoverDevices command to discover standalone devices and consistency groups configured for replication.

discoverDevices command must return a list of standalone devices and consistency groups configured for replication between a pair of storage arrays specified in the request. For each consistency group discoverDevices command must return a list of devices which make up the group. For each standalone device, consistency group and device within a consistency group discoverDevices command must return unique identifier which is opaque to SRM but resolvable by the SRA.

For each device (both standalone device and device within a consistency group) which is replication source, discoverDevices command must return identification information which is used by SRM to identify the device in ESX inventory. SRM supports following types of device identification:

- FC and iSCSI:
  - World Wide Name (WWN)
  - Logical Unit Numbers
- NFS
  - Mount path
- Datastore*
  - Managed Object Identifier (MoID)

SRM executes discoverDevices command at both sites and correlates results. The results of discoverDevices commands are expected to contain matching sets of standalone devices and consistency groups. A standalone device or consistency group reported as replication source at site A is expected to be reported as replication target at site B and vice versa. For each consistency group, list of devices reported at site A is expected to match the list of devices reported at site B.

Given array pair {00A, 00B} with array 00A located at site A and array 00B located at site B, SRM generates the following input for discoverDevices command:

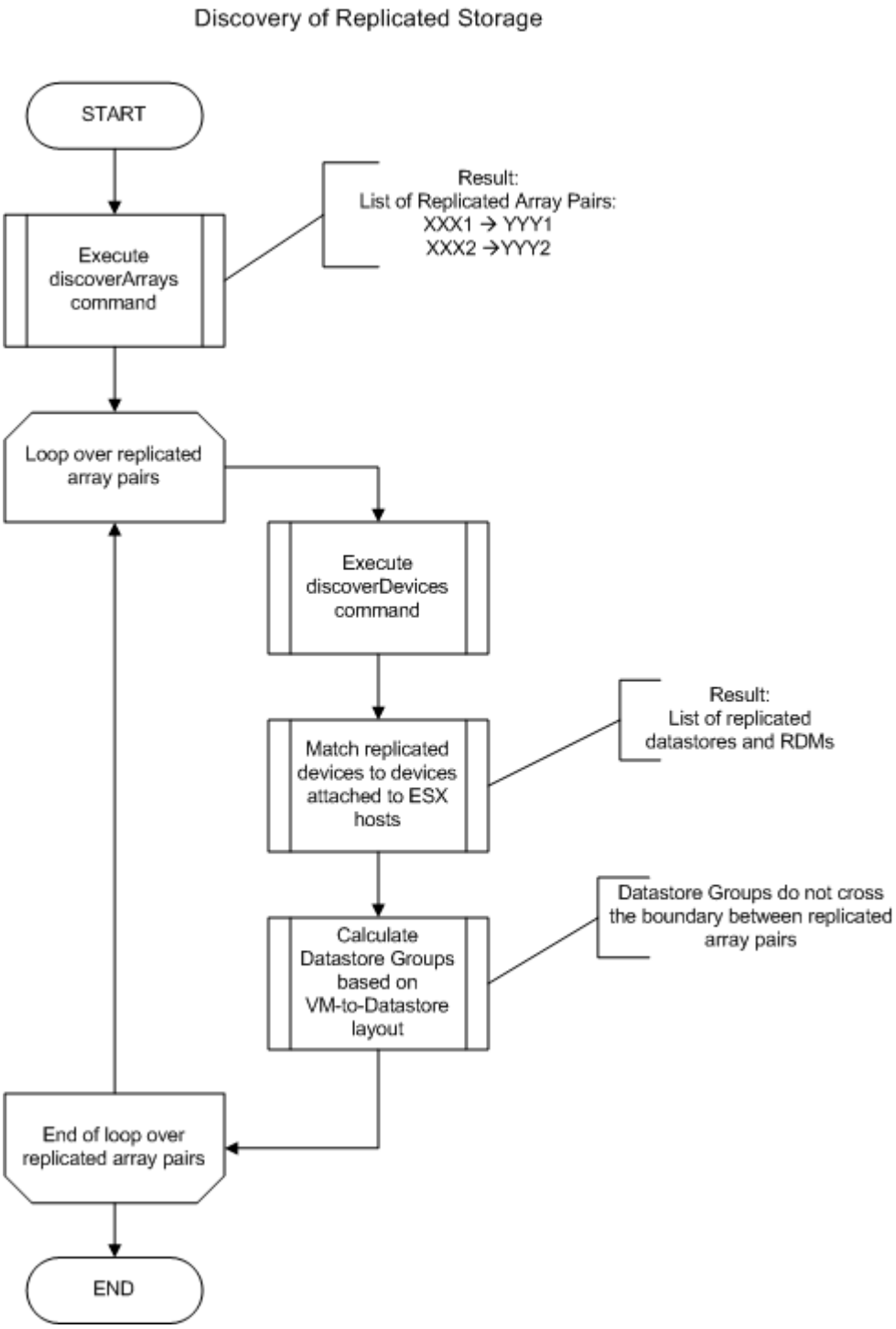
at Site A	at Site B
<DiscoverDevicesParameters> <ArrayId>00A</ArrayId> <PeerArrayId>00B</PeerArrayId> </DiscoverDevicesParameters>	<DiscoverDevicesParameters> <ArrayId>00B</ArrayId> <PeerArrayId>00A</PeerArrayId> </DiscoverDevicesParameters>

Given standalone device X replicated from 00A to 00B, discoverDevices command is expected to return the following:

at Site A	at Site B
<SourceDevice id="X" state="read-write"> <TargetDevice key="peer-of-X" /> </SourceDevice>	<TargetDevice key="peer-of-X" />

* Not supported by the Glenlivet release of VMware vCenter Site Recovery Manager.

The following flowchart shows how discoverArrays and discoverDevices fit into the Discovery of Replicated Storage workflow.



Non-disruptive Test Failover

SRM allows the user to perform a non-disruptive test of failover. The following commands are executed in the test failover workflow:

- checkTestFailoverStart* – checks configuration and runtime state of devices for test failover readiness.
- testFailoverStart – creates a temporary copy of the replicated data; the copy is writable and is deleted by testFailoverStop command. testFailoverStart must not have any effect on replication.
- testFailoverStop – deletes the temporary copy of the replicated data created by testFailoverStart

* Not supported by the Glenlivet release of VMware vCenter Site Recovery Manager

The results of checkTestFailoverStart command will be presented to the user for review. The user will have an option to cancel test failover and fix problems identified by checkTestFailoverStart first or ignore errors and proceed with the test failover.

SRM guarantees to execute a matching testFailoverStop for each testFailoverStart. Execution of testFailoverStart while a test failover is in progress must result in an error.

SRM allows a partially successful test failover. When a test failover can be executed for a subset of replicated devices, SRM proceeds with recovery of the virtual machines located on successfully failed-over devices and reports errors for the rest of the virtual machines. The SRA must support this functionality by processing as many devices as possible and reporting errors for individual devices that cannot be failed over in the test.

Failover

The following commands are executed in the failover workflow:

- checkFailover* – checks configuration and runtime state of devices for failover readiness.
- syncOnce – requests immediate replication
- querySyncStatus – queries status of the synchronization initiated by syncOnce
- queryReplicationSettings – queries replication settings for devices to be failed over
- prepareFailover – makes source devices read-only and optionally takes a snapshot of the source devices
- failover – permanently promotes target devices by stopping replication for these devices and making them writable

The results of checkFailover command will be presented to the user for review. The user will have an option to cancel failover and fix problems identified by checkFailover first or ignore errors and proceed with the failover.

SRM supports two types of failover.

- Failover when the protected site is up and running (planned migration and production test use cases);
- Failover when the protected site is not available (real disaster and temporary outage use cases).

In Planned Migration use case, SRM performs the following sequence of steps:

1. Execute queryReplicationSettings command to fetch latest replication settings;
2. Execute syncOnce command to start replicating outstanding changes;
    a. Wait for replication to complete by periodically checking replication status by executing querySyncStatus command;
3. Shutdown production VMs;
4. Unmount replicated datastores;
5. Execute prepareFailover command at protected site to unpresent devices or make them read-only;
6. Execute syncOnce command again to replicate remaining changes;
    a. Wait for replication to complete by periodically checking replication status by executing querySyncStatus command;
7. Execute failover command at recovery site.

SRM performs data sync twice to minimize downtime for production services. First sync is executed while production services are still running. That sync is expected to replicate changes accumulated since most recent scheduled sync. The second sync is executed after production services have been shutdown. Data changes accumulated between first and second syncs is expected to be relatively small and therefore second sync is expected to complete quicker.

Data sync performed after production services have been shutdown guarantees that there will be no data loss after failover.

In Disaster Recovery use case, SRM performs the following step:

* Not supported by the Glenlivet release of VMware vCenter Site Recovery Manager

1. Execute failover command at recovery site.

If and when protected site comes back up after Disaster Recovery, SRM performs the following sequence of steps to prepare protected site for failback:

1. Shutdown original production VMs if necessary;
2. Unmount original production datastores;
3. Execute prepareFailover command at protected site to unpresent devices or make them read-only.

If SRA doesn't support queryReplicationSettings, then SRM will skip the corresponding step.

If SRA doesn't support syncOnce, then SRM will pause the recovery and prompt the user to replicate outstanding changes manually (using array management tools).

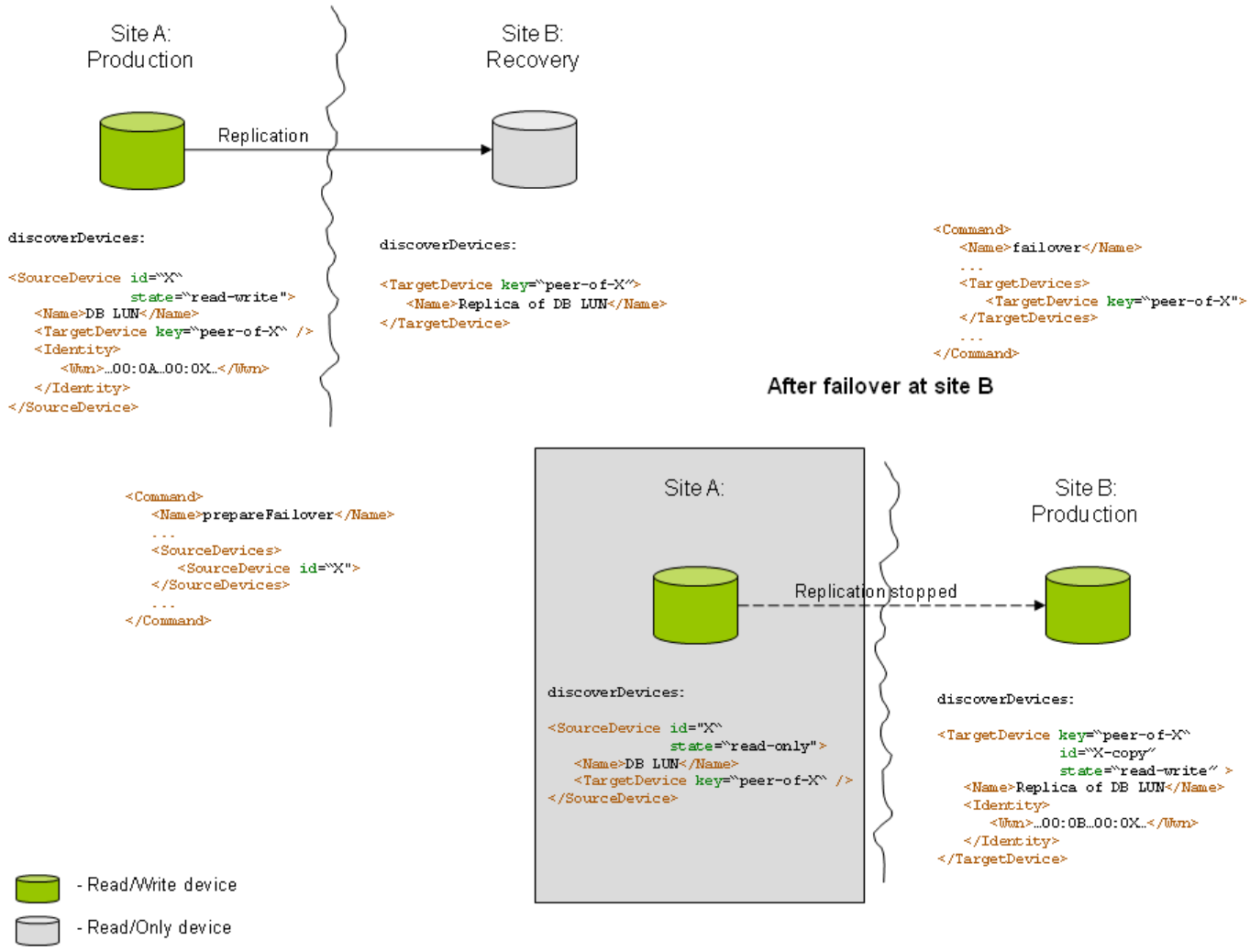
If SRA doesn't support prepareFailover, then SRM will skip the corresponding step.

SRM allows a partially successful failover. If failover succeeds for a subset of replicated devices, SRM proceeds with recovery of the virtual machines located on successfully failed-over devices and reports errors for the rest of the virtual machines. The SRA must support this functionality by processing as many devices as possible and reporting errors for individual devices and consistency groups that cannot be failed over.

SRM allows users to retry a partially successful failover. The SRA is required to support this functionality by allowing repeated (prepare) failover commands for the same devices. When a failover command is executed for a device that has already failed over, it must not change the device state, must log a warning, and must return same result as the original failover command.

The failover command is required to make target devices writable and is expected to keep original devices intact. This is important for arrays that support dynamic personality swap when sites are connected. SRA must take a snapshot of the original devices (as part of prepareFailover) before performing dynamic personality swap and revert to that snapshot if user chooses to restore replication from A to B afterwards.

The failover command is required to take a snapshot of the target device before making it writable. This is critical for robust production test experience. In case of a production site failure during revert phase of production test the failover command should restore data at recovery site from the snapshot.

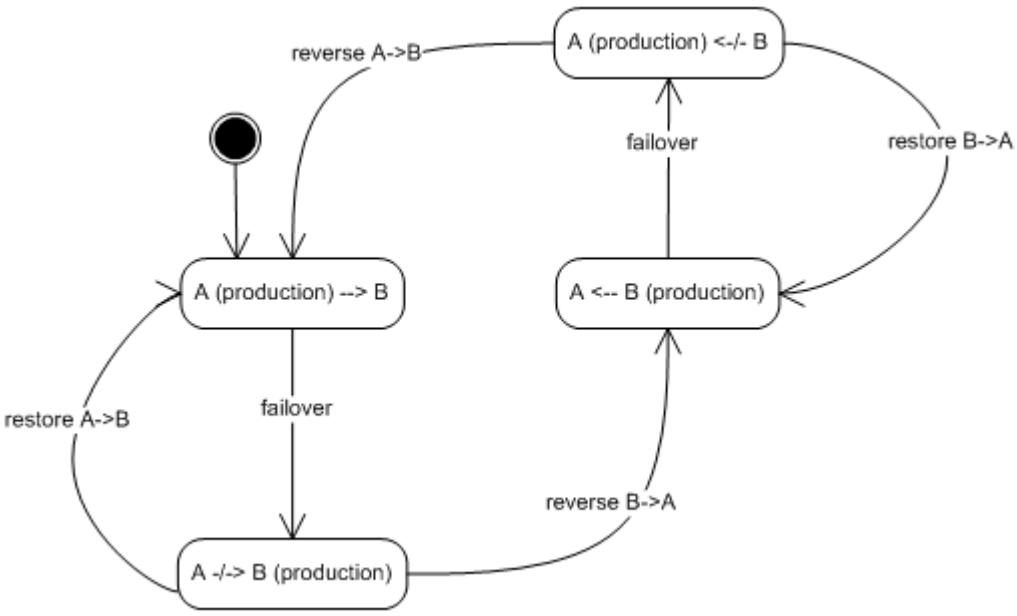


Failback

SRM treats failback as failover in the opposite direction. SRM doesn't impose any restrictions on how long a user runs virtual machines at the recovery site before returning them to the protected site. Failback consists of three separate workflows:

1. Reverse replication direction after failover and start replicating changes from the recovery site to the protected site;
2. Execute failover at the protected site (now acting as the recovery site) to bring production services back to the protected site;
3. Reverse replication direction and start replicating changes from the protected site to the recovery site. This restores both sites to their original roles.

The following state diagram shows how these workflows fit into the SRM failover-failback cycle. The diagram illustrates that SRM doesn't distinguish between failover and failback operations and treats failback as failover in the opposite direction.



Reverse Replication after Failover

The following commands are executed to reverse replication after failover and start replicating changes made at the recovery site to the original protected site.

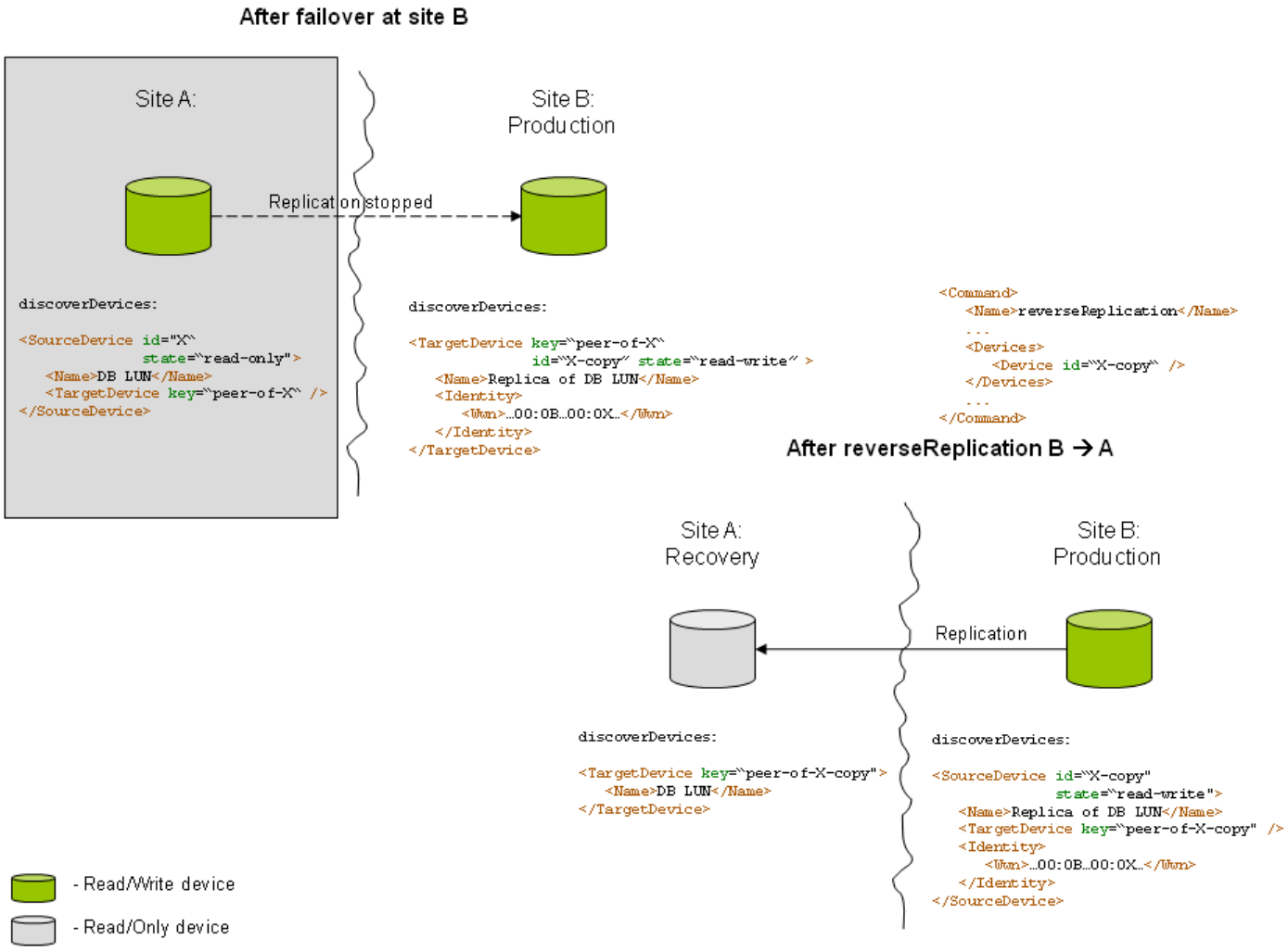
- reverseReplication – reverses replication direction to replicate data from original recovery site (B) to original protected site (A)
- syncOnce - requests immediate replication
- querySyncStatus - queries the status of the replication initiated by syncOnce.

SRM executes reverseReplication to reverse replication and start replicating changes from site B to site A. The SRA implementation of reverseReplication must replicate only changes made at site B since failover and avoid a full data sync. The SRA implementation of reverseReplication must be non-disruptive to production services running at site B.

SRM follows the reverseReplication command with syncOnce and querySyncStatus commands to initiate and monitor data transfer. This is necessary when production services have been running at site B for long enough to create a large amount of data that must be replicated back to site A. SRM will display the status of this replication to the user. Once the replication is complete SRM will indicate to the user that system is ready for failback. (After powering off production virtual machines at site B, SRM will execute syncOnce again as part of the failover workflow at site A.)

The reverseReplication command must be implemented in such a way that at any given point in time between the command is initiated and the followup data synchronization is completed there is a copy of the data available at the destination. It could be a copy of the production data saved in a snapshot taken during prepareFailover. This is critical to ensure that the customer can recover production services if the now protection site goes down during re-protect.

If the SRA supports queryReplicationSettings, SRM will execute that command before the (prepare) failover and pass the results to reverseReplication command. The SRA is expected to use this information as a hint on how to configure the replication after reversing replication direction.



Depending on the array replication architecture, array operations to configure replication in the opposite direction may need to be executed at either or both sites, perhaps in a particular sequence. SRM defines an optional prepareReverseReplication command that is executed before reverseReplication. The default model is to execute reverseReplication at the desired replication source. An SRA can specify a different model in the queryCapabilities command results as described in the following table.

prepareReverseReplication	reverseReplication	queryCapabilities response
	executed at the replication source	`<Command name="reverseReplication"><ExecutionLocation>source</ExecutionLocation></Command>`
	executed at the replication target	`<Command name="reverseReplication"><ExecutionLocation>target</ExecutionLocation></Command>`
executed at the replication target	executed at the replication source	`<Command><Name>prepareReverseReplication</Name></Command>` `<Command name="reverseReplication"><ExecutionLocation>source</ExecutionLocation></Command>`
executed at the replication source	executed at the replication target	`<Command><Name>prepareReverseReplication</Name></Command>`

		<Command name="reverseReplication"> <ExecutionLocation>target</ExecutionLocation> </Command>

Replication reversal must be repeatable. If replication is already configured in the requested direction, the SRA must report success. We recommend logging a warning and optionally returning a warning in the response. This requirement provides flexibility to the users to perform replication reversal on the array manually.

Restore Replication after Failover

SRM allows users to perform disruptive test by executing failover workflow followed by restore replication workflow. This feature allows users to perform a disruptive test identical to failover in case of a disaster. There will be downtime in production services in this case.

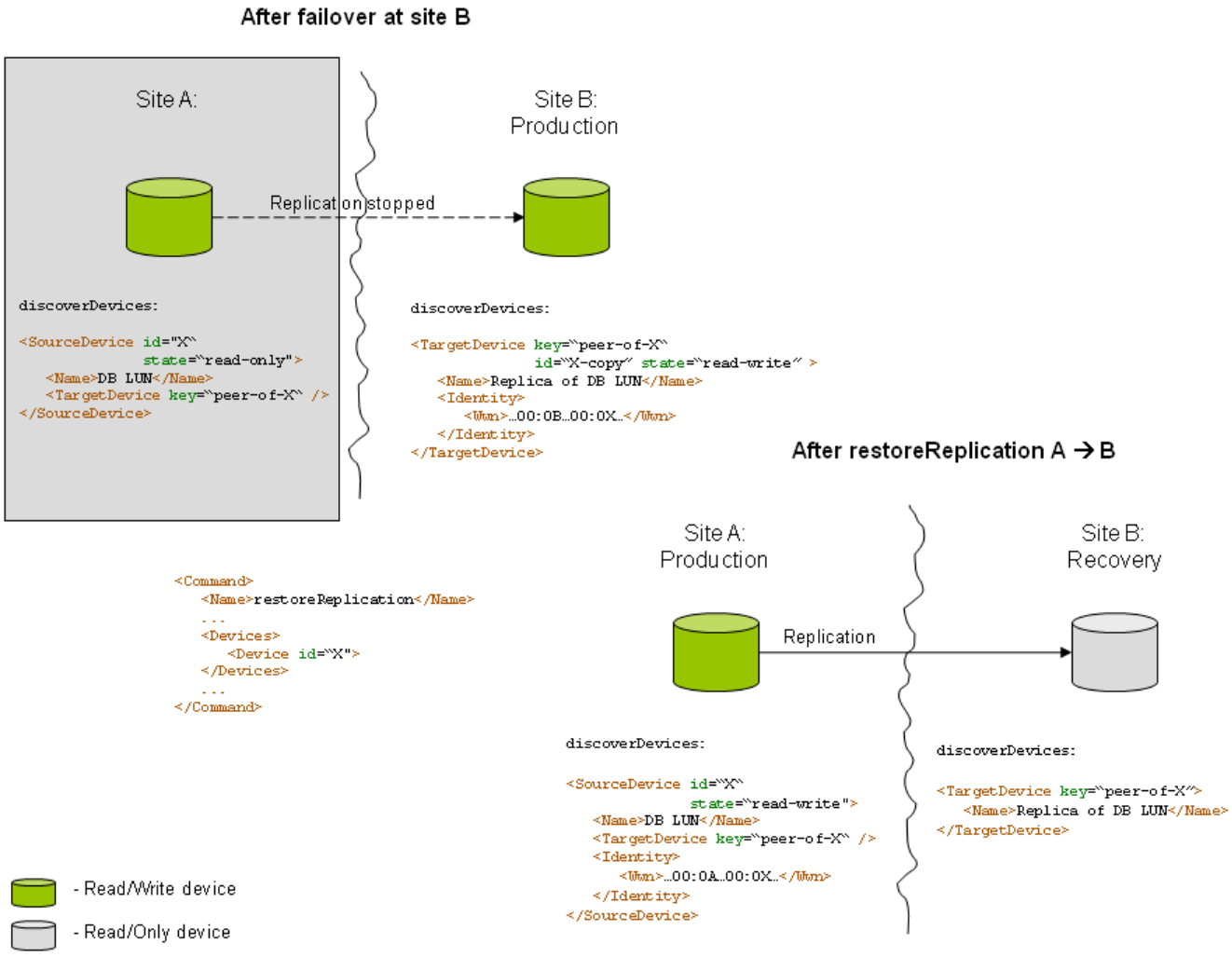
- restoreReplication – restores replication from original protected site (A) to original recovery site (B) discarding all changes made at site (B) since failover
- syncOnce - requests immediate replication
- querySyncStatus - queries the status of the replication initiated by syncOnce.

SRM executes restoreReplication to restore replication from protected site A to recovery site B after executing failover at site B. The SRA implementation of restoreReplication must discard all changes made at site B since failover and overwrite target devices with data from site A. If prepareFailover command was executed and devices at site A were made read-only before failover then restoreReplication command is expected to make devices writable again. If failover command performed a dynamic swap of devices then restoreReplication command must restore data on devices at site A from the snapshot taken by prepareFailover to a pre-failover state.

SRM follows the restoreReplication command with syncOnce and querySyncStatus commands to initiate and monitor data transfer. SRM will display status of this data transfer to the user. Once that data transfer is complete SRM will indicate to the user that system is ready for test and failover.

The restoreReplication command must be implemented in such a way that at any given point in time between the command is initiated and followup data synchronization is completed there is a copy of the data available at the destination. It could be a copy of the replicated data saved in a snapshot taken during failover. This is critical to ensure that the customer can recover production services if the protection site goes down during revert.

If the SRA supports queryReplicationSettings, SRM will execute that command before the (prepare) failover and pass the results to restoreReplication. The SRA is expected to use this information as a hint on how to configure the replication after restoring replication.



Depending on the array replication architecture, array operations to restore replication may need to be executed at either or both sites, perhaps in a particular sequence. SRM defines an optional prepareRestoreReplication command that is executed before restoreReplication. The default model is to execute restoreReplication at the desired replication source. An SRA can specify a different model in the queryCapabilities command results as described in the following table.

prepareRestoreReplication	restoreReplication	queryCapabilities response
	executed at the replication source	<Command> <Name>restoreReplication</Name> <ExecutionLocation>source</ExecutionLocation> </Command>
	executed at the replication target	<Command> <Name>restoreReplication</Name> <ExecutionLocation>target</ExecutionLocation> </Command>
executed at the replication target	executed at the replication source	<Command> <Name>prepareRestoreReplication</Name> <ExecutionLocation>target</ExecutionLocation> </Command> <Command> <Name>restoreReplication</Name> <ExecutionLocation>source</ExecutionLocation> </Command>

executed at the replication source	executed at the replication target	<Command> <Name>prepareRestoreReplication</Name> <ExecutionLocation>source</ExecutionLocation> </Command> <Command> <Name>restoreReplication</Name> <ExecutionLocation>target</ExecutionLocation> </Command>

Restore replication operation must be repeatable. If replication is already configured in the requested direction, the SRA must report success. We recommend logging a warning and optionally returning a warning in the response. This requirement provides flexibility to the users to restore replication on the array manually.

End-to-End Workflows

The following diagram illustrates how SRA is used in end-to-end SRM workflows.

Site A	Site B
Load SRA	**Load SRA**

• queryInfo
• queryCapabilities
• queryConnectionParameters
• queryStrings
• queryErrorDefinitions
 	
• Notes:
  ○ if SRA doesn't support queryStrings command, that command is not executed
  ○ if SRA doesn't support queryErrorDefinitions command, that command is not executed
	• queryInfo
	• queryCapabilities
	• queryConnectionParameters
	• queryStrings
	• queryErrorDefinitions

Create Array Manager	
• discoverArrays	
	Create Array Manager
	• discoverArrays
Enable Array Pair	
• discoverDevices • Identify replicated datastores and RDMs. Calculate Datastore Groups based on consistency groups and VM-to-Datastore topology.	• discoverDevices
Create Protection Group	
• Protection Group contains one or more Datastore Groups.	
	Create Recovery Plan
	• Recovery Plan contains one or more Protection Groups.

Protected Site	Recovery Site
	Test

• [if Sync Data option selected]
  ○ queryReplicationSettings
  ○ syncOnce [+ querySyncStatus]
--- 	
• Notes:
  ○ if SRA doesn's support queryReplicationSettings, that command is not executed
  ○ if SRA identified sync location as 'target', the above commands are executed at the recovery site	

	• testFailoverStart • Rescan HBAs • Mount datastores • Register VMs • Customize and Power On VMs
	Test Cleanup
	• Power Off VMs • Unregister VMs • Unmount datastores • testFailoverStop • Rescan HBAs

<table>
<tr><th colspan="2">Planned Migration to B</th></tr>
<tr><td>

queryReplicationSettings
syncOnce [+ querySyncStatus]
Shutdown VMs
Unregister VMs
Unmount datastores
prepareFailover
syncOnce [+ querySyncStatus]

<hr>

Notes:

if SRA identified sync location as 'target', the above commands are executed at the recovery site
if SRA doesn't support prepareFailover, that command is not executed

</td><td></td></tr>
<tr><td></td><td>

failover
Rescan HBAs
Mount datastores
Register VMs
Customize and Power On VMs

</td></tr>
<tr><th colspan="2">Reprotect B to A</th></tr>
<tr><td>

prepareReverseReplication

</td><td></td></tr>
<tr><td></td><td>

reverseReplication
queryReplicationSettings
syncOnce [+ querySyncStatus]

<hr>

Notes:

the above example is for SRA which supports prepareReverseReplication and identified reverseReplication location as 'source'
if SRA doesn't support prepareReverseReplication, that command is not executed
if SRA identifies reverseReplication location as 'target', then prepareReverseReplication is executed at B followed by reverseReplication executed at A
if SRA identifies sync location as 'target', queryReplicationSettings, syncOnce and querySyncStatus commands are executed at the original protected site

</td></tr>
</table>

• Rescan HBAs	

Recovery Site	Protected Site
Test	

	• [if Sync Data option selected]
	  ○ queryReplicationSettings
	  ○ syncOnce [+ querySyncStatus]

• testFailoverStart • Rescan HBAs • Mount datastores • Register VMs • Customize and Power On VMs	
Test Cleanup	
• Power Off VMs • Unregister VMs • Unmount datastores • testFailoverStop • Rescan HBAs	
Planned Migration to A	
	• queryReplicationSettings • syncOnce [+ querySyncStatus] • prepareFailover • syncOnce [+ querySyncStatus]
• failover • Rescan HBAs • Mount datastores • Register VMs • Customize and Power On VMs	
Reprotect A to B	
	• prepareReverseReplication
• reverseReplication • queryReplicationSettings • syncOnce [+ querySyncStatus]	
	• Rescan HBAs

The following table shows the sequence of actions in a disaster recovery use case. Some SRAs swap the roles of source and target devices automatically on failover. The following description includes SRM handling for such SRAs.

Protected Site	Recovery Site
	Disaster Recovery (site A is down)
	• Failover • Rescan HBAs • Mount datastores • Register VMs • Customize and Power On VMs
Site A comes back up	
• prepareFailover only if the original production device is reported as source	
Reprotect B to A	
• prepareReverseReplication only if the original production device is reported as source or demotedSource	

	• reverseReplication (the recovered device must be reported as a promotedTarget)
	• queryReplicationSettings
	• syncOnce [+ querySyncStatus]
	
	• Notes:
	  ○ the above example is for SRA which supports prepareReverseReplication and identified reverseReplication location as 'source'
	  ○ if SRA doesn't support prepareReverseReplication, that command is not executed
	  ○ if SRA identifies reverseReplication location as 'target', then prepareReverseReplication is executed at B followed by reverseReplication executed at A; in this case reverseReplication is executed at A only if the original production device is reported as source or demoted source. Warning: If a discoverDevices command is executed immediately after running prepareReverseReplication, the state of original production devices can change from source/demotedSource to target and this can cause SRM to skip running reverseReplication for these

	devices. This usually happens when running multiple recovery plans in parallel. SRM will take care of handling this particular situation. o if SRA identifies sync location as 'target', queryReplicationSettings, syncOnce and querySyncStatus commands are executed at A

Commands Syntax

queryInfo

Returns information describing the SRA.

Format of the queryInfo request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>queryInfo</Name>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
</Command>
```

Format of the queryInfo response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    <AdapterInfo>
        <Name stringId="AdapterName">Dell Equallogic PS Series SRA</Name>
        <Version>1.2</Version>
        <Vendor stringId="AdapterVendor">Dell EqualLogic. Inc</Vendor>
        <Uuid>xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx</Uuid>
        <HelpUrl>http://www.vendor.com/sra.html</HelpUrl>
    </AdapterInfo>
</Response>
```

Note: Name and Vendor strings are localizable.

The UUID is used by SRM to identify the SRA. It must remain the same across all versions of an SRA. (The Name, Version and Vendor properties may change.)

Table 1 queryInfo format

Element Name	Content	Description
Command	Name, OuputFile, StatusFile, LogLevel, LogDirectory	
Name		Command name, “queryInfo”.
OutputFile		Local file path to hold response XML.
LogLevel		Log level. Valid values are “trivia”, “verbose”, “info”, “warning”, “error”.

LogDirectory		Local directory path to hold SRA log files.
Response	(AdapterInfo, Warnings?) | Error	
AdapterInfo	Name, Version, Vendor, Uuid, HelpUrl?	Information describing the SRA
Name	Attributes: stringId (optional)	Adapter name displayed to the user. Localizable.
Version		Adapter version displayed to the user.
Vendor	Attributes: stringId (optional)	Vendor name displayed to the user. Localizable.
Uuid		Universally unique identifier of the SRA as defined in RFC 4122.
HelpUrl		URL for on-line documentation for the SRA. Optional.

queryStrings

Returns translations of the strings for a given locale.

Format of the queryStrings request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>queryStrings</Name>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <QueryStringsParameters>
        <Locale>ja</Locale>
    </QueryStringsParameters>
</Command>
```

Note: Value of the Locale element is a language tag, as defined in RFC 1766.

Format of the queryStrings response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    <Strings locale="ja">
       <!-- queryConnectionParameter strings -->
        <String id="PrimaryTitle">Primary SAN in Japanese</String>
        <String id="PrimaryHint"
            >Primary SAN connection parameters in Japanese</String>
        <String id="SpATitle">IP Address of SP-A in Japanese</String>
        <String …/>
        <!-- queryErrorDefinitions strings -->
        <String id="Description1001">...</String>
        <String id="FixHint1001">...</String>
        <String …/>
   </Strings>
</Response>
```

Table 2 queryStrings format

Element Name	Content	Description
Command	Name, OuputFile, StatusFile, LogLevel, LogDirectory, QueryStringsParameters	
Name		Command name, “queryStrings”
QueryStringsParameters	Locale	
Locale		Language tag as defined in RFC 1766
Response	(Strings, Warnings?) | Error	
Strings	String+ Attributes: locale	UTF-8 encoded string localized to the specified locale Value of locale attribute must be the one specified in the request.
String	Attributes: id	Translation of the string identified by specified id. Value of id attribute must be unique.

queryErrorDefinitions

Returns pre-defined array specific error and warning descriptions.

Format of the queryErrorDefinitions request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>queryErrorDefinitions</Name>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
</Command>
```

Format of the queryErrorDefinitions response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
   <ErrorDefinitions>
      <ErrorDefinition code="1001">
         <Description stringId="Description1001"
              >Failed to connect to CLI at {CliAddress}</Description>
         <FixHint stringId="FixHint1001"
              >Make sure the address {CliAddress} is correct and accessible</FixHint>
      </ErrorDefinition>
      ...
   </ErrorDefinitions>
   <WarningDefinitions>
      <WarningDefinition code="2001">
         <Description stringId="Description2001">...</Description>
         <FixHint stringId="FixHint2001">...</FixHint>
      </WarningDefinition>
      ...
```

```
   </WarningDefinitions>
</Response>
```

Example of a predefined error return:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    …
    <Error code="1001">
       <CliAddress>10.20.30.40</CliAddress>
    </Error>
    …
</Response>
```

Example of an undefined error return:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    …
    <Error>
        <Description
            >Failed to connect to CLI at 10.20.30.40</Description>
        <FixHint
            >Make sure the address 10.20.30.40 is correct
             and accessible</FixHint>
    </Error>
    …
</Response>
```

Example of warnings return:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    …
    <Warnings>
        <Warning code="2001" />
        <Warning>
            <Description>...</Description>
            <FixHint>...</FixHint>
        </Warning>
        <Warning …/>
    </Warnings>
    …
</Response>
```

Table 3 queryErrorDefinitions format

Element Name	Content	Description
Command	Name, OutputFile, StatusFile, LogLevel, LogDirectory	
Name		Command name, “queryErrorDefinitions”
Response	(ErrorDefinitions, WarningDefinitions? Warnings?) | Error	

ErrorDefinitions	ErrorDefinition+	Definitions of errors.
ErrorDefinition	Description, FixHint Attributes: code	Definition of an error. Value of code attribute must be unique.
Description		Problem description. May contain placeholders.
FixHint		Suggestion about how the user can try to fix the problem. May contain placeholders.
WarningDefinitions	WarningDefinition+	Definitions of warnings.
WarningDefinition	Description, FixHint Attributes: code	Definition of a warning. Value of code attribute must be unique.

queryCapabilities

Returns SRA capabilities, including supported versions of the array replication software, supported array models, and supported SRM commands.

Information about supported replication software and array models is not interpreted by SRM. This information is displayed unmodified to the user and is intended to help troubleshooting.

The list of supported SRM commands is used by SRM to determine the set of SRM workflows that the SRA supports.

The SRA must support simultaneous execution of commands so that SRM can support bi-directional protection and simultaneous execution of recovery plans.

The following capabilities are optional:

- MultiArrayDiscovery

  Ability to manage multiple storage arrays with a single set of connection parameter values.

- ConsistencyGroups

  Ability to manage group of devices replicated as a single unit.

- DynamicAccessRestriction

  Ability to present snapshot and promoted devices to a list of initiators specified in testFailoverStart and failover requests.

- VmImageConsistency

  VmImageConsistency capability is designed to support array replication solutions which are capable of replicating quiesced images of VMs. A solution like this would integrate with array replication to take quiesced snapshots of VMs before creating a replica:

  1. Create quiesced snapshots of all VMs on a datastore or a set of datastores.
  2. Create a replica.
  3. Delete snapshots created in Step 1.

This process would repeat every time a replica is created (based on replication schedule specified by the user).

As a result, recovered VMs will have a quiesced snapshot in addition to crash-consistent snapshot (current). SRM is capable of reverting recovered VM to a quiesced snapshot before customizing IP address and powering on the VM during Test and Recovery. To trigger this logic, SRA needs to identify that snapshot/promoted device contains VMs with quiesced snapshots in testFailoverStart/failover response. In addition, SRA needs to specify a regular expression which can be used to identify a quiesced VM snapshot by name in queryCapabilites response.

When VmImageConsistency feature is supported, the user expects that data sync requested during testFailover creates replicas with quiesced VM images. Therefore, SRA must ensure that syncOnce command creates replicas with quiesced VM images.

The VM snapshot must be taken with quiesce flag on and memory flag off.

- OneLunPerIScsiTarget

  OneLunPerIScsiTarget capability is designed to support arrays which create separate iSCSI target for each LUN.

  iSCSI initiators automatically add newly discovered iSCSI targets to a list of 'static iSCSI targets' but do not automatically remove entries from that list when corresponding iSCSI target is destroyed at the array. New targets are created and destroyed on the array every time SRM performs a Test Failover.

  If SRA reports OneLunPerIScsiTarget capability, SRM will update the list of 'static iSCSI targets' in iSCSI initiators and remove targets corresponding to snapshot LUNs during Test Failover cleanup.

  SRM will apply similar logic during Planned Migration to remove targets corresponding to demoted LUNs at original production site.

Format of the queryCapabilites request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>queryCapabilities</Name>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
</Command>
```

Format of the queryCapabilities response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
   <AdapterCapabilities>
      <ReplicationSoftwares>
         <ReplicationSoftware>
            <Name>Dell Equallogic Group Manager GUI</Name>
            <Version>3.3.2</Version>
         </ReplicationSoftware>
         <ReplicationSoftware …/>
      </ReplicationSoftwares>
      <ArrayModels>
         <ArrayModel>
            <Name>MODEL 70-0011</Name>
            <Vendor stringId="AdapterVendor">Dell EqualLogic. Inc</Vendor>
         </ArrayModel>
```

```
         <ArrayModel …/>
      </ArrayModels>
      <Features>
         <MultiArrayDiscovery />
         <ConsistencyGroups />
         <DynamicAccessRestriction />
         <DeviceIdentification>initiator+number</DeviceIdentification>
         <Protocols>
            <Protocol>FC</Protocol>
            <Protocol>iSCSI</Protocol>
         </Protocols>
         <VmImageConsistency>
            <SnapshotNamePattern>^foo-</SnapshotNamePattern>
         <VmImageConsistency>
         <OneLunPerIScsiTarget />
      </Features>
      <Commands>
         <Command name="queryInfo" />
         <Command name="queryStrings">
            <Locales>
               <Locale id="ja" />
               <Locale id="de" />
            </Locales>
         </Command>
         <Command name="queryErrorDefinitions" />
         <Command name="queryCapabilities" />
         <Command name="queryDeviceDetails">
            <CustomProperties>
               <CustomProperty name="…" stringId="…"
                  >Property label in English</CustomProperty>
               <CustomProperty …/>
                  …
            </CustomProperties>
         </Command>
         <Command name="discoverArrays" />
         <Command name="discoverDevices" />
         <Command name="syncOnce">
            <ExecutionLocation>source</ExecutionLocation>
         </Command>
         <Command name="querySyncStatus" />
         <Command name="queryReplicationSettings" />
         <Command name="queryRecoveryPoints" />
         <Command name="checkTestFailoverStart" />
         <Command name="testFailoverStart" />
         <Command name="testFailoverStop" />
         <Command name="checkFailover" />
         <Command name="prepareFailover" />
         <Command name="failover" />
         <Command name="prepareReverseReplication" />
         <Command name="reverseReplication">
            <ExecutionLocation>source</ExecutionLocation>
         </Command>
         <Command name="prepareRestoreReplication" />
         <Command name="restoreReplication">
            <ExecutionLocation>source</ExecutionLocation>
         </Command>
      </Commands>
   </AdapterCapabilities>
</Response>
```

Array replication architecture dictates whether the syncOnce, reverseReplication, and restoreReplication commands should execute at the replication source or target. Default execution locations for these commands are:

syncOnce	source (original protected site)
reverseReplication	source (original recovery site)
restoreReplication	source (original protected site)

The defaults can be overwritten by specifying <ExecutionLocation> tags. Valid values are “source” and “target”.

querySyncStatus and queryReplicationSettings are always executed at the same site as syncOnce.

prepareReverseReplication is always executed at the opposite site of reverseReplication.

prepareRestoreReplication is always executed at the opposite site of restoreReplication.

The testFailoverStart and failover commands can support dynamic access restrictions to the recovered devices if such access is supported by the array architecture and SCSI transport. SRAs that support dynamic access restrictions must include the XML tag <DynamicAccessRestriction /> in their response. When an SRA supports dynamic access restrictions, SRM command requests will include a list of initiators that need access to recovered volumes.

If syncOnce is supported, then querySyncStatus must be supported as well. If syncOnce is not supported, then querySyncStatus must not be supported as well.
If prepareReverseReplication is supported, then reverseReplication must be supported as well.

If prepareRestoreReplication is supported, then restoreReplication must be supported as well.

If restoreReplication is supported with execution location at the target, then prepareRestoreReplication must be supported to enable restore workflow in case when recovery site is not available.

Table 4 queryCapabilities format

Element Name	Content	Description
Command	Name, OutputFile, StatusFile, LogLevel, LogDirectory	
Name		Command name, queryCapabilities”.
Response	(AdapterCapabilities, Warnings?) | Error	
AdapterCapabilites	ReplicationSoftwares, ArrayModels, Features, Commands	
ReplicationSoftwares	ReplicationSoftware+	
ReplicationSoftware	Name, Version	
Name		Replication software name
Version		Replication software version
ArrayModels	ArrayModel+	

ArrayModel	Name, Vendor	Supported array model
Name		Array model name
Vendor	Attributes: stringId (optional)	Array model vendor. Localizable.
Features	MultiArrayDiscovery?, ConsistencyGroups?, DynamicAccessRestriction?, DeviceIdentification?, Protocols, VmImageConsistency?, OneLunPerIScsiTarget?	Supported features
MultiArrayDiscovery		Optional. If present indicates that discoverArrays command may return multiple arrays.
ConsistencyGroups		Optional. If present indicates that SRA and supported storage arrays support consistency groups.
DynamicAccessRestriction		Optional. If present indicates that SRA supports dynamic access restriction of the recovered devices during test and failover.
DeviceIdentification		Device identification scheme used by the SRA. Valid values are: wwn, initiator+number, number, number+target, initiator+number+target. Not applicable for NFS or Datastore.
Protocols	Protocol+	Supported protocols (FC, iSCSI, NFS, Datastore).
Protocol		Supported protocol. Valid values are: FC, iSCSI, NFS, Datastore. Case sensitive. “Datastore” is for secondary storage replication solutions which report datastores instead of raw storage devices.
VmImageConsistency	SnapshotNamePattern	Optional. If present indicates that underlying replication software takes a snapshot of the VM before replication. The snapshot is taken with quiesce flag on to ensure that disk snapshot represents a consistent state of the guest file systems.
OneLunPerIScsiTarget		Optional. If present indicates that underlying storage creates separate iSCSI target

		for each LUN. Only applies to SRAs which support iSCSI protocol and use number+target or initiator+number+target device identification.
SnapshotNamePattern		Regular expression describing the pattern of the name of the snapshot taken before replication. Regular expression must be Perl 5 compatible.
Commands	Command+	Supported commands. The following commands must be supported*: • queryInfo • queryCapabilities • queryConnectionParameters • discoverArrays • discoverDevices • testFailoverStart • testFailoverStop • failover • reverseReplication • restoreReplication
Command	Locales?, ExecutionLocation? , CustomProperties? Attributes: name	Supported command. Command name is specified in the name attribute. Valid values are: • queryInfo • queryStrings • queryErrorDefinitions • queryCapabilities • queryConnectionParameters • discoverArrays • discoverDevices • queryDeviceDetails† • syncOnce • querySyncStatus • queryReplicationSettings • queryRecoveryPoints† • checkTestFailoverStart • testFailoverStart • testFailoverStop • checkFailover • prepareFailover • failover

* syncOnce and querySyncStatus commands are required for the Glenlivet release of VMware vCenter Site Recovery Manager.
† Not supported by the Glenlivet release of VMware vCenter Site Recovery Manager.

		• prepareReverseReplication • reverseReplication • prepareRestoreReplication • restoreReplication Execution location can be specified for syncOnce, reverseReplication, and restoreReplication commands only. CustomProperties must be specified for queryDeviceDetails if supported.
Locales	Locale+	Optional. List of supported locales. Must be specified for queryStrings command.
Locale	Attributes: id	Supported locale. Value of id attribute is language tag as defined in RFC 1766.
CustomProperties	CustomProperty+	Optional. SRA specific properties of a device returned by queryDeviceDetails command. Must be specified for queryDeviceDetails command.
CustomProperty	Attributes: name, stringId (optional)	SRA specific property of a device. Property name is specified in the name attribute. Property label for display is specified in the tag content. Localizable.
ExecutionLocation		Optional. Execution location relative to the replication. Valid values are "source" and "target". Applicable only to syncOnce, reverseReplication, and restoreReplication commands. Default value is "source".

queryConnectionParamers

Returns a list of parameters required to establish connection with the array management system.

SRA connection parameters can be split into multiple groups. Each group of parameters consists of the following:

- one or more named addresses;
- username if required;
- password if required;
- zero or more named opaque parameters.

Each group of parameters must have a unique ID. Each named parameter in the group (such as an address or opaque parameter) must have an ID that is unique within the group.

Each group and each parameter in the group must have a title and optionally may have a hint message. The title is a localized string describing the parameter to the user. The hint message is a localized string providing more detail about what value the user should supply for the parameter.

A parameter can be defined as optional. By default all parameters are considered required.

Format of the queryConnectionParameters request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>queryConnectionParameters</Name>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
</Command>
```

Format of the queryConnectionParameters response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
   <AdapterConnectionParameters>
      <Connection id="primary">
         <Title stringId="PrimaryTitle">Primary SAN</Title>
         <Hint stringId="PrimaryTitle"
           >Primary SAN connection parameters</Hint>
         <Addresses>
            <Address id="spA">
               <Title stringId="SpATitle">IP Address of SP-A</Title>
               <Hint stringId="SpAHint"
                 >Enter IP address of the Storage Processor A</Hint>
            </Address>
            <Address …/>
         </Addresses>
         <Username>
            <Title stringId="PrimaryUsernameTitle">Username</Title>
            <Hint stringId="PrimaryUsernameHint"
              >Enter username for primary SAN</Hint>
         </Username>
         <Password>
            <Title stringId="PrimaryPasswordTitle">Password</Title>
            <Hint stringId="PrimaryPaswordHint"
              >Enter password for primary SAN</Hint>
         </Password>
         <Opaques>
            <Opaque id="volumeNameFilter" optional="true">
               <Title stringId="VolumeNameFilterTitle"
                 >Volume name prefix limiting discovery</Title>
               <Hint stringId="VolumeNameFilterHint"
                 >Leave empty for full discovery</Hint>
            </Opaque>
            <Opaque …/>
         </Opaques>
      </Connection>
      <Connection id="primary2" …/>
   </AdapterConnectionParameters>
</Response>
```

Format of the Connection tag passed to commands which communicate with the SAN management system:

```
<Connections>
   <Connection id="primary">
      <Addresses>
         <Address id="spA">10.20.30.40</Address>
      </Addresses>
      <Username>admin</Username>
      <Password>secret</Password>
      <Opaques>
         <Opaque id="volumeNameFilter">srm-</Opaque>
      </Opaques>
   </Connection>
   <Connection …/>
</Connections>
```

Table 5 queryConnectionParameters format

Element Name	Content	Description
Command	Name, OutputFile, LogLevel, LogDirectory	
Name		Command name, "queryConnectionParameters".
Response	(AdapterConnectionParameters, Warnings?) | Error	
AdapterConnectionParameters	Connection+	
Connection	Title, Hint, Addresses, Username?, Password?, Opaques? Attributes: id	Group of parameters required to establish a connection to the SAN management system. Value of id attribute must be unique.
Title	Attributes: stringId (optional)	Title for the whole group of parameters displayed to the user. Localizable. Max length: 64 characters.
Hint	Attributes: stringId (optional)	One or two sentences describing this group of parameters to the user. Localizable. Max length: 256 characters.
Addresses	Address+	
Address	Title, Hint Attributes: id	Defines a parameter of type "address". SRA is require to define at least one such parameter. Value of id attribute must be unique.
Title	Attributes: stringId (optional)	Parameter title displayed to the user. Localizable. Max length: 64 characters.

Hint	Attributes: stringId (optional)	One or two sentences describing this parameter to the user. Localizable. Max length: 256 characters.
Username	Title, Hint	Optional. If defined, SRM will protect the value of this parameter entered by user.
Password	Title, Hint	Optional. If defined, SRM will protect the value of this parameter entered by user.
Opaques	Opaque+	Optional.
Opaque	Title, Hint Attributes: id, optional (optional)	Defines a parameter opaque to SRM. Value of id attribute must be unique. Attribute "optional" is a boolean flag indicating whether this parameter is optional. By default all parameters are required. Valid values are "true" and "false".

Note: Optional Opaque connection parameters can be used to provide replicated device filtering functionality to the end user. By default discoverDevices command returns the list of all replicated devices for a given pair of replicated storage arrays. In some cases this list can be very long, so the user may want to limit device discovery to the subset of devices managed by SRM. An SRA may define one or more Opaque parameters to specify a filter for device discovery. For example, there could be a parameter named volumeNamePrefix or deviceNamePrefix that could be used to limit the discovery to replicated volumes/LUNs whose names begin with the specified string. It is strongly recommended that SRAs implement device name filtering. The filter should apply to all SRA commands, including discoverDevices where the SRA should skip devices that do not match the given pattern. This helps SRA commands to be more efficient and make the SRM recovery complete quicker.

discoverArrays

Returns the following information about storage arrays configured for replication:

- unique identifier of the storage array;
- user-friendly name of the storage array (optional);
- array model and vendor;
- replication software name and version;
- list of replication peers;

SRM doesn't interpret the unique identifier of the storage array. However this ID must be unique because SRM will use it to specify the array when executing other commands.

Array model and vendor are localizable strings.

Format of the discoverArrays request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>discoverArrays</Name>
    <Connections …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
</Command>
```

Format of the discoverArrays response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
   <Arrays>
      <Array id="10000A0">
         <Name>Foo-A0</Name>
         <Model>
            <Name>Celerra</Name>
            <Vendor stringId="CelerraVendor">EMC - Celerra iSCSI</Vendor>
         </Model>
         <ReplicationSoftware>
            <Name>Celerra Replicator v2</Name>
            <Version>2.0</Version>
         </ReplicationSoftware>
         <PeerArrays>
             <PeerArray id="10000B1"/>
            <PeerArray id="10000B2"/>
         </PeerArrays>
      </Array>
      <Array …/>
   </Arrays>
</Response>
```

Table 6 discoverArrays format

Element Name	Content	Description
Command	Name, OutputFile, StatusFile, LogLevel, LogDirectory, Connections	
Name		Command name, “discoverArrays”.
Response	(Arrays, Warnings?) | Error	
Arrays	Array*	
Array	Name?, Model, ReplicationSoftware, PeerArrays Attributed: id	Represents storage array configured for replication. Value of id attribute must be unique.
Name		Optional. User-friendly name of the array. If not specified then array ID will be displayed to the user.
Model	Name, Vendor	Array model.
Name		Array model name.
Vendor	Attributes: stringId (optional)	Array vendor. Localizable.
ReplicationSoftware	Name, Version	Version of replication software.
Name		Replication software product name.
Version		Replication software version.

PeerArrays	PeerArray+	
PeerArray	Attributed: id	Represents storage array configured as a replication target. Value of id attribute must be unique.

discoverDevices

Returns devices on the specified storage array configured for replication with the specified target array.

SRM recognizes the following device types:

- replication source – read-write device on the source array configured for replication to the target array;
- replication target – read-only replica on the target array identified by a string key;
- demoted source – replication source put in read-only mode in preparation for failover;
- promoted target – read-write device on the target array created from replication target during failover;

The following information is expected for each replication source:

- device ID unique within the scope of an array, used by SRM to execute commands like queryDeviceDetails and syncOnce;
- optional user-friendly name of the device;
- an identification, used by SRM to match the replicated device to a device attached to an ESX host;
- a unique key identifying the replication target, used by SRM to execute commands like testFailoverStart, testFailoverStop, and failover;

Device identification may not be present if device is not presented to any ESX host.

The following information is expected for each replication target:

- a unique key identifying the target;
- optional user-friendly name of the target;
- optional snapshot created by testFailoverStart

The following information is expected for each demoted source:

- a unique device ID;
- optional user-friendly name of the device;
- a unique key identifying the replication target;

Demoted source doesn't contain device identification because the device is not expected to be visible to any ESX host.

Replication target is identified by a key which could be different from the ID of the promoted device after failover. The promoted device should contain both key and ID of the promoted device. Once replication is reversed, the promoted target becomes a replication source, identified by device ID only.

The following information is expected for each promoted target:

- a unique key identifying the target;
- optional user-friendly name of the target;
- a unique device ID of the device created during failover;
- an identification, used by SRM to match the promoted device to a device attached to an ESX host.

Device identification may not be present if device is not presented to any ESX host.

SRM supports the following types of device identifications for FC and iSCSI devices:

- World Wide Name (WWN);
- Logical unit numbers.

SRM supports the following device identification for NFS devices:

- NFS mount path.

SRM supports NFS and VMFS datastore identification by MoId as well.

SRM supports consistency groups.

All devices in a consistency group must have the same type as consistency group itself. The only exception is that consistency group reported as replication source may have a mix of replication source and demoted source devices.

SRM supports bi-directional protection. In this case discoverDevices returns both the source and target devices.

Format of the discoverDevices request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>discoverDevices</Name>
    <Connections …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <DiscoverDevicesParameters>
        <ArrayId>10000A0</ArrayId>
        <PeerArrayId>10000B1</PeerArrayId>
    </DiscoverDevicesParameters>
</Command>
```

Format of the discoverDevices response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
   <ReplicatedDevices>

       <!-- Consistency groups -->
       <ConsistencyGroups>
           <ConsistencyGroup id=”Foo”>
               <Name>DB LUNs</Name>
               <SourceDevices …/>
           </ConsistencyGroup>
           …
       </ConsistencyGroups>
       <TargetGroups>
           <TargetGroup key=”peer-of-Bar”>
               <Name>Replicas of Mail Server LUNs</Name>
               <TargetDevices …/>
           </TargetGroup>
           …
       </TargetGroups>

       <!-- Standalone devices -->
       <SourceDevices>
           <SourceDevice id="a1” …/>
           <SourceDevice id="a2”…/>
           …
       </SourceDevices>
       <TargetDevices>
           <TargetDevice key=”peer-of-b1” …/>
```

```
            <TargetDevice key="peer-of-b2" …/>
            …
        </TargetDevices>
    </ReplicatedDevices>
</Response>
```

In case of replication configured in one direction only, discoverDevices response at the protected site contains replication sources and discoverDevices response at the recovery site contains matching replication targets.

In case of bi-directional replication, the discoverDevices response contains both replication sources and replication targets.

The following description of the format of SourceDevice, TargetDevice, ConsistencyGroup, and TargetGroup tags refers to site A as original protected site and site B as original recovery site.

SourceDevice and TargetDevice tags describe a standalone device. SourceDevice tag describes replication source, TargetDevice tag describes replication target.

- Protected site array contains replicated device X.

  discoverDevices at site A returns device ID and key of the replication target as following:

```
<SourceDevice id="X" state="read-write">
    <Name>DB LUN</Name>
    <TargetDevice key="peer-of-X" />
    <Identity>
        <Wwn>…00:0A…00:0X…</Wwn>
    </Identity>
</SourceDevice>
```

  discoverDevices at site B returns a replication target with matching key:

```
<TargetDevice key="peer-of-X">
    <Name>Replica of DB LUN</Name>
</TargetDevice>
```

  SRM will use "X" to reference the device in queryDeviceDetails, syncOnce (assuming it is configured to be executed at the replication source), and prepareFailover requests as follows:

```
<SourceDevices>
    <SourceDevice id="X" />
</SourceDevices>
```

  SRM will use "peer-of-X" to reference the replication target in queryRecoveryPoints, checkTestFailoverStart, testFailoverStart, testFailoverStop, checkFailover, and failover requests as follows:

```
<TargetDevices>
    <TargetDevice key="peer-of-X" />
</TargetDevices>
```

- After testFailoverStart, discoverDevices at site B returns information for the snapshot of the replication target as follows:

```
<TargetDevice key=”peer-of-X”>
    <Name>Replica of DB LUN</Name>
    <Snapshot id=”X-snap”>
        <Identity>
            <Wwn>…00:0B…01:0X…</Wwn>
        </Identity>
        <RecoveryPoint id="latest">
            <Time>2009-07-10T02:10:00.0</Time>
        </RecoveryPoint>
    </Snapshot>
</TargetDevice>
```

- After prepareFailover, discoverDevices at site A returns device state as “read-only”:

```
<SourceDevice id=”X” state=”read-only”>
    <Name>DB LUN</Name>
    <TargetDevice key=”peer-of-X” />
</SourceDevice>
```

- After failover, discoverDevices at site B returns information for the promoted device as follows:

```
<TargetDevice key=”peer-of-X” id=”X-copy” state=”read-write”>
    <Name>Replica of DB LUN</Name>
    <Identity>
        <Wwn>…00:0B…00:0X…</Wwn>
    </Identity>
</TargetDevice>
```

SRM will use “X-copy” to reference the promoted device in reverseReplication requests.

- After reverseReplication,

discoverDevices at site A returns:

```
<TargetDevice key=”peer-of-X-copy”>
    <Name>DB LUN</Name>
</TargetDevice>
```

discoverDevices at site B returns:

```
<SourceDevice id=”X-copy” state=”read-write”>
    <Name>Replica of DB LUN</Name>
    <TargetDevice key=”peer-of-X-copy” />
    <Identity>
        <Wwn>…00:0B…00:0X…</Wwn>
    </Identity>
</TargetDevice>
```

ConsistencyGroup tag describes a consistency group of devices configured as replication sources. TargetGroup tag describes a replication target for a group of devices.

- Protected site array contains a consistency group Foo made of two devices X and Y.

  discoverDevices at site A returns:

```
<ConsistencyGroup id="Foo" state="read-write">
    <Name>DB LUNs</Name>
    <TargetGroup key="peer-of-Foo" />
    <SourceDevices>
        <SourceDevice id="X">
            <Name>System Tablespace</Name>
            <TargetDevice key="peer-of-X" />
            <Identity>
                <Wwn>…00:0A…00:0X…</Wwn>
            </Identity>
        </SourceDevice>
        <SourceDevice id="Y">
            <Name>Users Tablespace</Name>
            <TargetDevice key="peer-of-Y" />
            <Identity>
                <Wwn>…00:0A…00:0Y…</Wwn>
            </Identity>
        </SourceDevice>
    </SourceDevices>
</ConsistencyGroup>
```

  discoverDevices at site B returns:

```
<TargetGroup key="peer-of-Foo">
    <Name>Replicas of DB LUNs</Name>
    <TargetDevices>
        <TargetDevice key="peer-of-X">
            <Name>Replica of System Tablespace</Name>
        </TargetDevice>
        <TargetDevice key="peer-of-Y">
            <Name>Replica of Users Tablespace</Name>
        </TargetDevice>
    </TargetDevices>
</TargetGroup>
```

SRM will use "Foo" to reference the consistency group in syncOnce (assuming it is configured to be executed at the replication source) and prepareFailover requests as follows:

```
<ConsistencyGroups>
    <ConsistencyGroup id="Foo" />
</ConsistencyGroups>
```

SRM will use "peer-of-Foo" to reference the consistency group in queryRecoveryPoints, checkTestFailoverStart, testFailoverStart, testFailoverStop, checkFailover, and failover requests as follows:

```
<TargetGroups>
    <TargetGroup key="peer-of-Foo" />
</TargetGroups>
```

- After testFailoverStart discoverDevices's response at site B changes to:

```
<TargetGroup key=”peer-of-Foo”>
    <Name>Replicas of DB LUNs</Name>
    <TargetDevices>
        <TargetDevice key=”peer-of-X”>
            <Name>Replica of System Tablespace</Name>
        </TargetDevice>
        <TargetDevice key=”peer-of-Y”>
            <Name>Replica of Users Tablespace</Name>
        </TargetDevice>
    </TargetDevices>
    <Snapshot id=”Foo-snap”>
        <TargetDevices>
            <TargetDevice key=”peer-of-X”>
                <Snapshot id=”X-snap”>
                    <Identity>
                        <Wwn>…00:0B…01:0X…</Wwn>
                    </Identity>
                </Snapshot>
            </TargetDevice>
            <TargetDevice key=”peer-of-Y”>
                <Snapshot id=”Y-snap”>
                    <Identity>
                        <Wwn>…00:0B…01:0Y…</Wwn>
                    </Identity>
                </Snapshot>
            </TargetDevice>
        </TargetDevices>
        <RecoveryPoint id="snap-a2">
            <Name>Hourly</Name>
            <Time>2009-07-10T04:00:00.0</Time>
            <VmImageConsistencyEnabled />
        </RecoveryPoint>
    </Snapshot>
</TargetGroup>
```

- After prepareFailover, discoverDevices’s reponse at site A changes to:

```
<ConsistencyGroup id=”Foo” state=”read-only”>
    <Name>DB LUNs</Name>
    <Target key=”peer-of-Foo” />
    <SourceDevices>
        <SourceDevice id=”X” state=”read-only”>
            <Name>System Tablespace</Name>
            <TargetDevice key=”peer-of-X” />
        </SourceDevice>
        <SourceDevice id=”Y” state=”read-only”>
            <Name>Users Tablespace</Name>
            <TargetDevice key=”peer-of-Y” />
        </SourceDevice>
    </SourceDevices>
</ConsistencyGroup>
```

- After failover, discoverDevices’s response at the site B changes to:

```
<TargetGroup key=”peer-of-Foo” id=”Foo-copy” state=”read-
```

```
write”>
    <Name>Replicas of DB LUNs</Name>
    <TargetDevices>
        <TargetDevice key=”peer-of-X” id=”X-copy” state=”read-
        write”>
            <Name>Replica of System Tablespace</Name>
            <Identity>
                <Wwn>…00:0B…00:0X…</Wwn>
            </Identity>
        </TargetDevice>
        <TargetDevice key=”peer-of-Y” id=”Y-copy” state=”read-
    write”>
            <Name>Replica of Users Tablespace</Name>
            <Identity>
                <Wwn>…00:0B…00:0Y…</Wwn>
            </Identity>
        </TargetDevice>
    </TargetDevices>
</TargetGroup>
```

- After reverseReplication,

  discoverDevices at site A returns:

```
<TargetGroup key=”peer-of-Foo-copy”>
    <Name>DB LUNs</Name>
    <TargetDevices>
        <TargetDevice key=”peer-of-X-copy”>
            <Name>System Tablespace</Name>
        </TargetDevice>
        <TargetDevice key=”peer-of-Y-copy”>
            <Name>Users Tablespace</Name>
        </TargetDevice>
    </TargetDevices>
</TargetGroup>
```

  discoverDevices at site B returns:

```
<ConsistencyGroup id=”Foo-copy” state=”read-write”>
    <Name>Replicas of DB LUNs</Name>
    <SourceDevices>
        <SourceDevice id=”X-copy” state=”read-write”>
            <Name>Replica of System Tablespace</Name>
            <TargetDevice key=”peer-of-X-copy” />
            <Identity>
                <Wwn>…00:0B…00:0X…</Wwn>
            </Identity>
        </SourceDevice>
        <SourceDevice id=”Y-copy” state=”read-write”>
            <Name>Replica of Users Tablespace</Name>
            <TargetDevice key=”peer-of-Y-copy” />
            <Identity>
                <Wwn>…00:0B…00:0Y…</Wwn>
            </Identity>
        </SourceDevice>
    </SourceDevices>
</ConsistencyGroup>
```

Identity tag describes how device can be identified (this information is used by SRM to match the device to a datastore or an RDM).

Table 7 Device Identification Format

Device Identification Type	Format of the Identity tag
(FC, iSCSI) By logical unit number only • Logical unit number is unique within the array; • Each device is presented to any initiator with the same logical unit number;	<Identity> <Lun>1</Lun> </Identity> ... <Identity> <Lun>2</Lun> </Identity> ... <StoragePorts> <StoragePort id="Z1:Z2:Z3:Z4:Z5:Z6:Z7:01" type="FC"/> <StoragePort id="Z1:Z2:Z3:Z4:Z5:Z6:Z7:02" type="FC"/> </StoragePorts> StoragePorts tag contains a list of SCSI targets on the storage array. This information is required.
(FC, iSCSI) By logical unit number per initiator group • Logical unit number is not unique within the array; • Same device can be presented to different initiators with different logical unit numbers; • Different devices can be presented to different initiators with the same logical unit number;	<Identity> <Lun initiatorGroupId="group-a">1</Lun> </Identity> ... <Identity> <Lun initiatorGroupId="group-b">2</Lun> <Lun initiatorGroupId="group-a">2</Lun> </Identity> ... <InitiatorGroups> <InitiatorGroup id="group-a"> <Initiator type="FC" id="21:00:00:E0…"/> <Initiator …/> </InitiatorGroup> <InitiatorGroup id="group-b" …/> </InitiatorGroups> <StoragePorts> <StoragePort id="iqn.z1..." type="iSCSI"/> <StoragePort id="iqn.z2..." type="iSCSI"/> </StoragePorts> StoragePorts tag contains a list of SCSI targets on the storage array. This information is required. InitiatorGroups tag contains a list of initiator groups. This information is required.
(FC, iSCSI) By a pair of logical unit number and target • Logical unit number is unique within the target; • Each device is presented with same logical unit number to	<Identity> <Lun target="iqn.aaa">0</Lun> </Identity> ... <Identity> <Lun target="iqn.bbb">0</Lun> </Identity>

different initiators	
(FC, iSCSI) By logical unit number and target per initiator group	<Identity> <Lun initiatorGroupId="group-a" target="iqn.aaa">0</Lun> </Identity> ... <Identity> <Lun initiatorGroupId="group-b" target="iqn.aaa">0</Lun> </Identity> ... <InitiatorGroups> <InitiatorGroup id="group-a"> <Initiator type="FC" id="21:00:00:E0…"/> <Initiator …/> </InitiatorGroup> <InitiatorGroup id="group-b" …/> </InitiatorGroups> InitiatorGroups tag contains a list of initiator groups. This information is required.
(FC, iSCSI) By World Wide Name	<Identity> <Wwn>A1:A2:A3:A4:A5:A6:A7:A8:A9:A0:A1:A2:A3:A4:A5:A6</Wwn> </Identity>
(NFS)	<Identity> <NfsName>/vol/xxx-1</NfsName> </Identity> ... <Identity> <NfsName>/vol/xxx-2</NfsName> </Identity> ... <StoragePorts> <StoragePort id="10.20.30.40" type="NFS"/> </StoragePorts> StoragePorts tag contains a list of IP addresses of the NAS server. This information is required.
(Other) By datastore MoId	<Identity> <DatastoreMoId>datastore-120</DatastoreMoId> </Identity>

Note: If an SRA identifies devices by logical unit number or by logical unit number and initiator group, it must return a list of array storage ports as well. Storage ports must be identified by World Wide Port Names. SRM uses storage ports to identify devices when several devices have the same logical unit number but are on different arrays.

Table 8 discoverDevices format

Element Name	Content	Description
Command	Name, OutputFile, StatusFile, LogLevel, LogDirectory, Connections, DiscoverDevicesParameters	
Name		Command name, “discoverDevices”.

DiscoverDevicesParameters	ArrayId, PeerArrayId	
ArrayId		ID of the local storage array.
PeerArrayId		ID of the remote storage array.
Response	(ReplicatedDevices, Warnings?) | Error	
ReplicatedDevices	ConsistencyGroups?, TargetGroups?, SourceDevices?, TargetDevices?, InitiatorGroups?, StoragePorts?	Represents devices participating in replication either individually or as part of a consistency group.
ConsistencyGroups	ConsistencyGroup+	
ConsistencyGroup	Name?, TargetGroup, SourceDevices Attributes: id, state	Represents a consistency group of devices configured as replication source. SRM will ensure that devices in the same consistency group are tested and failed over together. Value of id attribute is a unique ID of the group configured as replication source or promoted target. Value of state attribute can be either “read-write” or “read-only”.
Name		Optional. User-friendly name of the group. If not specified the group ID will be displayed to the user.
TargetGroup	Attributes: key	Value of key attribute is a unique key of the replication target for that group.
SourceDevices	SourceDevice+	
SourceDevice	Name?, Identity?, TargetDevice Attributes: id, state	Represents a device configured as replication source. Value of id attribute is a unique ID of the device. Value of state attribute can be either “read-write” or “read-only”.
Name		Optional. User-friendly name of the device. If not specified then device ID will be displayed to the user.
Identity	Wwn | Lun+ | NfsName	Optional. May not be present if device is not presented to any ESX host.

Wwn		World Wide Name of the device (FC and iSCSI only)
Lun	Attributes: target (optional), initiatorGroupId (optional)	Logical Unit Number of the device (FC and iSCSI only). Value of target attribute is: • FC - World Wide Name of the storage port formatted as 8 pairs of hexadecimal digits separated with colons. • iSCSI – iSCSI qualified name (IQN) defined in RFC 3721 of the target Value of initiatorGroupId must point to a group of initiators defined by InitiatorGroup tag in the same response. If target attribute is not specified, then StoragePorts element must be defined. If initiatorGroupId attribute is not specified then SRA is required to ensure that the device is presented with same logical unit number to different initiators.
NfsName		Mount point of NFS volume (NFS only)
Datastore MoId		Datastore Managed Object Identifier (MoId).
TargetDevice	Attributes: key	Represents a replication target on remote array. Value of key attribute must be unique.
TargetGroups	TargetGroup+	
TargetGroup	Name?, TargetDevices, Snapshot? Attributes: key, id (optional), state (optional)	Represents a replication target for a group of devices. Value of key attribute is a unique key of the target. Attributes id and state apply to promoted target only. Value of id attribute is a unique ID of the promoted consistency group. Value of state attribute is “read-write”. Promoted consistency group must have both id and state attributes specified.
TargetDevices	TargetDevice+	

TargetDevice	Name?, Identity? Attributes: key, id (optional), state (optional)	Represents a replication target on local array. Identity tag applies to promoted target only and describes promoted target device. Value of key attribute must be unique. Attributes id and state apply to promoted target only. Value of id attribute is a unique ID of the promoted device. Value of state attribute is “read-write”. Promoted target device must have both id and state attributes specified.
Name		Optional. User-friendly name of the replication target. If not specified then target key will be displayed to the user.
Snapshot	TargetDevices, RecoveryPoint Attributes: id	Represents a snapshot of the group created by testFailoverStart command. Value of ID attribute is the unique ID of the group created during testFailoverStart.
TargetDevices	TargetDevice+	
TargetDevice	Snapshot Attributes: key	
Snapshot	Identity? Attributes: id	Represents a device holding a snapshot of the replica. Value of id attribute is a unique ID of the device.

InitiatorGroups	InitiatorGroup+	Optional. This information is optional if devices are identified by World Wide Names or logical unit number combined with device target uniquely identifies the device. This tag is not supported for NFS. Represents groups of FC and iSCSI initiators with the following property. Devices are presented with the same logical unit numbers to all the initiators in the same group.
InitiatorGroup	Initiator+ Attributes: id	Value of id attribute must be unique within the scope of the array.
Initiator	Attributes: id, type	Value of id attribute is: • FC - World Wide Name of the initiator formatted as 8 pairs of hexadecimal digits separated with colons; • iSCSI – iSCSI qualified name (IQN) defined in RFC 3721 of the initiator. Valid values of type attribute are “FC” and “iSCSI”.
StoragePorts	StoragePort+	Optional. This information is required in case of NFS. It is also required in case of iSCSI and FC if device identification is “initiator+number” or “number”. In case of FC and iSCSI, this element specifies targets on the storage array. In case of NFS, this element specifies IP addresses of the remote host.
StoragePort	Attributes: id, type	Value of id attribute is: • FC - World Wide Name of the storage port formatted as 8 pairs of hexadecimal digits separated with colons; • iSCSI – iSCSI qualified name (IQN) defined in RFC 3721 of the target; • NFS – IP address of the remote host. Valid values of type attribute are “FC”, “iSCSI” and “NFS”.

queryReplicationSettings

Returns the replication settings which need to be preserved after failover when replication is reversed or restored such as replication schedule and settings for VM snapshots taken before replication for specified devices and consistency groups.

queryReplicationSettings is designed to allow SRM to query replication settings for replicated devices before failover and pass these setting to SRA after failover to ensure that replication settings are preserved when replication direction is reversed or restored. SRM passes this infomation as-is to the reverseReplication and restoreReplication commands. SRM issues a queryReplicationSettings command periodically and stores the results in its database. Because this information

may become obsolete, SRA should use it only as a hint when processing reverseReplication and restoreReplication commands.

It is strongly recommended that SRAs implement queryReplicationSettings. This helps avoid manually setting up replication settings after the failover. It is also recommended that SRA log the replication settings to the log file. SRA must issue a warning when the replication is not fully setup either during reverseReplication or restoreReplication commands.
Array replication architecture dictates whether the replication settings information is available at the replication source or at the replication target. By default SRM assumes that this information is available at the source, but an SRA can specify a different model in the queryCapabilities command results.

Format of the queryReplicationSettings request when initiated at replication source:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>queryReplicationSettings</Name>
    <Connections …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <QueryReplicationSettingsParameters>
        <ArrayId>XXX</ArrayId>
        <PeerArrayId>YYY</PeerArrayId>
        <ConsistencyGroups>
            <ConsistencyGroup id="srm-1" />
        </ConsistencyGroups>
        <SourceDevices>
            <SourceDevice id="xxx-2"/>
            <SourceDevice id="xxx-3"/>
        </SourceDevices>
    </QueryReplicationSettingsParameters>
</Command>
```

Format of the queryReplicationSettings response at replication source:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    <ReplicationSettings>
        <ConsistencyGroups>
            <ConsistencyGroup id="srm-1">
                <ReplicationSettings .../>
            </ConsistencyGroup>
        </ConsistencyGroups>
        <SourceDevices>
            <SourceDevice id="xxx-2">
                <ReplicationSettings .../>
            </SourceDevice>
            <SourceDevice id="xxx-3">
                <ReplicationSettings .../>
            </SourceDevice>
        </Devices>
    </ReplicationSettings>
</Response>
```

Format of the queryReplicationSettings request when initiated at replication target:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
```

```
    <Name>replicationSettings</Name>
    <Connection …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <QueryReplicationSettingsParameters>
        <ArrayId>YYY</ArrayId>
        <PeerArrayId>XXX</PeerArrayId>
        <TargetGroups>
            <TargetGroup key="srm-target-1" />
        </TargetGroups>
        <TargetDevices>
            <TargetDevice key="xxx-2-peer"/>
            <TargetDevice key="xxx-3-peer"/>
        </TargetDevices>
    </QueryReplicationSettingsParameters>
</Command>
```

Format of the queryReplicationSettings response at replication target:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    <ReplicationSettings>
        <TargetGroups>
            <TargetGroup id="srm-target-1">
                <ReplicationSettings .../>
            </TargetGroup>
        </TargetGroups>
        <TargetDevices>
            <TargetDevice key="xxx-2-peer">
                <ReplicationSettings .../>
            </TargetDevice>
            <TargetDevice key="xxx-3-peer">
                <ReplicationSettings .../>
            </TargetDevice>
        </TargetDevices>
    </ReplicationSettings>
</Response>
```

Element Name	Content	Description
Command	Name, OutputFile, StatusFile, LogLevel, LogDirectory, Connections, QueryReplicationSettingsParameters	
Name		Command name, “queryReplicationSettings”.
QueryReplicationSettingsParameters	ArrayId, PeerArrayId, (ConsistencyGroups?, SourceDevices?) | (TargetGroups?,	

	TargetDevices)	
ArrayId		ID of the local storage array.
PeerArrayId		ID of the remote storage array.
ConsistencyGroups	ConsistencyGroup+	
ConsistencyGroup	Attributes: id	Value of id attribute is consistency group ID returned by discoverDevices command.
TargetGroups	TargetGroup+	
TargetGroup	Attributes: key	
SourceDevices	SourceDevice+	
SourceDevice	Attributes: id	Value of id attribute is device ID returned by discoverDevices command.
TargetDevices	TargetDevice+	
TargetDevice	Attributes: key	Value of key attribute is device key returned by discoverDevices command.
Response	(ReplicationSettings, Warnings?) | Error	
ReplicationSettings	(ConsistencyGroups?, SourceDevices?) | (TargetGroups?, TargetDevices)	
ConsistencyGroups	ConsistencyGroup+	
ConsistencyGroup	(ReplicationSettings, Warnings?) | Error Attributes: id	
TargetGroups	TargetGroup+	
TargetGroup	(ReplicationSettings, Warnings?) | Error	

	Attributes: key	
SourceDevices	SourceDevice+	
SourceDevice	(ReplicationSettings, Warnings?) | Error Attributes: id	
TargetDevices	TargetDevice+	
TargetDevice	(ReplicationSettings, Warnings?) | Error Attributes: key	
ReplicationSettings		Format of this element is not defined and is specific to the SRA.

queryDeviceDetails[*]

Returns detailed information about a specified set of devices.

Format of the queryDeviceDetails request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>queryDeviceDetails</Name>
    <Connections …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <QueryDeviceDetailsParameters>
        <ArrayId>XXX</ArrayId>
        <Devices>
            <Device id="aaa-1" />
            <Device id="aaa-2" />
            <Device id="aaa-3" />
        </Devices>
    </QueryDeviceDetailsParameters>
</Command>
```

Format of the queryDeviceDetails response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    <DeviceDetails>
        <Device id="aaa-1">
            <CustomProperties>
                <CustomProperty name="…" value="…" />
                <CustomProperty name="…" value="…" />
                …
            </CustomProperties>
        </Device>
```

[*] Not supported by the Glenlivet release of VMware vCenter Site Recovery Manager.

```
        <Device id="aaa-2">
            <CustomProperties>
                <CustomProperty name="…" value="…" />
            </CustomProperties>
        </Device>
        <Device id="aaa-3">
            <CustomProperties>
                <CustomProperty name="…" value="…" />
            </CustomProperties>
        </Device>
    </DeviceDetails>
</Response>
```

Table 9 queryDeviceDetails format

Element Name	Content	Description
Command	Name, OutputFile, StatusFile, LogLevel, LogDirectory, Connections, QueryDeviceDetailsParameters	
Name		Command name, "queryDeviceDetails".
QueryDeviceDetailsParameters	ArrayId, Devices	
ArrayId		ID of the local storage array.
Devices	Device+	
Device	Attributes: id	Value of id attribute is device ID returned by discoverDevices command.
Response	(DeviceDetails, Warnings?) | Error	
DeviceDetails	Devices	
Devices	Device+	
Device	(CustomProperties, Warnings?) | Error Attributes: id	
CustomProperties	CustomProperty+	List of device properties presented as name-value pairs.
CustomProperty	Attributes: name, value	Device property presented as name-value pair.

queryRecoveryPoints[*]

Returns a list of available recovery points for specified devices and consistency groups.

This functionality is optional. If supported, queryRecoveryPoints command must be included in queryCapabilities response.

[*] Not supported by the Glenlivet release of VMware vCenter Site Recovery Manager.

Recovery point is described by a unique identifier, timestamp of the replica data, and optional user-friendly name (bookmark).

```
    <RecoveryPoint id="snap-a3">
        <Name>Nightly</Name>
        <Time>2009-07-10T03:30:00.0</Time>
    </RecoveryPoint>
```

Note: Timestamps must be in standard format described at http://www.w3.org/TR/xmlschema-2/#dateTime .

If SRA supports replication of quiesced virtual machine images, it can identify recovery points which contain quiesced images with VmImageConsistencyEnabled tag.

```
    <RecoveryPoint id="snap-a3">
        <Name>Nightly</Name>
        <Time>2009-07-10T03:30:00.0</Time>
        <VmImageConsistencyEnabled />
    </RecoveryPoint>
```

The recovery point ID is used to specify the recovery point in testFailoverStart and failover requests.

Recovery points are reported separately for each consistency group and each standalone device.

Format of the queryRecoveryPoints request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>queryRecoveryPoints</Name>
    <Connections …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <QueryRecoveryPointsParameters>
        <ArrayId>YYY</ArrayId>
        <TargetGroups>
            <TargetGroup key="srm-target-1" />
        </TargetGroups>
        <TargetDevices>
            <TargetDevice key="xxx-2-peer" />
            <TargetDevice key="xxx-3-peer" />
        </TargetDevices>
    </QueryRecoveryPointsParameters>
</Command>
```

Format of the queryRecoveryPoints response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    <QueryRecoveryPointsResults>
        <TargetGroups>
            <TargetGroup key="srm-target-1">
                <RecoveryPoints>
                    <RecoveryPoint id="snap-a3">
                        <Name>Nightly</Name>
                        <Time>2009-07-10T03:30:00.0</Time>
                    </RecoveryPoint>
                    <RecoveryPoint id="snap-a2">
                        <Name>Hourly</Name>
```

```
                    <Time>2009-07-10T02:00:00.0</Time>
                    <VmImageConsistencyEnabled />
                </RecoveryPoint>
                <RecoveryPoint id="snap-a1">
                    <Name>Hourly</Name>
                    <Time>2009-07-10T01:00:00.0</Time>
                </RecoveryPoint>
            </RecoveryPoints>
        </TargetGroup>
    </TargetGroups>
    <TargetDevices>
        <TargetDevice key="xxx-2-peer">
            <RecoveryPoints>
                <RecoveryPoint id="snap-b2">
                    <Name>Hourly</Name>
                    <Time>2009-07-10T02:05:00.0</Time>
                </RecoveryPoint>
                <RecoveryPoint id="snap-b1">
                    <Name>Hourly</Name>
                    <Time>2009-07-10T01:05:00.0</Time>
                </RecoveryPoint>
            </RecoveryPoints>
        </TargetDevice>
        <TargetDevice key="xxx-3-peer">
            <Error>
                <Description>Device 'xxx-3-peer' is not ready yet.</Description>
                <FixHint>Wait for initial replication to complete and retry.</FixHint>
            </Error>
        </TargetDevice>
    </TargetDevices>
</QueryRecoveryPointsResults>
</Response>
```

Table 10 queryRecoveryPoints format

Element Name	Content	Description
Command	Name, OutputFile, StatusFile, LogLevel, LogDirectory, Connections, QueryRecoveryPointsParameters	
Name		Command name, “queryRecoveryPoints”.
QueryRecoveryPointsParameters	ArrayId, TargetGroups?, TargetDevices?	
ArrayId		ID of the local storage array.
TargetGroups	TargetGroup+	
TargetGroup	Attributes: key	Value of key attribute is consistency group key returned by discoverDevices command.
TargetDevices	TargetDevice+	
TargetDevice	Attributes: key	Value of key attribute is device key returned by discoverDevices command.

Response	(QueryRecoveryPointsResults, Warnings?) | Error	
QueryRecoveryPointsResults	TargetGroups?, TargetDevices?	
TargetGroups	TargetGroup+	
TargetGroup	(RecoveryPoints, Warnings?) | Error Attributes: key	
RecoveryPoints	RecoveryPoint+	List of available recovery points for a consistency group or a single device.
RecoveryPoint	Name?, Time, VmImageConsistencyEnabled? Attributes: id	Represents a snapshot of the production data available for recovery. Value of id attribute is the unique identifier of the recovery point. This identifier will be used to specify the desired recovery point in testFailoverStart and failover requests.
Name		Optional. User-friendly name of the recovery point. If not specified then only timestamp of the snapshot will be displayed to the user.
Time		Timestamp of the snapshot. Format: YYYY-MM-DDThh:mm:ss.zzzzzz, where • YYYY indicates the year • MM indicates the month • DD indicates the day • T indicates the start of the required time section • hh indicates the hour • mm indicates the minute • ss indicates the second • zzzzzz (if present) indicates the timezone (More details can be found at http://www.w3.org/TR/xmlschema-2/#dateTime .)
VmImage ConsistencyEnabled		Optional. If specified indicated that this snapshot contains VMs with

		snapshots taken just before replication. Requires VmImageConsistency reported in queryCapabilities response.
TargetDevices	TargetDevice+	
TargetDevice	(RecoveryPoints, Warnings?) | Error Attributes: key	

checkTestFailoverStart[*] and checkFailover[*]

- checkTestFailoverStart validates environment and target devices before test failover.
- checkFailover validates environment and target devices before failover.

checkTestFailoverStart and checkFailover commands are expected to perform various configuration and runtime state checks on the specified target devices and identify problems which will prevent successful test or failover. A complete set of checks is defined by the SRA but the following checks are expected to be implemented by every SRA.

- Initial data synchronization has completed and there is a copy of the production data ready for test or failover.
- There is a valid license to perform test or failover operation.

The granularity of checks and results is consistency group or device if SRA doesn't support consistency groups. Individual result can be either success, success with warnings, or an error.

Format of a checkTestFailoverStart request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>checkTestFailoverStart</Name>
    <Connections …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <CheckTestFailoverStartParameters>
        <ArrayId>10000A0</ArrayId>
        <TargetGroups>
            <TargetGroup key="srm-target-1" />
        </TargetGroups>
        <TargetDevices>
            <TargetDevice key="xxx-2-peer" />
            <TargetDevice key="xxx-3-peer" />
        </TargetDevices>
    </CheckTestFailoverStartParameters>
</Command>
```

Format of a checkTestFailoverStart response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
   <CheckTestFailoverStartResults>
        <TargetGroups>
```

[*] Not supported by the Glenlivet release of VMware vCenter Site Recovery Manager.

```
            <TargetGroup key="srm-target-1">
                <Success />
            </TargetGroup>
        </TargetGroups>
        <TargetDevices>
            <TargetDevice key="xxx-2-peer">
                <Error …/>
            </TargetDevice>
            <TargetDevice key="xxx-3-peer">
                <Success />
                <Warnings …/>
            </TargetDevice>
        </TargetDevices>
    </CheckTestFailoverStartResults>
</Response>
```

Note: The response must contain all groups and devices specified in the request.

Format of a checkFailover request is identical to that of a checkTestFailoverStart request except for the command name and parameters tag.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>checkFailover</Name>
    …
    <CheckFailoverParameters>
        …
    </CheckFailoverParameters>
</Command>
```

Format of a checkFailover response is identical to that of a checkTestFailoverStart response except for the results tag,

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    <CheckFailoverResults>
        …
    </CheckFailoverResults>
</Response>
```

testFailoverStart, testFailoverStop and failover

- testFailoverStart creates writeable temporary copies of the requested replication targets and presents these copies to the requested hosts.
- testFailoverStop deletes the temporary copies created by testFailoverStart command.
- failover stops replication for the requested replication targets, makes writable devices from these targets and presents these devices to requested hosts.

Because testFailoverStart and testFailoverStop must support non-disruptive recovery tests, they must not affect replication. (failover, in contrast, must stop replication for the requested devices.)

The request contains a number of standalone devices and consistency groups. Each device and consistency group is identified by a key returned by discoverDevices command.

```
        <TargetGroup key="srm-target-1" />
        ...
        <TargetDevice key="xxx-2-peer" />
```

If an SRA supports recovery to multiple points in time (see queryCapabilities), the request may specify a recovery point for each device and consistency group. Recovery points are identified by IDs returned by queryRecoveryPoints command. If recovery point is not specified, SRA should use the latest available replica data.

```
<TargetGroup key="srm-target-1">
    <RecoveryPoint id="snap-a1" />
</TargetGroup>
...
<TargetDevice key="xxx-3-peer">
    <RecoveryPoint id="snap-c3" />
</TargetDevice>
```

If an SRA supports dynamic access restrictions for testFailoverStart and failover (see queryCapabilities), SRM will specify the set of ESX hosts that require access to failed-over devices. An ESX host is described as an initiator with a type and ID. The following initiator types are defined:

- FC
    - an initiator represents a Fibre Channel HBA on a host;
    - initiator ID is the World Wide Port Name (WWPN) of the HBA;

    ```
    <Initiator type="FC" id="21:00:00:E0:8B:8F:55:4B" />
    ```

- iSCSI
    - an initiator represents a hardware or a software iSCSI HBA on a host;
    - initiator ID is the iSCSI name of the HBA;
    - if CHAP authentication is enabled, the CHAP user name is specified in addition to iSCSI name;

    ```
    <Initiator type="iSCSI"
               id="iqn.2000-04.com.qlogic:qla4052c.gs10744a49800.1" />
    ```

    or

    ```
    <Initiator type="iSCSI"
               id="iqn.2000-04.com.qlogic:qla4052c.gs10744a49800.1"
               chapName="foo"  />
    ```

    when CHAP authentication is enabled

- NFS
    - initiator ID is the VMkernel IP address of the host;

    ```
    <Initiator type="NFS" id="1.2.3.4" />
    ```

- Host (corresponds to supported protocol “Datastore”)
    - an initiator represents a standalone ESX host or a cluster of ESX hosts in the vCenter inventory;
    - initiator ID is the Managed Object Identifier (MoId) of the compute resource;

    ```
    <Initiator type="Host" id="domain-s100" />
    ```

Initiators are organized in groups with unique identifiers called access groups. The exact grouping mechanism is opaque to SRA. In practice each group corresponds to a standalone ESX host or a cluster of ESX hosts.

```
<AccessGroup id="host-group-A">
    <Initiator type="FC" id="20:00:00:E0…"/>
    <Initiator …/>
</AccessGroup>
```

Finally, the request specifies a list of access groups for each device. The list of access groups maps to a list of hosts which needs to have access to the snapshot or failed over device.

```
<TargetDevice key="xxx-2-peer">
    <AccessGroups>
        <AccessGroup id="host-group-B"/>
        <AccessGroup …/>
    </AccessGroups>
</TargetDevice>
```

Access groups are specified separately for each standalone device and each device which is part of a consistency group. It is allowed that different devices within the same consistency group request access to different sets of hosts.

The response from testFailoverStart, testFailoverStop, and failover commands must contain the same set of standalone devices and consistency groups as specified in the request. The results of the operation are reported separately for each standalone device and each consistency group. The response may contain a mix of failures and successful results for different devices and groups.

A failure associated with a particular device or group is returned with an Error tag within the corresponding TargetDevice or TargetGroup tag.

```
<TargetGroup key="srm-target-1">
    <Error …/>
</TargetGroup>

<TargetDevice key="xxx-2-peer">
    <Error …/>
</TargetDevice>
```

Successful testFailoverStart operation returns information about the snapshot device including device identification and recovery point used to create the snapshot.

```
<TargetDevice key="xxx-3-peer">
    <Snapshot id="xxx-3-copy">
        <Identity … />
        <RecoveryPoint id="snap-c1">
            <Name>Hourly</Name>
            <Time>2009-07-10T02:10:00.0</Time>
            <VmImageConsistencyEnabled />
        </RecoveryPoint>
    </Snapshot>
</TargetDevice>
```

If SRA was able to successfully create a snapshot but failed to present snapshot device to any ESX host, SRA must return information about the snapshot, recovery point used to create that snapshot and a warning describing the reason for failure to present snapshot to ESX hosts. In this case reponse doesn't contain device identification.

```
<TargetDevice key="xxx-3-peer">
    <Snapshot id="xxx-3-copy">
        <RecoveryPoint id="snap-c1">
            <Name>Hourly</Name>
            <Time>2009-07-10T02:10:00.0</Time>
            <VmImageConsistencyEnabled />
        </RecoveryPoint>
    </Snapshot>
    <Warnings>
        <Warning …/>
```

```
    </Warnings>
</TargetDevice>
```

Recovery point information containing the timestamp of the snapshot data is required even if SRA doesn't support recovery to multiple points in time (e.g. doesn't support queryRecoveryPoints command). In that case it is acceptable to use some hard-coded recovery point ID, for example "latest", in the response.

Successful testFailoverStop operation returns a simple indication of success without any additional information.

```
<TargetGroup key="srm-target-1">
    <Success />
</TargetGroup>

<TargetDevice key="xxx-2-peer">
    <Success />
</TargetDevice>
```

Successful failover operation returns information about the failed over device including device identification and recovery point used to failover the device.

```
<TargetDevice key="xxx-3-peer" id="xxx-3-copy" state="read-write">
    <Identity … />
    <RecoveryPoint id="snap-c1">
        <Name>Hourly</Name>
        <Time>2009-07-10T02:10:00.0</Time>
        <VmImageConsistencyEnabled />
    </RecoveryPoint>
</TargetDevice>
```

If SRA was able to successfully promote device but failed to present promoted device to any ESX host, SRA must return information about the promoted device, recovery point used to promote the device and a warning describing the reason for failure to present promoted device to ESX hosts. In this case reponse doesn't contain device identification.

```
<TargetDevice key="xxx-3-peer" id="xxx-3-copy" state="read-write">
    <RecoveryPoint id="snap-c1">
        <Name>Hourly</Name>
        <Time>2009-07-10T02:10:00.0</Time>
        <VmImageConsistencyEnabled />
    </RecoveryPoint>
    <Warnings>
        <Warning …/>
    </Warnings>
</TargetDevice>
```

failover command must be idempotent. If specified devices are already failed over, SRA must return success with optional warnings indicating that operation was a no-op. If specified devices are already failed over but not presented to ESX hosts, SRA must present these devices. If SRA supports dynamic access restriction and specified devices are already failed over, SRA must verify and reconfigure, if necessary, access restrictions for these devices.

testFailoverStop command must also be idempotent. SRA must return success with optional warnings indicating that operation was a no-op. SRA must verify that the testFailover snapshot is either deleted or not presented to ESX hosts anymore.

Format of the testFailoverStart request for an SRA that doesn't support dynamic access restriction:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
```

```
    <Name>testFailoverStart</Name>
    <Connections …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <TestFailoverStartParameters>
        <ArrayId>10000A0</ArrayId>
        <TargetGroups>
            <TargetGroup key="srm-target-1">
                <RecoveryPoint id="snap-a1" />
            </TargetGroup>
        </TargetGroups>
        <TargetDevices>
            <TargetDevice key="xxx-2-peer" />
            <TargetDevice key="xxx-3-peer">
                <RecoveryPoint id="snap-c3" />
            </TargetDevice>
        </TargetDevices>
    </TestFailoverStartParameters>
</Command>
```

Format of the testFailoverStart request for an SRA that supports dynamic access restriction:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>testFailoverStart</Name>
    <Connections …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <TestFailoverStartParameters>
        <ArrayId>10000A0</ArrayId>
        <AccessGroups>
            <AccessGroup id="host-group-A">
                <Initiator type="FC" id="20:00:00:E0…"/>
                <Initiator …/>
            </AccessGroup>
            <AccessGroup id="host-group-B">
                <Initiator type="FC" id="20:00:00:E1…"/>
            </AccessGroup>
        </AccessGroups>
        <TargetGroups>
            <TargetGroup key="srm-target-1">
                <RecoveryPoint id="snap-a2" />
                <TargetDevices>
                    <TargetDevice key="xxx-1-peer">
                        <AccessGroups>
                            <AccessGroup id="host-group-A"/>
                            <AccessGroup id="host-group-B"/>
                        </AccessGroups>
                    </TargetDevice>
                </TargetDevices>
            </TargetGroup>
        </TargetGroups>
        <TargetDevices>
            <TargetDevice key="xxx-2-peer">
                <AccessGroups>
                    <AccessGroup id="host-group-B"/>
```

```
            </AccessGroups>
        </TargetDevice>
        <TargetDevice key="xxx-3-peer">
            <AccessGroups>
                <AccessGroup id="host-group-B"/>
            </AccessGroups>
        </TargetDevice>
    </TargetDevices>
</TestFailoverStartParameters>
</Command>
```

The format of a failover request is identical to that of a testFailoverStart request except for the command name and parameters tag.

```
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>failover</Name>
    …
    <FailoverParameters>
        …
```

Format of the testFailoverStop request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>testFailoverStop</Name>
    <Connection …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <TestFailoverStopParameters>
        <ArrayId>10000A0</ArrayId>
        <TargetGroups>
            <TargetGroup key="srm-target-1" />
        </TargetGroups>
        <TargetDevices>
            <TargetDevice key="xxx-2-peer" />
            <TargetDevice key="xxx-3-peer" />
        </TargetDevices>
    </TestFailoverStopParameters>
</Command>
```

Note: the testFailoverStop request format is not affected by an SRA's support of dynamic access restriction.

Format of the testFailoverStart response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    <TestFailoverStartResults>
        <TargetGroups>
            <TargetGroup key="srm-target-1">
                <Snapshot id="srm-target-1-snap">
                    <RecoveryPoint id="snap-a2">
                        <Name>Hourly</Name>
                        <Time>2009-07-10T02:00:00.0</Time>
                        <VmImageConsistencyEnabled />
                    </RecoveryPoint>
                    <TargetDevices>
                        <TargetDevice key="xxx-1-peer">
```

```
                        <Snapshot id="xxx-1-snap">
                            <Identity …/>
                        </Snapshot>
                    </TargetDevice>
                </TargetDevices>
            </Snapshot>
        </TargetGroup>
    </TargetGroups>
    <TargetDevices>
        <TargetDevice key="xxx-2-peer">
            <Error …/>
        </TargetDevice>
        <TargetDevice key="xxx-3-peer">
            <Snapshot id="xxx-3-copy">
                <Idenity … />
                <RecoveryPoint id="snap-c1">
                    <Name>Hourly</Name>
                    <Time>2009-07-10T02:10:00.0</Time>
                    <VmImageConsistencyEnabled />
                </RecoveryPoint>
            </Snapshot>
        </TargetDevice>
    </TargetDevices>
</TestFailoverStartResults>
</Response>
```

The response must contain all groups and devices specified in the request.

Format of the failover response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    <FailoverResults>
        <TargetGroups>
            <TargetGroup key="srm-target-1" id="srm-target-1-copy” state="read-write”>
                <RecoveryPoint id="snap-a2">
                    <Name>Hourly</Name>
                    <Time>2009-07-10T02:00:00.0</Time>
                    <VmImageConsistencyEnabled />
                </RecoveryPoint>
                <TargetDevices>
                    <TargetDevice key="xxx-1-peer" id=”xxx-1-copy” state=”read-write”>
                        <Identity …/>
                     </TargetDevice>
                </TargetDevices>
            </TargetGroup>
        </TargetGroups>
        <TargetDevices>
            <TargetDevice key="xxx-2-peer">
                <Error …/>
            </TargetDevice>
            <TargetDevice key="xxx-3-peer" id=”xxx-3-copy” state=”read-write”>
                <Idenity … />
                <RecoveryPoint id="snap-c1">
                    <Name>Hourly</Name>
                    <Time>2009-07-10T02:10:00.0</Time>
                    <VmImageConsistencyEnabled />
                </RecoveryPoint>
            </TargetDevice>
        </TargetDevices>
```

```
    </FailoverResults>
</Response>
```

The response must contain all groups and devices specified in the request.

Format of the Identity tag is the same as in the discoverDevices command response.

Format of the testFailoverStop response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
   <TestFailoverStopResults>
       <TargetGroups>
           <TargetGroup key="srm-target-1">
               <Success />
           </TargetGroup>
       </TargetGroups>
       <TargetDevices>
           <TargetDevice key="xxx-2-peer">
               <Error …/>
           </TargetDevice>
           <TargetDevice key="xxx-3-peer">
               <Success />
           </TargetDevice>
       </TargetDevices>
   </TestFailoverStopResults>
</Response>
```

The response must contain all groups and devices specified in the request.

Table 11 testFailoverStart and failover format

Element Name	Content	Description
Command	Name, OutputFile, StatusFile, LogLevel, LogDirectory, Connections, (TestFailoverStartParameters | FailoverParameters)	
Name		Command name, “testFailoverStart” or “failo
TestFailoverStartParameters	ArrayId, AccessGroups?, TargetGroups?, TargetDevices?	
FailoverParameters	ArrayId, AccessGroups?, TargetGroups?, TargetDevices?	
ArrayId		ID of the local storage array hosting target de to be tested or failed over.
AccessGroups	AccessGroup+	Optional. Applies only if SRA identified supp for dynamic access restrictions in queryCapabilities command response. Defines initiator groups used to request acces

		test or failed over devices.
AccessGroup	Initiator+ Attributes: id	Value of id attribute contains unique identifie the group.
Initiator	Attributes: type, id, chapName	Valid values of type attribute are “FC”, “iSCS “NFS”, and “Host”. Value of id attribute is: • FC - World Wide Name of the initiat formatted as 8 pairs of hexadecimal d separated with colons; • iSCSI – iSCSI qualified name (IQN) defined in RFC 3721 of the initiator; • NFS – VMkernel IP address of the ES host; • Host – Managed Object Identifier (M of a standalone host or a cluster in vC inventory. Optional chapName attribute contains CHAP name (iSCSI only).
TargetGroups	TargetGroup+	Consistency groups to snapshot or failover.
TargetGroup	RecoveryPoint?, TargetDevices? Attributes: key	Consistency group to failover along with opti recovery point to use and optional access restrictions to set for contained devices. TargetDevices tag is present only if SRA iden support for dynamic access restrictions in queryCapabilities command response.
TargetDevices	TargetDevice+	Access restrictions requested for devices cont in the consistency group.
TargetDevice	AccessGroups Attributes: key	Access restrictions for a single device in a consistency group specified as a list of groups initiators requested to get access to snapshot c failed over device.
TargetDevices	TargetDevice+	
TargetDevice	AccessGroups?, RecoveryPoint? Attributes: key	Value of key attribute is target device key retu by the discoverDevices command.
AccessGroups	AccessGroup+	
AccessGroup	Attributes: id	Optional. Identifies a group of initiators reque to get access to test or failed over device.
RecoveryPoint	Attributes: id	Optional. Identifies a specific recovery point device. Latest recovery point should be used i specified. Value of id attribute matches one of the value

		returned by queryRecoveryPoints.
Response	(TestFailoverStartResults | FailoverResults, Warnings?) | Error	
TestFailoverStartResults	TargetGroups?, TargetDevices?, InitiatorGroups?, StoragePorts?	Represents per-group and per-device results o testFailoverStart command.
TargetGroups	TargetGroup+	Represents per-group results.
TargetGroup	(Snapshot, Warnings?) | Error Attributes: key	Represents results of testFailoverStart for a si consistency group. Attribute key identifies the consistency group
Snapshot	RecoveryPoint, TargetDevices Attributes: id	Represents a read-write copy of a group as a whole. Value of id attribute is unique identifier of the snapshot in the scope of the group. Different groups may have snapshots with the same id.
TargetDevices	TargetDevice+	Represents per-device results.
TargetDevice	Snapshot Attributes: key	Represents results of the testFailoverStart for single device within a consistency group. Attribute key identifies the device.
Snapshot	Identity? Attributes: id	Represents a read-write copy of the target dev Value of id attribute is unique identifier of the snapshot in the scope of the device. Different devices may have snapshots with the same id.
TargetDevices	TargetDevice+	Represents per-device results for standalone devices.
TargetDevice	(Snapshot, Warnings?) | Error Attributes: key	Represents results of the testFailoverStart for single standalone device. Attribute key identifies the device.
Snapshot	Identity?, RecoveryPoint Attributes: id	Represents a read-write copy of the target dev Value of id attribute is unique identifier of the snapshot in the scope of the device. Different devices may have snapshots with the same id.
RecoveryPoint	Name?,	Describes the snapshot of the production data

	Time, VmImageConsistencyEnabled? Attributes: id	to create a read-write copy of the target devic See queryRecoveryPoints
InitiatorGroups		See discoverDevices
StoragePorts		See discoverDevices
FailoverResults	TargetGroups?, TargetDevices?, InitiatorGroups?, StoragePorts?	Represents per-group and per-device results o failover command.
TargetGroups	TargetGroup+	Represents per-group results.
TargetGroup	(RecoveryPoint, TargetDevices, Warnings?) | Error Attributes: key, id (optional), state (optional)	Represents results of failover for a single consistency group. Attribute key identifies the consistency group Value of id attribute is a unique identifier of t promoted group. Value of state attribute can b either “read-write” or “read-only”. Both id an state attributes must be specified if group was successfully promoted.
TargetDevices	TargetDevice+	Represents per-device results.
TargetDevice	Identity? Attributes: key, id (optional), state (optional)	Represents results of failover for a single devi within a consistency group. Attribute key identifies the device. Value of id attribute is a unique identifier of t promoted device. Value of state attribute can either “read-write” or “read-only”. Both id an state attributes must be specified if device wa successfully promoted.
TargetDevices	TargetDevice+	Represents per-device results for standalone devices.
TargetDevice	(Identity?, RecoveryPoint, Warnings?) | Error Attributes: key, id(optional), state(optional)	Represents results of failover for a standalone device. Attribute key identifies the device. Value of id attribute is a unique identifier of t promoted device. Value of state attribute can either “read-write” or “read-only”. Both id an state attributes must be specified if device wa successfully promoted.
InitiatorGroups		See discoverDevices
StoragePorts		See discoverDevices

Table 12 testFailoverStop format

Element Name	Content	Description
Command	Name, OutputFile, StatusFile, LogLevel, LogDirectory, TestFailoverStopParameters	
Name		Command name, "testFailoverStop".
TestFailoverStopParameters	ArrayId, TargetGroups?, TargetDevices?	
ArrayId		ID of the local storage array hosting target devices being tested.
TargetGroups	TargetGroup+	
TargetGroup	Attributes: key	
TargetDevices	TargetDevice+	
TargetDevice	Attributes: key	Value of key attribute is target device key returned by the discoverDevices command.
Response	(TestFailoverStopResults, Warnings?) | Error	
TestFailoverStopResults	TargetGroups?, TargetDevices?	
TargetGroups	TargetGroup+	
TargetGroup	(Success, Warnings?) | Error Attributes: key	
TargetDevices	TargetDevice+	
TargetDevice	(Success, Warnings?) | Error Attributes: key	Value of key attribute must correspond to a target device key specified in the request.

prepareFailover

Makes source devices read-only and takes a snapshot of these devices in anticipation of a failover.

This command must be idempotent. SRM may execute prepareFailover command several times in succession on the same set of devices and groups. SRA must expect this scenario and respond with optional warnings indicating that the command succeeded but was a no-op as applicable.

In Planned Migration scenario SRM executes prepareFailover command at protected site before executing failover command at recovery site. However, in Disaster Recovery scenario the order of execution of failover

and prepareFailover commands is reversed. While protected site is down SRM executes failover command at recovery site. Later, when protected site comes back up, SRM executes prepareFailover command at protected site. SRA must ensure that prepareFailover command succeeds in this scenario with optional warnings indicating that command was executed in an unusual scenario.

Format of the prepareFailover request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>prepareFailover</Name>
    <Connections …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <PrepareFailoverParameters>
        <ArrayId>10000A0</ArrayId>
        <ConsistencyGroups>
            <ConsistencyGroup id="srm-1" />
        </ConsistencyGroups>
        <SourceDevices>
            <SourceDevice id="xxx-2" />
            <SourceDevice id="xxx-3" />
        </SourceDevices>
    </PrepareFailoverParameters>
</Command>
```

Format of the prepareFailover response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    <PrepareFailoverResults>
        <ConsistencyGroups>
            <ConsistencyGroup id="srm-1">
                <Success />
            </ConsistencyGroup>
        </ConsistencyGroups>
        <SourceDevices>
            <SourceDevice id="xxx-2 ">
                <Success />
            </SourceDevice>
            <SourceDevice id="xxx-3 ">
                <Error …/>
            </SourceDevice>
        </SourceDevices>
    </PrepareFailoverResults>
</Response>
```

Note: The response must contain all groups and devices specified in the request.

Table 13 prepareFailover format

Element Name	Content	Description
Command	Name, OutputFile, StatusFile, LogLevel, LogDirectory, Connections, PrepareFailoverParameters	

Name		Command name, “prepareFailover”
PrepareFailoverParameters	ArrayId, ConsistencyGroups?, SourceDevices?	
ArrayId		ID of the local storage array.
ConsistencyGroups	ConsistencyGroup+	
ConsistencyGroup	Attributes: id	
SourceDevices	SourceDevice+	Replicated read-write devices to be made read-only.
SourceDevice	Attributes: id	Value of id attribute is device ID returned by discoverDevices command.
Response	(PrepareFailoverResults, Warnings?) | Error	
PrepareFailoverResults	ConsistencyGroups?, SourceDevices?	
ConsistencyGroups	ConsistencyGroup+	
ConsistencyGroup	(Success, Warnings?) | Error Attributes: id	Value of id attribute must correspond to a consistency group ID specified in the request.
SourceDevices	SourceDevice+	
SourceDevice	(Success, Warnings?) | Error Attributes: id	Value of id attribute must correspond to a device ID specified in the request.

syncOnce

Requests immediate replication of the specified devices and consistency groups.

syncOnce must return as soon as possible, without waiting for replication to complete.

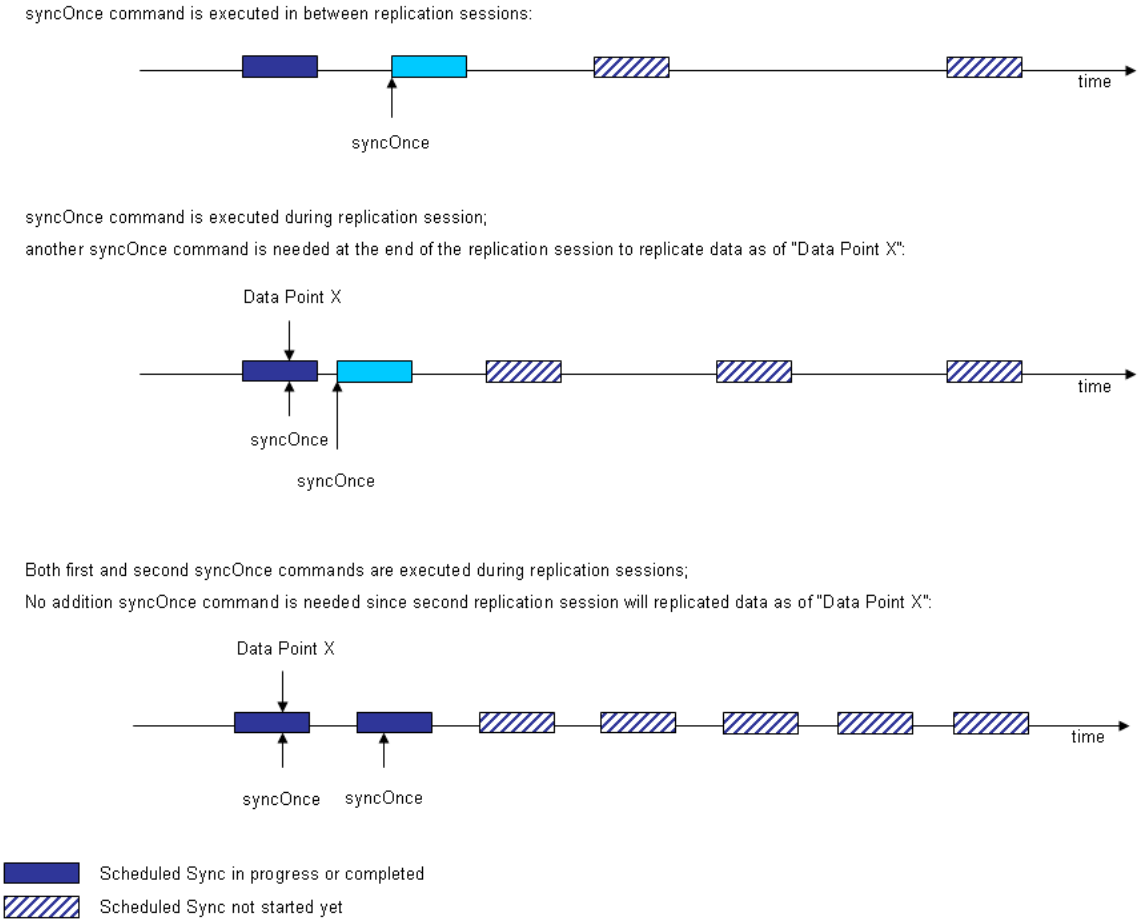
After executing syncOnce, SRM executes periodic querySyncStatus commands to check replication progress. Checking continues until replication completes or times out.

Array replication architecture dictates whether replication should be initiated at the source (a “push” model) or target (a “pull” model). SRM assumes a push model but an SRA can specify a different model in the queryCapabilities command results. SRM uses this information to determine the appropriate site to execute syncOnce and querySyncStatus commands.

Note: Both syncOnce and querySyncStatus must be executed at the same site (either source or target). SRM does not support execution of syncOnce and querySyncStatus at different sites.

Handling an “Another Replication in Progress” error condition

If replication is in progress when syncOnce executes, syncOnce must return a pre-defined error, along with information about the replication in progress so that SRM can monitor that replication with querySyncStatus and initiate another replication after it completes. If the subsequent syncOnce command fails with same “Another Replication in Progress” error, SRM waits for that replication to complete and then returns success from syncOnce.



Frequency of Sync Status Check

Replication time depends on a number of factors (storage array type, amount of data out of sync, and available bandwidth, to name a few). One global (per SRM server) configuration parameter specifying frequency of replication progress check may not be enough.

If SRA is capable of estimating the remaining replication time it should return it in syncOnce and querySyncStatus response. In that case, SRM will use this information to calculate the time of the next querySyncStatus.

Format of syncOnce request when initiated at replication source:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>syncOnce</Name>
```

```
    <Connections …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <SyncOnceParameters>
        <ArrayId>XXX</ArrayId>
        <PeerArrayId>YYY</PeerArrayId>
        <ConsistencyGroups>
            <ConsistencyGroup id="srm-1" />
        </ConsistencyGroups>
        <SourceDevices>
            <SourceDevice id="xxx-2" />
            <SourceDevice id="xxx-3" />
        </SourceDevices>
    </SyncOnceParameters>
</Command>
```

Format of syncOnce response when initiated at replication source:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
    <SyncOnceResults>
        <ConsistencyGroups>
            <ConsistencyGroup id="srm-1">
                <DeviceSync id="sync-1020" status="inProgress">
                    <Progress>0</Progress>
                    <RemainingTimeEstimate>150</RemainingTimeEstimate>
                </DeviceSync>
            </ConsistencyGroup>
        </ConsistencyGroups>
        <SourceDevices>
          <SourceDevice id="xxx-2">
             <DeviceSync id="sync-1022" status="inProgress">
                <Progress>0</Progress>
                <RemainingTimeEstimate>150</RemainingTimeEstimate>
             </DeviceSync>
          </SourceDevice>
          <SourceDevice id="xxx-3">
             <Warning code="500"/>
               <!--
                500 - predefined code for "another sync in progress"
                The following DeviceSync tag describes the replication in
                progress. SRM will monitor this replication and initiate
                another one once this replication completes.
               -->
             <DeviceSync id="sync-1022" status="inProgress">
                <Progress>15</Progress>
                <RemainingTimeEstimate>60</RemainingTimeEstimate>
             </DeviceSync>
          </SourceDevice>
        </SourceDevices>
    </SyncOnceResults>
</Response>
```

Format of syncOnce request when initiated at replication target:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>syncOnce</Name>
```

```
    <Connection …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <SyncOnceParameters>
        <ArrayId>YYY</ArrayId>
        <PeerArrayId>XXX</PeerArrayId>
        <TargetGroups>
            <TargetGroup key="srm-target-1" />
        </TargetGroups>
        <TargetDevices>
            <TargetDevice key="xxx-2-peer"/>
            <TargetDevice key="xxx-3-peer"/>
        </TargetDevices>
    </SyncOnceParameters>
</Command>
```

Format of syncOnce response when initiated at replication target:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
   <SyncOnceResults>
        <TargetGroups>
            <TargetGroup key="srm-target-1">
                <DeviceSync id="sync-1021" status="inProgress">
                    <Progress>0</Progress>
                    <RemainingTimeEstimate>120</RemainingTimeEstimate>
                </DeviceSync>
            </TargetGroup>
        </TargetGroups>
        <TargetDevices>
            <TargetDevice key="xxx-2-peer">
                <DeviceSync id="sync-1022" status="inProgress">
                    <Progress>0</Progress>
                    <RemainingTimeEstimate>150</RemainingTimeEstimate>
                </DeviceSync>
            </TargetDevice>
            <TargetDevice key="xxx-3-peer">
                <Warning code="500"/>
                 <!--
                    500 - predefined code for "another sync in progress"
                    -->
                <DeviceSync id="sync-1023" status="inProgress">
                    <Progress>15</Progress>
                    <RemainingTimeEstimate>60</RemainingTimeEstimate>
                </DeviceSync>
            </TargetDevice>
        </TargetDevices>
    </SyncOnceResults>
</Response>
```

Progress – replication progress in percent
RemainingTimeEstimate – estimate of remaining time in seconds (optional).

querySyncStatus

Returns status of replication sessions initiated with syncOnce command.

Format of querySyncStatus request initiated at replication source:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>querySyncStatus</Name>
    <Connections …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <QuerySyncStatusParameters>
        <ArrayId>XXX</ArrayId>
        <PeerArrayId>YYY</PeerArrayId>
        <ConsistencyGroups>
            <ConsistencyGroup id="srm-1" syncId="sync-1020"/>
        </ConsistencyGroups>
        <SourceDevices>
            <SourceDevice id="xxx-2" syncId="sync-1022"/>
            <SourceDevice id="xxx-3" syncId="sync-1023"/>
        </SourceDevices>
    </QuerySyncStatusParameters>
</Command>
```

Format of the querySyncStatus response initiated at replication source:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
   <QuerySyncStatusResults>
        <ConsistencyGroups>
            <ConsistencyGroup id="srm-1">
                <DeviceSync id="sync-1020" status="inProgress">
                    <Progress>10</Progress>
                    <RemainingTimeEstimate>100</RemainingTimeEstimate>
                </DeviceSync>
            </ConsistencyGroup>
        </ConsistencyGroups>
        <SourceDevices>
            <SourceDevice id="xxx-2">
                <DeviceSync id="sync-1022" status="complete"/>
            </SourceDevice>
            <SourceDevice id="xxx-3">
                <Error code="1209"/>
            </SourceDevice>
        </SourceDevices>
    </QuerySyncStatusResults>
</Response>
```

Format of the querySyncStatus request initiated at replication target:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>querySyncStatus</Name>
    <Connection …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <QuerySyncStatusParameters>
        <ArrayId>YYY</ArrayId>
        <PeerArrayId>XXX</PeerArrayId>
        <TargetGroups>
```

```
            <TargetGroup key="srm-target-1" syncId="sync-1020"/>
        </TargetGroups>
        <TargetDevices>
            <TargetDevice key="xxx-2-peer" syncId="sync-1022"/>
            <TargetDevice key="xxx-3-peer" syncId="sync-1023"/>
        </TargetDevices>
    </QuerySyncStatusParameters>
</Command>
```

Format of the querySyncStatus response initiated at replication target:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
   <QuerySyncStatusResults>
        <TargetGroups>
            <TargetGroup key="srm-target-1">
                <DeviceSync id="sync-1020" status="inProgress">
                    <Progress>10</Progress>
                    <RemainingTimeEstimate>100</RemainingTimeEstimate>
                </DeviceSync>
            </TargetGroup>
        </TargetGroups>
        <TargetDevices>
            <TargetDevice key="xxx-2-peer">
                <DeviceSync id="sync-1022" status="completed"/>
            </TargetDevice>
            <TargetDevice key="xxx-3-peer">
                <Error code="1401" />
            </TargetDevice>
        </TargetDevices>
    </QuerySyncStatusResults>
</Response>
```

Note: querySyncStatus response format is identical to syncOnce response format except for the name of immediate child tag of the root Response tag.

Table 14 syncOnce and querySyncStatus format

Element Name	Content	Description
Command	Name, OutputFile, StatusFile, LogLevel, LogDirectory, Connections, SyncOnceParameters | QuerySyncStatusParameters	
Name		Command name, “syncOnce” or “querySyncStatus”
SyncOnceParameters	ArrayId, PeerArrayId, (ConsistencyGroups?, SourceDevices?) | (TargetGroups?, TargetDevices?)	

QuerySyncStatusParameters	ArrayId, PeerArrayId, (ConsistencyGroups?, SourceDevices?) | (TargetGroups?, TargetDevices?)	
ArrayId		ID of the local storage array.
PeerArrayId		ID of the remote storage array.
ConsistencyGroups	ConsistencyGroup+	
ConsistencyGroup	Attributes: id, syncId (querySyncStatus command only)	
TargetGroups	TargetGroup	
TargetGroup	Attributes: key, syncId (querySyncStatus command only)	
SourceDevices	SourceDevice+	Replicated read-write devices to be synchronized if command is executed at the replication source.
SourceDevice	Attributes: id, syncId (querySyncStatus command only)	Value of id attribute is device ID returned by discoverDevices command. Value of syncId attribute id is ID of replication session initiated with syncOnce command. Applies to querySyncStatus command only.
TargetDevices	TargetDevice+	Replication target devices to be synchronized if command is executed at the replication target.
TargetDevice	Attributes: key, syncId (querySyncStatus command only)	Value of key attribute is target device key returned by discoverDevices command. Value of syncId attribute id is ID of replication session initiated with syncOnce command. Applies to querySyncStatus command only.
Response	(	

	SyncOnceResults | QuerySyncStatusResults, Warnings?) | Error	
SyncOnceResults	(ConsistencyGroups?, SourceDevices?) | (TargetGroups?, TargetDevices?)	
ConsistencyGroups	ConsistencyGroup+	
ConsistencyGroup	(DeviceSync, Warnings?) | Error Attributes: id	
DeviceSync	Progress?, RemainingTimeEstimate? Attributes: id, status	Value of id attribute must be unique. This value identifies a particular replication session initiated. SRM will use this value to check status of this replication session with querySyncStatus command. Valid values of status attribute are “inProgress” and “complete”.
Progress		Optional. Progress of the replication measured in percents completed.
RemainingTimeEstimate		Optional. Estimate of the time remaining before replication completes in seconds.
TargetGroups	TargetGroup+	
TargetGroup	(DeviceSync, Warnings?) | Error Attributes: key	
SourceDevices	SourceDevice+	
SourceDevice	(DeviceSync, Warnings?) | Error Attributes: id	Value of id attribute must correspond to a device ID specified in the request.
TargetDevices	TargetDevice+	

TargetDevice	(DeviceSync, Warnings?) | Error Attributes: key	Value of key attribute must correspond to a target device key specified in the request.
QuerySyncStatusResults	ConsistencyGroups?, (Devices | TargetDevices)?	

reverseReplication (and prepareReverseReplication)

Configures replication in reverse (see Reverse Replication after Failover).

SRM follows prepareReverseReplication and reverseReplication commands with syncOnce and querySyncStatus commands to initiate and monitor data transfer. SRM displays the status of this data transfer to the user. Once the data transfer completes SRM will indicate to the user that system is ready.

The SRA implementation of prepareReverseReplication and reverseReplication commands must be non-disruptive to production services running at the recovery site.

Commands prepareReverseReplication and reverseReplication must be idempotent.
Both commands are expected to succeed, even when the given replication settings are not valid. In this case, the SRA should complete the required actions, but return a warning concerning the need to correct the replication settings manually.

Format of the reverseReplication request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>reverseReplication</Name>
    <Connections …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <ReverseReplicationParameters>
        <ArrayId>YYY</ArrayId>
        <PeerArrayId>XXX</PeerArrayId>
        <ConsistencyGroups>
            <ConsistencyGroup id="srm-target-1">
                <ReplicationSettings …/>
            </ConsistencyGroup>
        </ConsistencyGroups>
        <Devices>
            <Device id="xxx-2-copy">
                <ReplicationSettings …/>
            </Device>
            <Device id="xxx-3-copy">
                <ReplicationSettings …/>
            </Device>
        </Devices>
    </ReverseReplicationParameters>
</Command>
```

ArrayId points to the local storage array.

PeerArrayId points to the remote storage array.

The value of the id attribute of the Device tag depends on where the command is executed (source or target):

- When executed at the desired replication source, it is the ID of the failed-over device (returned by failover command)
- When executed at the desired replication target, it is the ID of the original source device (returned by discoverDevices command)

Format of the prepareReverseReplication request is identical to the format of the reverseReplication request except for the command name.

```
<Command xmlns="http://www.vmware.com/srm/sra/v2">
   <Name>prepareReverseReplication</Name>
   …
```

Format of reverseReplication response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
   <ReverseReplicationResults>
       <ConsistencyGroups>
          <ConsistencyGroup id="srm-target-1">
             <Success />
          </ConsistencyGroup>
       </ConsistencyGroups>
       <Devices>
          <Device id="xxx-2-copy">
             <Error …/>
          </Device>
          <Device id="xxx-3-copy">
             <Success />
          </Device>
       </Devices>
   </ReverseReplicationResults>
</Response>
```

Note: The response must contain all groups and devices specified in the request.

The format of the prepareReverseReplication response is identical to the format of the reverseReplication response except for the name of the immediate child of the Response tag, which is PrepareReverseReplicationResults.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
   <PrepareReverseReplicationResults>
      …
   </PrepareReverseReplicationResults>
</Response>
```

restoreReplication (and prepareRestoreReplication)

Discards any changes made to failed over devices and restores replication from original protected site to the recovery (see Restore Replication after Failover).

SRM stops production services running at the recovery site (started during failover) before executing prepareRestoreReplication and restoreReplication commands.

SRM re-starts production services (stopped during failover) at the original protected site once restoreReplication completes.

SRM follows prepareRestoreReplication and restoreReplication with syncOnce and querySyncStatus commands to initiate and monitor data transfer. SRM displays the status of this data transfer to the user. Once the data transfer completes SRM indicate to the user that system is ready.

Commands prepareRestoreReplication and restoreReplication must be idempotent.
Both commands are expected to succeed, even when the given replication settings are not valid. In this case, the SRA should complete the required actions, but return a warning concerning the need to correct the replication settings manually.

Format of the restoreReplication request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>restoreReplication</Name>
    <Connections …/>
    <OutputFile>path/to/response.xml</OutputFile>
    <StatusFile>path/to/status.xml</StatusFile>
    <LogLevel>verbose</LogLevel>
    <LogDirectory>path/to/log/directory</LogDirectory>
    <RestoreReplicationParameters>
        <ArrayId>XXX</ArrayId>
        <PeerArrayId>YYY</PeerArrayId>
        <ConsistencyGroups>
            <ConsistencyGroup id="srm-1">
                <ReplicationSettings …/>
            </ConsistencyGroup>
        </ConsistencyGroups>
        <Devices>
            <Device id="xxx-2">
                <ReplicationSettings …/>
            </Device>
            <Device id="xxx-3">
                <ReplicationSettings …/>
            </Device>
        </Devices>
    </RestoreReplicationParameters>
</Command>
```

ArrayId points to the local storage array.

PeerArrayId points to the remote storage array.

The value of the id attribute of the Device tag depends on where the command is executed (source or target):

- When executed at the desired replication source, it is the ID of the original source device (returned by discoverDevices command)
- When executed at the desired replication target, it is the ID of the failed-over device (returned by failover command)

The format of the prepareRestoreReplication request is identical to the format of the restoreReplication request except for the value of the command name.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Command xmlns="http://www.vmware.com/srm/sra/v2">
    <Name>prepareRestoreReplication</Name>
    …
```

Format of the restoreReplication response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
   <RestoreReplicationResults>
       <ConsistencyGroups>
           <ConsistencyGroup id="srm-1">
               <Success />
           </ConsistencyGroup>
       </ConsistencyGroups>
       <Devices>
           <Device id="xxx-2">
               <Error …/>
           </Device>
           <Device id="xxx-3">
               <Success />
           </Device>
       </Devices>
   </RestoreReplicationResults>
</Response>
```

Note: The response must contain all groups and devices specified in the request.

The format of the prepareRestoreReplication response is identical to the format of the restoreReplication response except for the RestoreReplicationResults tag name which becomes PrepareRestoreReplicationResults.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Response xmlns="http://www.vmware.com/srm/sra/v2">
   <PrepareRestoreReplicationResults>
      …
   </PrepareRestoreReplicationResults>
</Response>
```

Table 15 reverseReplication and restoreReplication format

Element Name	Content	Description
Command	Name, OutputFile, StatusFile, LogLevel, LogDirectory, Connections, PrepareReverseReplicationParameters | ReverseReplicationParameters | PrepareRestoreReplicationParameters | RestoreReplicationParameters	
Name		Command name, “prepareReverseReplication”, “reverseReplication”, “prepareRestoreReplication”, or “restoreReplication”.
ReverseReplicationParameters	ArrayId, PeerArrayId, ConsistencyGroups?, Devices?	
ArrayId		ID of the local storage array.
PeerArrayId		ID of the remote storage array.
ConsistencyGroups	ConsistencyGroup+	
ConsistencyGroup	ReplicationSettings?	

	Attributes: id	
Devices	Device+	Devices on local storage array.
Device	ReplicationSettings? Attributes: id	Value of id attribute is device ID returned by discoverDevices command.
ReplicationSettings		Optional. The format of this tag is defined by the SRA. SRM passes this tag unmodified as returned by queryReplicationSettings command.
PrepareReverseReplicationParameters	ArrayId, PeerArrayId, ConsistencyGroups?, Devices?	
PrepareRestoreReplicationParameters	ArrayId, PeerArrayId, ConsistencyGroups?, Devices?	
RestoreReplicationParameters	ArrayId, PeerArrayId, ConsistencyGroups?, Devices?	
Response	((ReverseReplicationResults | PrepareReverseReplicationResults | RestoreReplicationResults | PrepareRestoreReplicationResults), Warnings?) | Error	
ReverseReplicationResults	ConsistencyGroups?, Devices?	
ConsistencyGroups	ConsistencyGroup+	
ConsistencyGroup	(Success, Warnings?) | Error Attributes: id	
Devices	Device+	
Device	(Success, Warnings?) | Error Attributes: id	Value of id attribute must correspond to a device ID specified in the request.
Success		Indicates that device is now in the requested state. The device might have been in this state or SRA performed

		necessary operations to get the device into desired state.
PrepareReverseReplicationResults	ConsistencyGroups?, Devices?	
RestoreReplicationResults	ConsistencyGroups?, Devices?	
PrepareRestoreReplicationResults	ConsistencyGroups?, Devices?	

Known Limitations of the Glenlivet Release of Site Recovery Manager

The first release of VMware vCenter Site Recovery Manager which supports the interface described in this specification will have several limitations which will be removed in future releases.

1. Some *optional* commands are actually *required* for this release (i.e., syncOnce and querySyncStatus)
2. Some optional commands are not supported (i.e., queryRecoveryPoints and queryDeviceDetails)

Appendix – A: Mandatory Elements of SRA Logging

Introduction

The SRA Specification 2.0 requires that SRAs be capable of logging at five levels: trivia, verbose, info, warning and error. Although certification only requires logging at the verbose level, trivia level logging should also be implemented for support reasons, and info level logging should be included for log brevity, for use when desired by the end user in well tested, stable SRM environments.

SRAs certified by VMware for use with all versions of Site Recovery Manager are required to log all actions taken in service of any request issued by SRM. For SRA certification, SRA logging is mandated to be "verbose", meaning a log which provides extremely detailed, granular information pertaining to all objects being acted on in any way. All actions taken by the SRA, *or on behalf of the SRA by any remote service*, must be logged in a meaningful way that should be fathomable by the SRM and storage administrators.

The mandatory elements of SRA logging, detailed herein, will be required for SRM 5.1 and above due to increasing support issues and customer concerns. With the increasing capabilities of SRM 5.x, it has become obvious to VMware support and many SRM subscribers that the lack of more stringent, well-defined (if not standardized) logging by Storage Replication Adapters is untenable.

VMware Global Support Services requires verbose, intelligible logs to help perform root cause analysis of storage related SRM issues.

VMware Ecosystem Engineering also requires verbose SRA logging in order to verify SRA behavior as conforming to the SRA Specification. This is not 100% achievable through our SRA Certification Suite, and so SRA logging has become a standard part of SRA Certification reviews. SRAs with non-compliant logging will no longer be certified by VMware.

Object Names

Object names must be used in all log messages pertaining to any object named in the incoming request. The object name must match the name used in the SRM or Certsuite request. Other object names may also be used, but they must be associated with the common name as it appears in the XML document containing the request being processed.
Some common object names, taken from a sample request, might be:

- An array ID like: **APM000999876123767-server_2**
- An initiator ID: **iqn.1998-01.com.vmware:10.20.177.12**
- A target device key:
  fs972_T6_LUN20_APM00999876123767_fs2056_T9_LUN20_APM00999876121011

Whenever the object name includes embedded blanks, they should be double-quoted in the log message.
Excerpt of sample follows --

```
<ArrayId>APM000999876123767-server_2</ArrayId>
    <AccessGroups>
      <AccessGroup id="domain-s277">
        <Initiator type="iSCSI" id="iqn.1998-01.com.vmware:10.20.177.12"/>
        <Initiator type="NFS" id="10.20.177.13"/>
      </AccessGroup>
    </AccessGroups>
```

```
      <TargetDevices>
         <TargetDevice
key="fs972_T6_LUN20_APM00999876123767_fs2056_T9_LUN20_APM00999876121011">
            <AccessGroups>
               <AccessGroup id="domain-s277"/>
            </AccessGroups>
         </TargetDevice>
      </TargetDevices>
```

Step Messages

Every SRM workflow involves several discrete steps to accomplish the desired end result. The SRA Specification 2.0 states what must occur in each workflow, what may occur if certain options are implemented, and, in some cases, what may not occur (i.e. replication may NOT be interrupted during test recovery). At minimum, each SRA must output intelligible (non-cryptic) messages, with object names, for each step of each workflow. All SRA log messages must be written with the current date and time, including all step messages. In order to clearly delineate each step within the context of a truly verbose log, the SRA must issue `commence/complete` message pairs for each step, with other step specific messages in between the `commence` message and the `complete` message. [This is not to suggest that VMware mandates any type of serialization; because of concurrency, it is possible that many LUNs are operated on at the same time, and the commence/complete sections will overlap. A unique identifier in each message header, following the date/time stamp, would likely be required to distinguish every constituent message of each commence/complete block of messages.]

An example for two steps of the recovery workflow might appear as:

```
--> Wed Nov 16 16:01:32 2011:[verbose]Commence: Suspending array replication prior to failover
--> Wed Nov 16 16:01:32 2011:[verbose]Executing method [quiesceReplicationSession] with argument
[SRM_PROD_LUN_12] on array APM0095792232-server_2
--> Wed Nov 16 16:01:33 2011:[verbose]WARNING: Replication session for volume[SRM_PROD_LUN_12]
on array APM0095792232-server_2 has been terminated.
--> Wed Nov 16 16:01:33 2011:[verbose]Complete: Suspension of array replication is now COMPLETE
--> Wed Nov 16 16:01:33 2011:[verbose]Commence: Promotion of LUN replica SRM_PROD_LUN_12 as
writable device.
--> Wed Nov 16 16:01:33 2011:[verbose]Executing method [modifyVolume] with argument
[SRM_PROD_LUN_12] on array APM0095792232-server_2
--> Wed Nov 16 16:01:33 2011:[verbose]  Property [accessType] value [1]
--> Wed Nov 16 16:01:33 2011:[verbose]  Property [adminStatus] value [2]
--> Wed Nov 16 16:01:33 2011:[verbose]  Property [replicaService] value [6]
--> Wed Nov 16 16:01:33 2011:[verbose]  Property [promotionPolicy] value [1]
--> Wed Nov 16 16:01:33 2011:[verbose]  Property [failBackMode] value [2]
--> Wed Nov 16 16:01:33 2011:[verbose]  Property [multiAccess] value [0]

--> Wed Nov 16 16:01:34 2011:[verbose]Complete: Promotion of LUN replica SRM_PROD_LUN_12 as
writable device is now COMPLETE.
--> Wed Nov 16 16:01:34 2011:[verbose]Commence: Resuming array replication post-failover
--> Wed Nov 16 16:01:35 2011:[verbose]NOTICE: Replication session for volume[SRM_PROD_LUN_12] on
array APM0095792232-server_2 has been successfully re-established.
```

--> Wed Nov 16 16:01:35 2011:[verbose]Complete: Resuming array replication post-failover is now COMPLETE

Required Messaging by Workflow or Command

(*) indicates “required if implemented” for optional features or parameters

All commands

- SRA release signature – showing official SRA name, version, and build date.
- Incoming request – the actual request received from SRM, with appropriate indentation

discoverDevices

- Generic discovery message – detailing on which array device discovery is occurring, and any device filtering that may be set via opaque parameters.
- Device exclusion messages – detailing the reason any replicated device will not be reported in the discoverDevices response.

testFailoverStart

- Snapshot creation messages – snapshot commence/complete with object name to be given in XML response; must detail replica selection for snapshot base; all parameters used for actual snapshot (temporary copy) creation
- Snapshot presentation messages – must detail all parameters for exposure of the snapshot copy to the recovery site host(s) including: host/initiator name(s), access type, multi-host access, etc.

testFailoverStop

- Snapshot masking messages – must detail all actions taken to remove host access to the temporary copy (snapshot)
- Snapshot removal messages – commence/complete message pair showing actual destruction of the temporary copy with object name to be given in XML response

prepareFailover*

- Source device demotion messages - commence/complete message pair showing actual demotion of the source devices and/or groups with object name(s) to be given in XML response
- Source device snapshot messages* - if snapshots are taken of the source devices or groups, a commence/complete pair must detail each snapshot with all parameters used for its creation, including the object name to be used in the XML response

failover

- Replication suspend/resume messages – commence/complete message pairs for both suspension and resumption of replication sessions for affected target devices. If replication suspension is array-wide (all replication ceases) this should be noted, otherwise object names of suspended sessions must be identifiable with target devices/groups to be promoted. If replication is suspended automatically at failover, SRA should log an appropriate message that this has occurred, and at what level.

- Revert snapshot messages* – SRAs may take a snapshot of a target device or group prior to promoting it, to be retained for target reversion when restoreReplication is used. Creation of these "revert" snapshots must be detailed by messages showing all snapshot details.
- Target device promotion messages - commence/complete message pair showing actual promotion of the target devices and/or groups with object name(s) to be given in XML response.
- Target presentation messages – must detail all parameters for exposure of the snapshot copy to the recovery site host(s) including: host/initiator name(s), access type, multi-host access, etc. Any presentation errors must also be detailed with initiator IDs and explicit failure information.

reverseReplication

- Replication reversal messages – commence/complete series with detailed information for each device whose role is being reversed, including any replication settings that are being applied after the replication reversal is accomplished.

restoreReplication

- Source device messages – commence/complete series with detailed information for each source device that is being promoted and presented to protected site ESX hosts.
- Target device messages – commence/complete series detailing how target devices are reset to pre-failover state with all post-failover changes discarded. This series must also include device masking messages showing all access to the targets being removed from all recovery site hosts.

syncOnce

- Replica creation messages – explicit messages detailing each new replica creation and the replication method (sync, async, continuous) being used. If SRA supports multiple recovery points, recoveryPoint information must be included. If replica is already current, or immediately becomes current, a completion message must also be logged.
- reverseReplication failsafe message – when syncOnce is issued as part of Re-Protect workflow (after successful reverseReplication), a message indicating the name of the failsafe snapshot at the intended destination should be logged. This may be the snapshot taken by prepareFailover which could be used to restore production services at the destination (the intended Recovery Site for failback) if the current Protection Site were to fail before the syncOnce was completed.
- restoreReplication failsafe message* – when syncOnce is issued as part of Restore workflow (after successful restoreReplication), a message indicating the name of the failsafe snapshot at the intended destination should be logged. This may be the snapshot taken by failover which could be used to restore production services at the destination (the Recovery Site) if the current Protection Site were to fail before the syncOnce was completed.

prepareReverseReplication*

- Target messages - when reverseReplication uses default execution location of source, prepare is run at the intended target site. Target messages will clearly show all actions taken against target site storage objects, including LUNs, removal of host access, and replication settings.
- Source messages - when reverseReplication uses execution location of target, prepare is run at the intended source site. Source messages will show all actions taken against all source site storage objects.

prepareRestoreReplication*

- Target messages - when restoreReplication uses default execution location of source, prepare is run at the intended target site. Target messages will clearly show all actions taken against target site storage objects, including LUNs, host access, and replication sessions.
- Source messages - when restoreReplication uses execution location of target, prepare is run at the intended source site. Source messages will show all actions taken against all source site storage objects, including: access type change to r/w; device promotion; host access restoration.

Example SRA Logs

The following are general examples, shown without the required time stamps. Only major log elements are shown in the examples. Other verbose messages are array or replication software specific and must be logged in context to the mandatory elements.

testFailoverStart

```
Commence: Temporary Snapshot Copy of [TEST_LUN_01.001] is being created
for testFailoverStart request.
→Replica of [TEST_LUN_01] with key [TEST_LUN_01.001] is being cloned from
Recovery Point [20111211-080852.422]
→Snapshot copy [TEST_LUN_01-TestFailoverSnap] has attributes: [r/w]
[multihost] [tp] [chap]
→Snapshot copy [TEST_LUN_01-TestFailoverSnap] is being exposed to host
[iqn.1998-01.com.vmware:esx-host01] on target [iqn.2005-07.com.webm:reco-
array-port0a] using chapName [shakeitplz]
→Snapshot copy [TEST_LUN_01-TestFailoverSnap] is being exposed to host
[iqn.1998-01.com.vmware:esx-host02] on target [iqn.2005-07.com.webm:reco-
array-port0a] using chapName [shakeitplz]
Complete: Temporary Snapshot Copy of [TEST_LUN_01.001] has been created
and presented to (2) hosts for testFailoverStart request.
```

testFailoverStop

```
Commence: Temporary Snapshot Copy of [TEST_LUN_01.001] is being destroyed
for testFailoverStop request.
→Snapshot copy [TEST_LUN_01-TestFailoverSnap] is being masked from host
[iqn.1998-01.com.vmware:esx-host01]
→Snapshot copy [TEST_LUN_01-TestFailoverSnap] is being masked from host
[iqn.1998-01.com.vmware:esx-host02]
→Snapshot copy [TEST_LUN_01-TestFailoverSnap] is being offlined
→Snapshot copy [TEST_LUN_01-TestFailoverSnap] is being deleted
Complete: Temporary Snapshot Copy of [TEST_LUN_01.001] has been removed
from service and destroyed for testFailoverStop request.
```

Failover

```
Commence: Suspending replication of [TEST_LUN_01, TEST_LUN_02] on array
[SiteB-Array] for failover request
→Replication session [TEST_LUN_01] to target [TEST_LUN_01.001] on array
[SiteB-Array] is terminated. Target is now promotable.
→Replication session [TEST_LUN_02] to target [TEST_LUN_02.001] on array
[SiteB-Array] is terminated. Target is now promotable.
```

Complete: Suspension of replication of [TEST_LUN_01, TEST_LUN_02] on array [SiteB-Array] for failover request is now COMPLETE
Commence: Revert Snapshot Copy of [TEST_LUN_01.001] is being created for potential restoreReplication request.
→Replica of [TEST_LUN_01] with key [TEST_LUN_01.001] is being cloned from Recovery Point [20111212-082242.589]
→Revert Snapshot copy [TEST_LUN_01.001-Revert] has attributes: [r/o] [offline] [tp]
Complete: Revert Snapshot Copy of [TEST_LUN_01.001] has been created for potential use by a restoreReplication request.
Commence: Revert Snapshot Copy of [TEST_LUN_02.001] is being created for potential restoreReplication request.
→Replica of [TEST_LUN_02] with key [TEST_LUN_02.001] is being cloned from Recovery Point [20111212-082242.747]
→Revert Snapshot copy [TEST_LUN_02.001-Revert] has attributes: [r/o] [offline] [tp]
Complete: Revert Snapshot Copy of [TEST_LUN_02.001] has been created for potential use by a restoreReplication request.

Commence: Promotion of devices [TEST_LUN_01.001, TEST_LUN_02.001] on array [SiteB-Array] for failover request
→Replica of [TEST_LUN_01] with key [TEST_LUN_01.001] is being promoted from Recovery Point [20111212-082242.589]
→Promoted target [TEST_LUN_01.001] has attributes: [r/w] [tp] [chap]
→Promoted target [TEST_LUN_01.001] is being exposed to host [iqn.1998-01.com.vmware:esx-host123] using chapName [shakeitplz]
→Replica of [TEST_LUN_02] with key [TEST_LUN_02.001] is being promoted from Recovery Point [20111212-082242.747]
→Promoted target [TEST_LUN_02.001] has attributes: [r/w] [multihost] [tp] [chap] [vmimageconsistent]
→Promoted target [TEST_LUN_02.001] is being exposed to host [iqn.1998-01.com.vmware:esx-host123] using chapName [shakeitplz]
Complete: Promotion is now complete for devices [TEST_LUN_01.001, TEST_LUN_02.001] for failover request.

prepareFailover

Commence: Demotion of devices [TEST_LUN_03] on array [SiteA-Array] for prepareFailover request
→Source device [TEST_LUN_03] is being masked from host [iqn.1998-01.com.vmware:esx-host44]
→Source device [TEST_LUN_03] is being offlined
→Source device [TEST_LUN_03] host access being changed to [r/o]
→Source device [TEST_LUN_03] is being onlined
Complete: Demotion of source device [TEST_LUN_03] is now COMPLETE for prepareFailover request.
Commence: Pre-Failover Snapshot Copy of [TEST_LUN_03] is being created for prepareFailover request.
→Snapshot [TEST_LUN_03-preFailover] of Source device [TEST_LUN_03] has been created with Recovery Point [20111212-185500.003]
→Snapshot [TEST_LUN_03-preFailover] has attributes: [r/o] [offline]
Complete: Pre-Failover Snapshot Copy of [TEST_LUN_03] has been created for prepareFailover request.