EMC VPLEX: LEVERAGING ARRAY BASED AND NATIVE COPY TECHNOLOGIES
Third Edition

Abstract
This white paper provides best practices planning and use cases for using array-based and native replication solutions with EMC® VPLEX™ Local and EMC VPLEX Metro.
Table of Contents

**Executive summary** ............................................................................................................. 4

**Document scope and limitations** ....................................................................................... 4
  Audience ................................................................................................................................. 4

**Introduction** .......................................................................................................................... 5
  Terminology ............................................................................................................................. 5

**VPLEX Technology** ............................................................................................................. 6
  EMC VPLEX Virtual Storage ................................................................................................. 6
  EMC VPLEX Architecture ..................................................................................................... 7
  EMC VPLEX Family ............................................................................................................... 8

**Section 1: Array-based Replication with VPLEX** ................................................................. 9
  Introduction ............................................................................................................................ 9
  Array-based copy for VPLEX RAID-0 volumes .................................................................. 10
    Creating VPLEX RAID-0 Copies ....................................................................................... 12
    Resynchronizing (updating) local array-based copies presented through VPLEX .......... 15
  Advanced array-based copy use cases ................................................................................. 19
    Creating VPLEX RAID-1 or Distributed RAID-1 array-based copies ......................... 20
    Procedure: Resynchronizing (refreshing) array-based copies of RAID-1 VPLEX virtual volumes ......................................................................................................................................... 23
  Array-based Restore for VPLEX RAID-0 Production Volumes ......................................... 28
  Array-based Restore of VPLEX RAID-1 Production Volumes .......................................... 32

**Section 2: VPLEX Native Copy Capabilities** ................................................................ 39
  Scope and limitations ............................................................................................................. 39
  Introduction ............................................................................................................................ 39
  EMC VPLEX Native Copy Overview .................................................................................. 40
  EMC VPLEX native copying with Unisphere for VPLEX ................................................... 42
    Creating native copies of VPLEX virtual volumes ......................................................... 42
    Restoring from VPLEX native copies .............................................................................. 51
  EMC VPLEX Native copies with the VPLEX CLI ............................................................... 52
    Creating VPLEX Native copies ....................................................................................... 52
    Restoring from VPLEX native copies .............................................................................. 55

**Conclusion** ......................................................................................................................... 56

**References** ......................................................................................................................... 56
Executive summary
The EMC® VPLEX™ family removes physical barriers within, across, and between data centers. VPLEX Local provides simplified management and non-disruptive data mobility across heterogeneous arrays. VPLEX Metro provides high availability and non-disruptive mobility between two data centers within synchronous distances. With unique scale-up architecture, VPLEX's advanced data caching and distributed cache coherency provides workload resiliency, automatic sharing, and balancing and failover of storage domains, and enables both local and remote data access with predictable service levels.

Preserving investments in array-based storage replication technologies like EMC MirrorView™, TimeFinder™, SnapView™, and SRDF™ is crucial in today's IT environments. This paper outlines the considerations and methodologies for array-based replication technologies within VPLEX Local and Metro environments. Procedural examples for various replication scenarios, examples of host based scripting, and VPLEX native copy capabilities are examined. The key conclusion of the paper is array based replication technologies continue to deliver their original business value with VPLEX environments.

Document scope and limitations
VPLEX is constantly evolving as a platform. The procedures and technology discussed in this white paper are only applicable to the VPLEX Local and VPLEX Metro products. In particular, the procedures described only apply to the write-through caching versions of VPLEX running GeoSynchrony version 5.2 or higher. Please consult with your local EMC support representatives if you are uncertain as to the applicability of these procedures to your VPLEX environment.

Audience
This white paper is intended for technology architects, storage administrators, and system administrators who are responsible for architecting, creating, managing, and using the IT environments that utilize EMC VPLEX technologies. It is assumed that the reader is familiar with EMC VPLEX and storage array-based replication technologies.
**Introduction**

This white paper reviews the following topics:

- Creation of
Section 1: Array-based Replication with VPLEX

- volumes
- Re-synchronization of array-based copies of VPLEX volumes
- Restoration of VPLEX volumes using array-based copies
- VPLEX native copy capabilities

Section 1 of this paper examines the technical impact of VPLEX on array-based replication productions like EMC MirrorView, TimeFinder, SnapView, SRDF, and XtremIO snapshots. The necessary pre-copy considerations, logical checks, process changes and use case examples are provided. The examples illustrate the correct way to account for VPLEX and potential storage conditions that could impact the consistency of the copy or snapshot. Section 2 explores VPLEX’s native copy capabilities for both the Unisphere for VPLEX GUI and for the VPLEX CLI. VPLEX CLI commands can be accessed via the vapi script (available from your EMC account team), direct ssh login or via the VPLEX RESTful API.

**Note:** Using the built-in copy capability of VPLEX opens the door to creating copies of virtual volumes from heterogeneous storage arrays within and across data centers.

### Terminology

**Table 1: Operational Definitions**

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage volume</td>
<td>LUN or unit of storage presented by the back-end arrays.</td>
</tr>
<tr>
<td>Metadata volume</td>
<td>System volume that contains metadata about the devices, virtual volumes,</td>
</tr>
<tr>
<td></td>
<td>and cluster configuration.</td>
</tr>
<tr>
<td>Extent</td>
<td>All or part of a storage volume.</td>
</tr>
<tr>
<td>Device</td>
<td>Raid protection scheme applied to an extent or group of extents.</td>
</tr>
<tr>
<td>Virtual volume</td>
<td>Unit of storage presented by the VPLEX front-end ports to hosts.</td>
</tr>
<tr>
<td>Front-end port</td>
<td>Director port connected to host initiators (acts as a target).</td>
</tr>
<tr>
<td>Back-end port</td>
<td>Director port connected to storage arrays (acts as an initiator).</td>
</tr>
<tr>
<td>Director</td>
<td>The central processing and intelligence of the VPLEX solution. There are</td>
</tr>
<tr>
<td></td>
<td>redundant (A and B) directors in each VPLEX Engine.</td>
</tr>
<tr>
<td>Acronym/Abbreviation</td>
<td>Definition</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Copy</td>
<td>An independent full disk copy of another device.</td>
</tr>
<tr>
<td>Source device</td>
<td>Standard or primary array device. Typically used to run production applications.</td>
</tr>
<tr>
<td>Target Device</td>
<td>A secondary array device designated to receive new data (overwritten) from the source device.</td>
</tr>
<tr>
<td>FE port</td>
<td>Front-end (target ports visible to hosts).</td>
</tr>
<tr>
<td>BE port</td>
<td>Back-end (initiator ports visible to storage arrays).</td>
</tr>
</tbody>
</table>

### VPLEX Technology

#### EMC VPLEX Virtual Storage

EMC VPLEX virtualizes traditional physical storage array devices by applying three layers of logical abstraction. The logical relationships between each abstraction layer are shown below in Figure 1.

Extents are the mechanism used by VPLEX to divide storage volumes. Extents may be all or part of the underlying storage volume. EMC VPLEX aggregates extents and applies various RAID geometries (i.e. RAID-0, RAID-1, or RAID-C) to them within the device layer. Devices are constructed using one or more extents, and can be combined into more complex RAID schemes and device structures as desired. At the top layer of the VPLEX storage structures are virtual volumes. Virtual volumes are created from devices and inherit the size of underlying device. Virtual volumes are the elements VPLEX exposes to hosts via its FE ports. Access to virtual volumes is controlled using storage views. Storage views are analogous to Auto-provisioning Groups on EMC Symmetrix® and VMAX® or to storage groups on EMC.
CLARiiON® and VNX®. They act as logical containers that determine host initiator access to VPLEX FE ports and virtual volumes.

**Figure 1: EMC VPLEX logical storage structures**

**EMC VPLEX Architecture**

As shown in Figure 2, VPLEX is a virtual storage solution for heterogeneous host operating system environments and for both EMC and non-EMC block storage. VPLEX resides between the servers and heterogeneous storage assets and introduces a new architecture with unique characteristics:

- Scale-out clustering hardware, which lets customers to start small and grow big with predictable service levels
- Advanced data caching utilizing large-scale SDRAM cache to improve performance and reduce I/O latency and array contention
- Distributed cache coherence for automatic sharing, balancing, and failover of I/O across the cluster
- Consistent view of one or more LUNs across VPLEX clusters separated either by a few feet within a data center or across synchronous distances (up to 10ms), enabling new models of high availability and workload relocation
EMC VPLEX Family

The following two EMC VPLEX offerings are relevant to the discussion in this paper:

**VPLEX Local**: This solution is appropriate for customers who desire non-disruptive and fluid mobility, high availability, and a single point of management for heterogeneous block storage systems within a single data center.

**VPLEX Metro**: This solution is for customers who desire non-disruptive and fluid mobility, high availability, and a single point of management for heterogeneous block storage systems within and across data centers at synchronous distance. The VPLEX Metro offering also includes the unique capability to remotely export virtual storage across datacenters without the need for physical storage at the remote site.
Section 1: Array-based Replication with VPLEX

Introduction

Array based copy technologies are feature rich, mature, robust, scalable, and purpose-built to provide enterprise-class replication services. Deploying VPLEX in front of storage frames with these services does not diminish their function or value. Underlying storage devices are left untouched by VPLEX and the array replication technologies can continue to provide backup, business continuity, operational recovery, or QA/Test/Dev functionality.

VPLEX enables array-based replication products like EMC MirrorView, TimeFinder, SRDF, SnapView, and XtremIO snapshots to continue to deliver full functionality and business value. Unlike other virtual storage solutions that require (or strongly recommend) slicing physical storage into small pieces and a complete reconstruction of those components into virtual storage, VPLEX has no such requirement. This is because VPLEX uses one to one physical to virtual storage mappings as the default methodology. When storage arrays are virtualized using VPLEX they do not lose their original replication capabilities (local or remote). Similar to traditional array based restoration or refresh, prior to restore host applications must be stopped or quiesced. Once the restore process is completed, the virtual volume can once again be accessed by the host application.

As mentioned previously, VPLEX Local and Metro use a write-through caching architecture. One of the implications of this architecture is that the VPLEX cache (not the underlying storage array) is the source for many for host read requests. Cache invalidation (for both hosts and VPLEX), therefore, becomes extremely important for array-based restoration operations because the array is modifying the underlying physical storage outside of the host and VPLEX I/O path. When this happens, the read caches maintained by the host and by VPLEX will not match the data within the array. This condition has the potential to cause data loss or corruption. For this reason, the cache invalidation has to happen immediately following the array restore/refresh (prior to the resumption of host I/O). VPLEX virtual volume read cache invalidation can be accomplished in a number different ways outlined in the following sections. The version of VPLEX code on the system will dictate which methods are available. Starting with VPLEX GeoSynchrony 5.2 patch 1, an enhancement was added to facilitate direct invalidation of VPLEX cache for single virtual volumes or for entire consistency groups.
Array-based copy for VPLEX RAID-0 volumes

Let’s start our discussion with some examples of using array-based replication (i.e. TimeFinder, SnapView, or XtremIO Snapshots) to create VPLEX RAID-0 copies. Generally speaking, array-based copy technologies provide independent full disk copies (Clones or BCVs) and/or space efficient copies (Snapshots). From a VPLEX point of view, both types of copies represent normal back-end array storage volumes.

**Note:** Snapshots or any thinly provisioned storage volumes should be identified as thin devices during the VPLEX claiming process.

These copies can either be presented back through VPLEX or directly from the underlying storage array to a backup host, to a test environment, or even back to the original host. For our discussion, it is assumed the copies will be presented back through VPLEX to a host. For this basic use case, each array-based copy has a one-to-one VPLEX virtual volume configuration (VPLEX device capacity = extent capacity = array storage volume capacity) and has a single extent VPLEX RAID-0 or two extent RAID-1 device geometry. The process for managing more complex geometry array-based copies (for example, a copy that is RAID-1 or distributed RAID-1) is covered in the advanced use case discussion at the end of this section.

Potential host, VPLEX, and storage array connectivity combinations for the basic use case are illustrated in Figures Figure 3 and Figure 4. Figure 3 shows a single array with a single extent (RAID-0 geometry) source and single extent (RAID-0) array based copy. Both the source and the copy storage volumes pass through VPLEX and then back to hosts. The array-based copy is presented to a separate second host.

![Figure 3: Array-based RAID-0 Copy of a RAID-0 Production Virtual Volume](image-url)
In Figure 4, the source volume has single extent RAID-1 geometry. Each mirror leg is a single extent that is equal in size to the underlying storage volume from each array. The difference in this topology is that there is an option to use the array-based copy services provided by array 1 and/or by array 2. The choice of which array to copy from can be based on available capacity, array workload or, perhaps, feature licensing. The objective remains the same as with Figure 3 -- to create a copy that has single extent (RAID-0) geometry.
The following procedures address the examples shown below:

1. Creation of RAID-0 array-based copies of VPLEX RAID-0 or RAID-1 source / production virtual volumes
2. Resynchronizing or updating RAID-0 array-based copies

**Example 1: Create / Refresh VPLEX RAID-0 Copy of RAID-0 Source**

**Example 2: Create / Refresh VPLEX RAID-0 Copy of RAID-1 Source**

Creating VPLEX RAID-0 Copies

This process augments the traditional best practices for local array-based copies using products like EMC TimeFinder, EMC SnapView, and EMC XtremIO snapshots. It accounts for VPLEX and for host access through VPLEX to array-based copies.

**Procedure: Creating VPLEX RAID-0 array based copies:**

1. For each storage array, identify source storage volumes to copy. The VPLEX storage-view will contain all VPLEX volumes that are visible to a given host or group of hosts. To determine the VPLEX to array device mapping use one of following methods:
   - Using the VPLEX CLI invoke the ‘show-use-hierarchy’ or ‘device drill-down’ command for the virtual volume(s) and underlying device(s) you wish to make a copy of. These outputs provide the virtual to physical storage volume mapping for each VPLEX volume.
- Use the Storage Viewer feature of Virtual Storage Integrator (VSI) for VMware vCenter Server (for ESX environments) to see the virtual device to physical array device mapping.

- Correlate the VPLEX virtual volumes to physical storage using the VPDID from the virtual volume to map to the back end storage array device. This information is available to the host via SCSI inquiry and from VPLEX via the /clusters/<cluster name>/exports/storage-view context. To get the back-end array device, you can map virtual volume VPDID back through the VPLEX and get the array LUN. From a VPLEX perspective there is a VPDID tied to the virtual volume the host sees and a different VPDID tied to the back end storage array.

2. Using the VPLEX CLI or REST API, check for valid copy conditions to ensure data consistency:

   **Note:** Each of the following checks are typically scripted or built into code that orchestrates the overall copy process on the array.

   a. Confirm a VPLEX ndu is not in progress. Using the VPLEX CLI issue the ‘ndu status’ command and confirm that the response is ‘*No firmware, BIOS/POST, or SSD upgrade is in progress.*’

   b. Confirm the device is healthy. Issue the ‘ll’ command from the /clusters/<cluster name>/virtual-volumes/<virtual volume name> context for each volume(s) to be copied.

   i. Confirm the underlying device status is not marked ‘out of date’ or in a ‘rebuilding’ state.
   
   ii. Confirm Health Status is ‘ok’
   
   iii. Confirm Operational Status is ‘ok’

   **Note:** Consistency can't be guaranteed if Host I/O is ongoing and a R1 or DR1 is being snapped/claoned

   c. Confirm the source VPLEX virtual volume device geometry is not RAID-C. Device geometry can be determined by issuing ll at the /clusters/<cluster name>/devices/<device name> context.

   d. Confirm each volume is 1:1 mapped (single extent) RAID-0 or 1:1 mapped (two extent) local RAID-1. Distributed RAID-1 device legs must be a combination of RAID-0 (single extent) and/or RAID-1 (two extent) device geometries.

   e. Confirm the device is not being protected by RecoverPoint. Issue the ‘ll’ command from the /clusters/<cluster name>/virtual-volumes/<virtual volume name> context and check ‘recoverpoint-protection-at’ is set to [] and ‘recoverpoint-usage’ is set to ‘-’.

   f. Confirm VPLEX volumes to be copied do not have ‘remote’ locality (from same VPLEX cluster). Issue the ‘ll’ command against the /clusters/<local cluster name>/virtual volumes/<virtual volume name> context and confirm locality is ‘local’ or ‘distributed’.

   g. Ensure virtual volumes are members of the same VPLEX consistency group and the same array-based consistency group (if available). In most cases all members of the consistency group should be copied together. Consistency group membership can be determined by issuing ll from the /clusters/<cluster name>/consistency-groups/<consistency group name> context.
Note: VPLEX consistency group membership should align with array based consistency group membership whenever possible. Some array replication technologies may not have a concept such as a consistency group.

h. For RAID-1 or distributed RAID-1 based virtual volumes, confirm underlying storage volume status is not failed or in an error state. Issue the ‘ll’ command from the /clusters/<cluster name>/devices context or from /distributed-storage/distributed-devices/<distributed device name>/components context.

i. For distributed RAID-1 confirm WAN links are not down. Issue the ‘cluster status’ command and confirm ‘wan-com’ status is ‘ok’ or ‘degraded’. If WAN links are completely down, confirm array based copy is being made at winning site.

Note: Best practice is to set the VPLEX consistency group detach rule (winning site) to match site where array based copies are made.

3. Follow standard array-based procedure(s) to generate desired array-based copies (i.e SnapView, TimeFinder, XtremIO User or CLI guides). The most up to date documentation for EMC products is available at https://support.emc.com.

Once the copy is created on the storage array, it can be presented (zoned and lun masked) directly from the storage array back to a test or backup host. Most often, however, the preference is to present the array based copy back through VPLEX to a host. Since a single array contains the copy(s) / snapshot(s), the resulting virtual volume(s) will always be a local virtual volume(s). For VPLEX presented copies, perform these additional VPLEX specific steps:

4. Confirm the array-based copies are visible to VPLEX BE ports. As necessary, perform storage array to VPLEX masking/storage group modification to add the copies.

5. Perform one-to-one encapsulation through the Unisphere for VPLEX UI or VPLEX CLI:
   a. Claim storage volumes from the storage array containing the copies
   b. Identify the logical units from the array containing the array-based copies. One way to do this is to use the host lun number assigned by the array to each device in the array masking view (VMAX/XtremIO) or storage group (CX4/VNX).
   c. Create VPLEX virtual volumes from the copies using the VPLEX UI or CLI:
      i. Use the ‘Create Virtual Volumes’ button from the Arrays context within the UI or
      ii. Create single extent, single member RAID-0 geometry device, and virtual volume(s) with the ‘storage-tool-compose’ (5.3 and higher) command from the VPLEX CLI

6. Present VPLEX virtual volumes built from array-based copies to host(s)
   a. As necessary, create VPLEX storage view(s)
   b. Add virtual volumes built from array-based copies to storage view(s)
c. As necessary, perform zoning of virtual volumes to hosts following traditional zoning best practices

**Resynchronizing (updating) local array-based copies presented through VPLEX**

![Diagram showing resynchronizing a VPLEX RAID-0 copy](image)

**Figure 5: Resynchronizing a VPLEX RAID-0 copy**

This process follows the traditional best practices for refreshing non-VPLEX array based copies such as those provided by products such as TimeFinder, SnapView, and XtremIO Snapshots. During the resynchronization process, the storage array writes new data to the copy outside of the host and VPLEX IO path. It is very important that the host does not have the copies mounted and in-use. Failure to stop applications and to clear both the host read cache and the VPLEX read cache prior to re-synchronization may lead to data consistency issues on the copies.

**Technical Note:** These steps require using the VPLEX CLI or REST API only.

To re-synchronize (update) VPLEX virtual volumes based on array-based copies:

1. Confirm the device(s) you wish to restore is/are not RecoverPoint protected. Issue the 'll' command from the /clusters/<cluster name>/virtual volumes/<virtual volume name> context. Check 'recoverpoint-protection-at' is set to [] and 'recoverpoint-usage' is set to `-`. If the virtual volume is a member of a consistency group, the consistency group context will indicate if it is RecoverPoint protected or not as well.

2. Confirm a VPLEX ndu is not in progress. Using the VPLEX CLI issue the 'ndu status' command and confirm that the response is 'No firmware, BIOS/POST, or SSD upgrade is in progress.'

3. Shut down any applications using the VPLEX virtual volumes that contain the array-based copies to be re-synchronized. Unmount the associated virtual volumes from the host. The key here is to force the host to invalidate its own read cache and to prevent any new writes during the resynchronization of the array-based copy.

4. Invalidate the VPLEX read cache. There are several options to achieve invalidation depending on your VPLEX GeoSynchrony code version:
   A. For pre-5.2 code, remove the virtual volume(s) from all storage views. Make note of the virtual volume lun numbers within the storage view prior to removing them. You will need this information in step 6 below.

or
B. For 5.2 and higher code,

1. Use **virtual-volume cache-invalidate** to invalidate an individual volume:

   VPlexcli:/> virtual-volume cache-invalidate <virtual volume>

   or

   Use **consistency-group cache-invalidate** to invalidate an entire VPLEX consistency group.

   VPlexcli:/> consistency-group cache-invalidate <consistency group>

2. Follow each command in step 1 with the command **virtual-volume cache-invalidate-status** to confirm the cache invalidation process has completed.

   VPlexcli:/> virtual-volume cache-invalidate-status <virtual volume>

   **Example output for a cache invalidation job in progress:**

   cache-invalidate-status
   ------------------------
   director-1-1-A  status: in-progress
   result: -
   cause: -

   **Example output for a cache invalidation job completed successfully:**

   cache-invalidate-status
   ------------------------
   director-1-1-A  status: completed
   result: successful
   cause: -

**Technical Note 1:** For pre-5.2 code, it is recommended to wait a minimum of 30 seconds to ensure that the VPLEX read cache has been invalidated for each virtual volume. This can be done concurrently with Step 3.

**Technical Note 2:** The **virtual-volume cache-invalidate** commands operate on a single virtual-volume at a time. This is the case even when the consistency group command is used. The **consistency-group cache-invalidate** command will fail if any member virtual volume doesn’t invalidate. Invalidation will fail if host I/O is still in progress to the virtual volumes. The invalidation process can take a non-trivial amount of time (usually not more than a few seconds), so the use of the **virtual-volume cache-invalidate-status** command is recommended to confirm completion of invalidation tasks. With GeoSynchrony 5.4 SP1+
code the cache invalidate command runs in the foreground and does not require a follow up status check. Once the command completes the invalidation is complete.

**Technical Note 3:** Cache-invalidate command must not be executed on the RecoverPoint enabled virtual-volumes. This means copying should be done using RecoverPoint or using array-based copies, but not both for a given virtual volume. The VPLEX Clusters should not be undergoing a NDU while this command is being executed.

5. Identify the source storage volumes within the array you wish to resynchronize. Follow your normal array resynchronization procedure(s) to refresh the desired array-based copies.

6. Confirm the IO Status of storage volumes based on array-based copies is “alive” by doing a long listing against the storage-volumes context for your cluster.

   For example:

   ```
   Vplexcli:/> /clusters/cluster-1/storage-elements/storage-volumes/
   
   /clusters/cluster-1/storage-elements/storage-volumes/
       Name     Vendor   Status  Type
       LUN Id   Capacity  Use
   
   Smw15s5TDev_001
   Smw15s5TDev_002
   Smw15s5TDev_003
   Smw15s5TDev_004
   Smw15s5TDev_005
   
   5. Confirm VPLEX back-end paths are healthy by issuing the “connectivity validate-be” command from the VPLEX CLI. Ensure that there are no errors or connectivity issues to the back-end storage devices. Resolve any error conditions with the back-end storage before proceeding.

   Example output showing desired back-end status:

   ```
   Vplexcli:/clusters/cluster-1> connectivity validate-be
   Summary
   Cluster cluster-2
   This cluster has 0 storage-volumes which are dead or unreachable
   This cluster has 0 storage-volumes which do not have dual paths
   This cluster has 0 storage-volumes which are not visible from all directors
   Cluster cluster-1
   This cluster has 0 storage-volumes which are dead or unreachable
   This cluster has 0 storage-volumes which do not have dual paths
   This cluster has 0 storage-volumes which are not visible from all directors
   ```

6. For pre-5.2 code, restore access to virtual volumes based on array-based copies for host(s). If you removed the virtual volume from a storage view, add the virtual volume back, specifying the original LUN number (noted in step 2) using the VPLEX CLI:

   ```
   /clusters/<cluster name>/exports/storage-views> addvirtualvolume -v
   storage_view_name/ -o (lun#, virtual_volume_name) -f
   ```

7. As necessary, rescan devices and restore paths (for example, powermt restore) on hosts.
8. As necessary, re-mount VPLEX virtual volumes on each host.
9. Restart applications.

**Technical Note:** VPLEX does not alter operating system limitations relative to array-based copies. When presenting an array-based copy back to the same host, you will need to confirm both the host operating system and the logical volume manager support such an operation. If the host OS does not support seeing the same device signature or logical volume manager identifier, VPLEX will not change this situation.
**Advanced array-based copy use cases**

When an array-based copy (e.g. a Snapshot or a full disk copy) is a component of a VPLEX RAID-1 virtual volume, it is important to properly account for both mirror legs to avoid potential data inconsistency. VPLEX is providing the mirroring, but during the copy and resynchronization process the array is writing to one or more of the mirror legs outside of the VPLEX IO path. The key here is that VPLEX is not able to mirror writes unless they pass through its IO stack. For this reason, an additional step must be added to the array-based copy process to temporarily detach the mirror leg that will not be updated by the array. Once the array-based copy process completes, the final step would be to back the leg that was detached in order to trigger a VPLEX RAID-1 rebuild (as shown in Figure 3 below). This step must be done during both the creation and the resynchronization (update) processes.

**Example 3: Create VPLEX RAID-1 Copy or Resynchronize VPLEX RAID-1 Copy**

This section assumes VPLEX is providing RAID-1 device geometry for the array-based copy. The VPLEX copy virtual volume consists of one mirror leg that contains an array-based copy and a second mirror leg that may (Figure 7) or may not (Figure 6) contain array based copy.

**Note:** Array based copies on two different arrays are feasible, but consistency across disparate arrays cannot be assumed due to differences in things such as copy timing and copy performance. That said, some array management software interfaces do have a concept of consistency across arrays. See individual array administrative guides for further details.

For some use cases, it may be desirable to create copies of each mirror leg to protect against the loss of an array or loss of connectivity to an array. In addition, the second copy can be located within the same array or a separate second array. Figure 6 illustrates a local VPLEX device protected using an array-based copy with RAID-1 geometry. This configuration applies to both array-based copies within a local RAID-1 device and to array-based copies that make up distributed (VPLEX Metro) RAID-1 devices. The steps that follow are critical to ensure proper mirror synchronization (for the non-copy leg) and to ensure each virtual volume’s read cache is properly updated.
Prerequisites

This section assumes you are using or planning to use distributed or local RAID-1 VPLEX virtual volumes built with at least one mirror leg that is an array-based copy. In addition, the VPLEX virtual volumes must possess both of the following attributes:

- Be comprised of devices that have a one-to-one storage volume pass-through configuration to VPLEX (device capacity = extent capacity = storage volume capacity).
- Have a single device with single-extent RAID-1 (two single-extent devices being mirrored) geometry.

Creating VPLEX RAID-1 or Distributed RAID-1 array-based copies

This process augments the traditional best practices for non-VPLEX array-based copies using TimeFinder and SnapView. It accounts for VPLEX and for host access through VPLEX to array-based copies.
copy process, the storage array will be writing new data to the array-based copy outside of the VPLEX and the host IO path.

Procedure: Creating VPLEX RAID-1 or Distributed RAID-1 array-based copies

1. For each storage array, identify source storage volumes to copy. The VPLEX storage-view will contain all VPLEX volumes that are visible to a given host or group of hosts. To determine the VPLEX to array device mapping use one of the following methods:
   - Using the VPLEX CLI invoke the ‘show-use-hierarchy’ or ‘device drill-down’ command for the virtual volume(s) and underlying device(s) you wish to make a copy of. These outputs provide the virtual to physical volume mapping for each VPLEX volume.
   - Use the Storage Viewer feature of Virtual Storage Integrator (VSI) for VMware vCenter Server (for ESX environments) to see the virtual device to physical array device mapping.
   - Correlate the VPLEX virtual volumes to physical storage using the VPDID from the virtual volume to map to the back end storage array device. This information is available to the host via SCSI inquiry and from VPLEX via the /clusters/cluster name/export/storage-view context. To get the back-end array device, you can map virtual volume VPDID back through the VPLEX and get the array LUN. From a VPLEX perspective there is a VPDID tied to the virtual volume the host sees and a different VPDID tied to the back end storage array.

2. Using the VPLEX CLI or REST API, check for valid copy conditions to ensure data consistency:

   **Note:** Each of the following checks are typically scripted or built into code that orchestrates the overall copy process on the array.

   a. Confirm a VPLEX ndu is not in progress. Using the VPLEX CLI issue the ‘ndu status’ command and confirm that the response is ‘No firmware, BIOS/POST, or SSD upgrade is in progress.’
   b. Confirm the device is healthy. Issue the ‘ll’ command from the /clusters/cluster name/virtual-volumes/virtual volume name context for each volume(s) to be copied.
      i. Confirm the underlying device status is not marked ‘out of date’ or in a ‘rebuilding’ state.
      ii. Confirm Health Status is ‘ok’
      iii. Confirm Operational Status is ‘ok’
   c. Confirm the underlying device geometry is not RAID-c. Device geometry can be determined by issuing ll at the /clusters/cluster name/devices/device name context.
   d. Confirm each volume is 1:1 mapped (single extent) RAID-0 or 1:1 mapped (two extent) local RAID-1. Distributed RAID-1 device legs must be a combination of RAID-0 (single extent) and/or RAID-1 (two extent) device geometries.
e. Confirm the device is not being protected by RecoverPoint. Issue the ‘ll’ command from the /clusters/<cluster name>/virtual-volumes/<virtual volume name> context and check ‘recoverpoint-protection-at’ is set to [] and ‘recoverpoint-usage’ is set to ‘‘.

f. Confirm VPLEX volumes to be copied have the same locality (from same VPLEX cluster).

g. Ensure virtual volumes are members of the same VPLEX consistency group and the same array based consistency group (if available). In most cases all members of the consistency group should be copied together. Consistency group membership can be determined by issuing ll from the /clusters/<cluster name>/consistency-groups/<consistency group name> context. The array CLI or API should be used to determine the array’s consistency group membership.

Note: VPLEX consistency group membership should align with array based consistency group membership whenever possible.

h. For RAID-1 or distributed RAID-1 based virtual volumes, confirm underlying storage volume status is not failed or in an error state. Issue the ‘ll’ command from the /clusters/<cluster name>/devices context or from /distributed-storage/distributed-devices/<distributed device name>/components context.

i. For distributed RAID-1 confirm WAN links are not down. Issue the ‘cluster status’ command and confirm ‘wan-com’ status is ‘ok’ or ‘degraded’. If WAN links are completely down, confirm array based copy is being made at winning site.

Note: Best practice is to set the consistency group detach rule (winning site) to match site where array based copy is made.

3. Follow standard array-based procedure(s) to generate desired array-based copies. For example, use the TimeFinder, SnapView, or XtremIO Snapshots cli documentation. The most up to date versions of these documents are available at http://support.emc.com

Perform VPLEX specific steps:

4. Confirm the array-based copy (Clone/Snapshot) devices are visible to VPLEX. As necessary, perform storage array to VPLEX LUN masking and storage array to VPLEX SAN zoning.

5. Perform one-to-one encapsulation through the Unisphere for VPLEX UI or VPLEX CLI:
   a. Claim storage volumes from the storage array containing the copies
   b. Identify the logical units from the array containing the array-based copies. One way to do this is to use the host lun number assigned by the array to each device in the array masking view (Symmetrix/VMAX) or storage group (CX4/VNX).
   c. Create VPLEX virtual volumes from the copies using the VPLEX UI or CLI:
      i. Use the ‘Create Virtual Volumes’ button from the Arrays context within the UI
or

ii. Create single extent, single member RAID-0 geometry device, and virtual volume(s) from the VPLEX CLI

d. Add local or remote mirrors to virtual volumes created in the previous step using the Unisphere for VPLEX UI or the VPLEX CLI. See https://support.emc.com for the latest VPLEX CLI and Users guides for further details.

6. Present VPLEX virtual volumes built from array-based copies to host(s)

   a. As necessary, create VPLEX storage view(s)

   b. Add virtual volumes built from array-based copies to storage view(s)

   c. As necessary, perform zoning of virtual volumes to hosts following traditional zoning best practices

Procedure: Resynchronizing (refreshing) array-based copies of RAID-1 VPLEX virtual volumes

1. Confirm the device(s) you wish to resynchronize is/are not RecoverPoint protected. Issue the ‘ll’ command from the /clusters/<cluster name>/virtual volumes/<virtual volume name> context. Check ‘recoverpoint-protection-at’ is set to [] and ‘recoverpoint-usage’ is set to ‘-’. If the virtual volume is a member of a consistency group, the consistency group context will indicate if it is RecoverPoint protected or not as well.

2. Confirm a VPLEX ndu is not in progress. Using the VPLEX CLI issue the ‘ndu status’ command and confirm that the response is ‘No firmware, BIOS/POST, or SSD upgrade is in progress.’

3. Shut down any applications accessing the VPLEX virtual volumes that contain the array-based copies to be resynchronized. This must be done at both clusters if the virtual volumes are being accessed across multiple sites. If necessary, unmount the associated virtual volumes from the host(s). The key here is to force the hosts to invalidate their read cache and to prevent any new writes during the resynchronization of the array-based copy.

4. Invalidate the VPLEX read cache. There are two options to achieve invalidation depending on your VPLEX GeoSynchrony code version:

   A. For pre-5.2 code, remove the virtual volume(s) from all storage views.

      1. If the virtual volume is built from a local RAID-1 device and/or is a member of a single storage view, using the VPLEX CLI run:

         /clusters/<cluster name>/exports/storage-views>
         removevirtualvolume -v storage_view_name -o virtual_volume_name -f

      2. If the virtual volume is built from a distributed device and is a member of storage views in both clusters, using the VPLEX CLI run:
/clusters/<local cluster name>/exports/storage-views>
removevirtualvolume -v storage_view_name -o
distributed_device_name_vol -f

/clusters/<remote cluster name>/exports/storage-views>
removevirtualvolume -v storage_view_name -o
distributed_device_name_vol -f

Make note of the virtual volume lun numbers within the VPLEX storage view prior to removing them. You will need this information in step 7 below.

Or

B. For 5.2 and higher code,

1. Use `virtual-volume cache-invalidate` to invalidate an individual volume:
   
   VPlexcli:/> virtual-volume cache-invalidate <virtual volume>
   
   or

   Use `consistency-group cache-invalidate` to invalidate an entire VPLEX consistency group.
   
   Vpexcli:/> consistency-group cache-invalidate <consistency group>

2. Follow each command in step 1 with the command `virtual-volume cache-invalidate-status` to confirm the cache invalidation process has completed.

   Vpexcli:/> virtual-volume cache-invalidate-status <virtual volume>

Example output for a cache invalidation job in progress:

   cache-invalidate-status
   ------------------------
   director-1-1-A status: in-progress
   result: -
   cause: -

Example output for a cache invalidation job completed successfully:

   cache-invalidate-status
   ------------------------
   director-1-1-A status: completed
   result: successful
   cause: -
Technical Note 1: For pre-5.2 code, it is recommended to wait a minimum of 30 seconds to ensure that the VPLEX read cache has been invalidated for each virtual volume. This can be done concurrently with Step 3.

Technical Note 2: The virtual-volume cache-invalidate commands operate on a single virtual-volume at a time. This is the case even when the consistency group command is used. The consistency-group invalidate-cache command will fail if any member virtual volume hasn’t been invalidated. Invalidation will fail if host I/O is still in progress to the virtual volumes. The invalidation process can take a non-trivial amount of time, so the use of the virtual-volume cache-invalidate-status command is recommended to confirm completion of invalidation tasks. With GeoSynchrony 5.4 SP1+ code the cache invalidate command runs in the foreground and does not require a follow up status check. Once the command completes the invalidation is complete.

Technical Note 3: Cache-invalidate command must not be executed on the Recover Point enabled virtual-volumes. This means using either RecoverPoint or array-based copies, but not both for any given virtual volume. The VPLEX Clusters should not be undergoing a NDU while this command is being executed.

5. Detach the VPLEX device mirror leg that will *not* be updated during the array-based replication or resynchronization processes:
   ```
   device detach-mirror -m <device_mirror_to_detach> -d <distributed_device_name> -i -f
   ```

6. Perform resynchronization (update) of array-based copies using standard TimeFinder, SRDF, SnapView, MirrorView, or XtremIO administrative procedures. See [http://support.emc.com](http://support.emc.com) for the latest copies of EMC product documentation.

7. Confirm the IO Status of storage volumes based on array-based copies is “alive” by doing a long listing against the storage-volumes context for your cluster.

   For example:

   ```
   VPLEXCLI://> li /clusters/cluster-1/storages/elements/storage-volumes/
   /clusters/cluster-1/storages/elements/storage-volumes:
   Name | VPLEX 3B | Capacity | Use | Vendor | IO Status | Type
   ---- | --------- | -------- | ---- | ------ | ---------- | ----
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 | VF0E7E16089D900001280155545339089324 | 100 | used | EMC | alive | normal
   Symm5SmTv_e816 |VF0E7E16089D900001280155545339089324 |100 | used | EMC | alive | normal
   ```

   In addition, confirm VPLEX back-end paths are healthy by issuing the “connectivity validate-be” command from the VPLEX CLI. Ensure that there are no errors or connectivity issues to the back-end storage devices. Resolve any error conditions with the back-end storage before proceeding.

   Example output showing desired back-end status:
8. Reattach the second mirror leg:

**RAID-1:**

```
device attach-mirror -m <2nd mirror leg to attach> -d /clusters/<local cluster name>/devices/<existing RAID-1 device>
```

**Distributed RAID-1:**

```
device attach-mirror -m <2nd mirror leg to attach> -d /clusters/<cluster name>/devices/<existing distributed RAID-1 device>
```

**Technical Note:** The device you are attaching is the non-copy mirror leg. It will be overwritten with the data from the copy. Depending on the amount of data being resynchronized this process can take quite some time. This should be factored into the RTO consideration.

9. For pre-5.2 GeoSynchrony code, restore host access to VPLEX volume(s).

If the virtual volume is built from a local RAID 1 device:

```
/clusters/<local cluster name>/exports/storage-views> addvirtualvolume -v storage_view_name/ -o (lun#,device_Symm0191_065_1_vol/) -f
```

If the virtual volume is built from a distributed RAID 1 device:

```
/clusters/<remote cluster name>/exports/storage-views> addvirtualvolume -v storage_view_name/ -o (lun#, distributed_device_name_vol) -f
/clusters/<local cluster name>/exports/storage-views> addvirtualvolume -v storage_view_name/ -o (lun#, distributed_device_name_vol) -f
```

The `lun#` is the previously recorded value from step 2 for each virtual volume.

**Technical Note 1:** For pre-5.2 GeoSynchrony code, EMC recommends waiting at least 30 seconds after removing access from a storage view to restore access. Waiting ensures that the VPLEX cache has been cleared for the volumes. The array-based resynchronization will likely will take 30 seconds, but if you are scripting, be sure to add a pause prior to performing this step.

**Technical Note 2:** Some hosts and applications are sensitive to LUN numbering changes. Use the information you recorded in step 2 to ensure the same LUN numbering for the host(s) when the virtual volume access is restored.
Technical Note 3: It is not necessary to perform full mirror synchronization prior to restoring access to virtual volumes. VPLEX will synchronize the second mirror leg in the background while using the first mirror leg as necessary to service reads to any unsynchronized blocks.

10. Rescan devices and restore paths (powermt restore) on hosts.
11. Mount devices (if mounts are used).
12. Restart applications.
Array-based Restore for VPLEX RAID-0 Production Volumes

The array-based restore or recovery process with VPLEX is similar to the previous array-based resynchronization use cases. The primary difference is that the array is now writing data from the copy/snapshot back to the production or source volume. In this section we review the array-based restore process. It is assumed the array-based copy / snapshot is located on the array containing the source device (restore target). For the basic local restore use case, each source (restore target) device must have a one-to-one storage volume pass-through configuration (device capacity = extent capacity = storage volume capacity) through VPLEX and a RAID-0 *(single extent only)* device geometry. This is the most basic restore use case for restore and recovery. Error! Reference source not found. illustrates the case where data is being written within a storage array from a copy (Snapshot or Full disk copy) or backup media to a storage volume (standard device) that is used by VPLEX.

**Example 4: Restore From Array-based Copy to VPLEX RAID-0 Production Volume**

**Procedure: Array-based Restore to VPLEX RAID-0 Production Virtual volumes**

1. Confirm the production storage volume(s) you wish to restore is/are not RecoverPoint protected. Issue the ‘ll’ command from the /clusters/<cluster name>/virtual volumes/<virtual volume name> context. Check ‘recoverpoint-protection-at’ is set to [] and ‘recoverpoint-usage’ is set to ‘-’. If the virtual volume is a member of a consistency group, the consistency group context will indicate if it is RecoverPoint protected or not as well.

2. Confirm a VPLEX ndu is not in progress. Using the VPLEX CLI issue the ‘ndu status’ command and confirm that the response is ‘No firmware, BIOS/POST, or SSD upgrade is in progress.’

3. Confirm consistency group membership and dependent virtual volume relationships. In other words, confirn that you are restoring all of the volumes that the application requires to run and not a subset of all volumes.

4. Shut down any host applications using the source VPLEX volume(s) that will be restored. As necessary, unmount the associated virtual volumes on the host. The objectives here are to prevent host access and to clear the host’s read cache.

5. Invalidate VPLEX read cache on the source virtual volume(s). There are several options to achieve VPLEX read cache invalidation depending on your VPLEX GeoSynchrony code version:
   - A. For pre-5.2 code, remove the source virtual volume(s) from all storage views. Make note of the virtual volume lun numbers within the storage view prior to removing them. You will need this information in step 5 below.
B. For 5.2 and higher code,

1. Use **virtual-volume cache-invalidate** to invalidate the individual source volume(s):

   VPLEXcli:/> virtual-volume cache-invalidate <virtual volume>

   or

   Use **consistency-group cache-invalidate** to invalidate an entire VPLEX consistency group of source volumes.

   VPLEXcli:/> consistency-group cache-invalidate <consistency group>

2. Follow each command in step 1 with the command **virtual-volume cache-invalidate-status** to confirm the cache invalidation process has completed.

   VPLEXcli:/> virtual-volume cache-invalidate-status <virtual volume>

Example output for a cache invalidation job in progress:

```
cache-invalidate-status
-----------------------
director-1-1-A         status: in-progress
                      result: -
                      cause: -
```

Example output for a cache invalidation job completed successfully:

```
cache-invalidate-status
-----------------------
director-1-1-A         status: completed
                      result: successful

                      cause: -
```

**Technical Note 1:** For pre-5.2 code, it is recommended to wait a minimum of 30 seconds to ensure that the VPLEX read cache has been invalidated for each virtual volume. This can be done concurrently with Step 3.

**Technical Note 2:** The **virtual-volume cache-invalidate** commands operate on a single virtual-volume at a time. This is the case even when the consistency group command is used. The **consistency-group cache-invalidate** command will fail if any member virtual volume doesn't invalidate. Invalidation will fail if host I/O is still in progress to the virtual volumes. The invalidation process can take a non-trivial amount of time, so the use of the **virtual-volume**
**cache-invalidate-status** command is recommended to confirm completion of invalidation tasks. With GeoSynchrony 5.4 SP1+ code the cache invalidate command runs in the foreground and does not require a follow up status check. Once the command completes the invalidation is complete.

**Technical Note 3:** Cache-invalidate command must not be executed on the RecoverPoint enabled virtual-volumes. This means using either RecoverPoint or array-based copies, but not both, for any given virtual volume. The VPLEX Clusters should not be undergoing a NDU while this command is being executed.

6. Identify the copy (BCV/Clone/Snapshot) to source device pairing(s) within the array(s). For EMC products, follow the TimeFinder, SnapView, SRDF, MirrorView, XtremIO Snapshot restore procedure(s) to restore data to the desired source devices. See [http://support.emc.com](http://support.emc.com) for the latest EMC product documentation for TimeFinder, SnapView, SRDF, MirrorView, or XtremIO Snapshots.

7. Confirm the IO Status of the source storage volumes within VPLEX is “alive” by doing a long listing against the storage volumes context for your cluster.

For example:

```
TLVplexcli:/> 11 /clusters/cluster-1/storage-elements/storage-volumes/
/cluster/cluster-1/storage-elements/storage-volumes:
Name: Capacity: Type: Status: Total
symv54YRev_0610 7100000000000000000000000000000000000000000
symv54YRev_0611 7100000000000000000000000000000000000000000
symv54YRev_0612 7100000000000000000000000000000000000000000
symv54YRev_0613 7100000000000000000000000000000000000000000
symv54YRev_0614 7100000000000000000000000000000000000000000
symv54YRev_0615 7100000000000000000000000000000000000000000
symv54YRev_0616 7100000000000000000000000000000000000000000
```

In addition, confirm VPLEX back-end paths are healthy by issuing the “connectivity validate-be” command from the VPLEX CLI. Ensure that there are no errors or connectivity issues to the back-end storage devices. Resolve any error conditions with the back-end storage before proceeding.

Example output showing desired back-end status:

```
Vplexcli:/clusters/cluster-1> connectivity validate-be
Summary
Cluster cluster-2
This cluster has 0 storage-volumes which are dead or unreachable
This cluster has 0 storage-volumes which do not have dual paths
This cluster has 0 storage-volumes which are not visible from all directors
Cluster cluster-1
This cluster has 0 storage-volumes which are dead or unreachable
This cluster has 0 storage-volumes which do not have dual paths
This cluster has 0 storage-volumes which are not visible from all directors
```

8. For Pre 5.2 code, restore access to virtual volume(s) based on source devices for host(s): Add the virtual volume back to the view, specifying the original LUN number (noted in step 2) using VPLEX CLI:

```
/clusters/<cluster name>/exports/storage-views> addvirtualvolume -v storage_view_name/ -o (lun#, virtual_volume_name) -f
```
9. As necessary, rescan devices and restore paths (for example, powermt restore) on hosts.
10. As necessary, mount devices.
11. Restart applications.

**Note:** Even though applications are able to run during resync, the standard RTO is not delivered until both mirror legs are completely synchronize.
Array-based Restore of VPLEX RAID-1 Production Volumes

When VPLEX virtual volumes have RAID-1 geometry, the restore process must take into account the second (non-restored) mirror leg. This applies for both local RAID-1 and distributed (VPLEX Metro) RAID-1 VPLEX devices. The typical array-based source device restore procedure will only restore one of the two mirror legs of a VPLEX RAID 1 device. In order to synchronize the second VPLEX device, users need to add a step to resynchronize the 2nd mirror leg. These steps are critical to ensure proper synchronization of the second VPLEX RAID-1 mirror leg (the one that is not being restored by the array) and to ensure each virtual volume’s read cache is properly updated. Example 8 (below) illustrates the 5-step workflow when an array-based copy is used to restore to a RAID-1 or distributed RAID-1 source volume. RAID-

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pre-Restore</strong></td>
<td><strong>Restore Preparation: Detach Mirror Leg</strong></td>
</tr>
<tr>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Active Restore</strong></td>
<td><strong>Post Restore Rebuild</strong></td>
<td><strong>Post Restore Complete</strong></td>
</tr>
<tr>
<td><img src="image3" alt="Diagram" /></td>
<td><img src="image4" alt="Diagram" /></td>
<td><img src="image5" alt="Diagram" /></td>
</tr>
</tbody>
</table>

**Example 5: Array-based Restore of VPLEX RAID-1 Production Volumes**

**Technical Note:** This same set of steps can be applied to remote array-based copy products like SRDF or MirrorView. For example, an SRDF R2 or MirrorView Secondary Image is essentially identical in function to local array-based copy. The remote copy, in this case, can be used to do a restore to a production (R1/Primary Image) volume.

**Prerequisites**

This section assumes users have existing distributed or local RAID-1 VPLEX virtual volumes built from the array source devices being restored. In addition, the VPLEX virtual volumes must possess both of the following attributes:
- The volumes must comprise of devices that have a one-to-one storage volume pass-through VPLEX configuration (device capacity = extent capacity = storage volume capacity).
- The volumes must have a single-extent RAID-1 (two single extents being mirrored) geometry. Distributed, local, and distributed on top of local RAID-1 all meet this pre-requisite.

Procedure: Array-based restore to VPLEX RAID-1 production virtual volumes

1. Using the VPLEX CLI or REST API, check for valid restore conditions to ensure data consistency:

   **Note:** Each of the following checks are typically scripted or built into code that orchestrates the overall restore process on the array.

   a. Confirm a VPLEX ndu is not in progress. Using the VPLEX CLI issue the ‘ndu status’ command and confirm that the response is ‘*No firmware, BIOS/POST, or SSD upgrade is in progress.*’

   b. Confirm the restore target device is healthy. Issue the ‘ll’ command from the /clusters/<cluster name>/virtual-volumes/<virtual volume name> context for each volume(s) to be copied.

      i. Confirm the underlying device status is not marked ‘out of date’ or in a ‘rebuilding’ state.

      ii. Confirm Health Status is ‘ok’

      iii. Confirm Operational Status is ‘ok’

   c. Confirm the underlying restore target device geometry is **not** RAID-c. Device geometry can be determined by issuing ll at the /clusters/<cluster name>/devices/<device name> context.

   d. Confirm each volume is 1:1 mapped (single extent) RAID-0 or 1:1 mapped (two extent) local RAID-1. Distributed RAID-1 device legs must be a combination of RAID-0 (single extent) and/or RAID-1 (two extent) device geometries.

   e. Confirm the restore target device is **not** being protected by RecoverPoint. Issue the ‘ll’ command from the /clusters/<cluster name>/virtual-volumes/<virtual volume name> context and check ‘recoverpoint-protection-at’ is set to [] and ‘recoverpoint-usage’ is set to ‘-’.

   f. Confirm VPLEX volumes to be restore have the same locality (from same VPLEX cluster) and are in the same array.

   g. Ensure virtual volumes are members of the same VPLEX consistency group and, when possible, part of a similar array-based consistency group construct within the array. In most cases all members of the consistency should be copied together. Consistency group membership can be determined by issuing ll from the /clusters/<cluster name>/consistency-groups/<consistency group name> context. Array based consistency group membership can be determined using the appropriate array cli and/or api.

   h. For RAID-1 or distributed RAID-1 based virtual volumes, confirm underlying storage volume status is not failed or in an error state. Issue the ‘ll’ command from the /clusters/<cluster
name>/devices context or from /distributed-storage/distributed-devices/<distributed device name>/components context.

i. For distributed RAID-1 confirm WAN links are not down. Issue the ‘cluster status’ command and confirm ‘wan-com’ status is ‘ok’ or ‘degraded’. If WAN links are completely down, confirm array based restore is being done at winning site.

**Note:** Best practice is to set the consistency group detach rule (winning site) to match site where array based restore is being done.

2. Shut down any host applications (both local and remote with DR1) using the source VPLEX volume(s) that will be restored. Unmount the associated virtual volumes on the host. The objectives here are to prevent host access and to clear the host’s read cache.

3. Invalidate VPLEX read cache on the source virtual volume(s). There are several options to achieve VPLEX read cache invalidation depending on your VPLEX GeoSynchrony code version:

   A. For pre-5.2 code, remove the source virtual volume(s) from all storage views. Make note of the virtual volume lun numbers within the storage view prior to removing them. You will need this information in step 7 below.

   or

   B. For 5.2 and higher code,

   1. Use **virtual-volume cache-invalidate** to invalidate the individual source volume(s):

      VPLEXcli:/> virtual-volume cache-invalidate <virtual volume>

      or

      Use **consistency-group cache-invalidate** to invalidate an entire VPLEX consistency group of source volumes.

      VPLEXcli:/> consistency-group cache-invalidate <consistency group>

   2. Follow each command in step 1 with the command **virtual-volume cache-invalidate-status** to confirm the cache invalidation process has completed.

      VPLEXcli:/> virtual-volume cache-invalidate-status <virtual volume>

      Example output for a cache invalidation job in progress:

      ```
      cache-invalidate-status
      -----------------------
      director-1-1-A         status: in-progress
                               result: -
                               cause: -
      ```
Example output for a cache invalidation job completed successfully:

```
cache-invalidate-status
-----------------------
director-1-1-A status: completed
result: successful
cause: -
```

**Technical Note 1:** For pre-5.2 code, it is recommended to wait a minimum of 30 seconds to ensure that the VPLEX read cache has been invalidated for each virtual volume. This can be done concurrently with Step 3.

**Technical Note 2:** The virtual-volume cache-invalidate commands operate on a single virtual-volume at a time. This is the case even when the consistency group command is used. The consistency-group invalidate-cache command will fail if any member virtual volume doesn't invalidate. Invalidation will fail if host I/O is still in progress to the virtual volumes. The invalidation process can take a non-trivial amount of time, so the use of the virtual-volume cache-invalidate-status command is recommended to confirm completion of invalidation tasks. With GeoSynchrony 5.4 SP1+ code the cache invalidate command runs in the foreground and does not require a follow up status check. Once the command completes the invalidation is complete.

**Technical Note 3:** Cache-invalidate command must not be executed on the Recover Point enabled virtual-volumes. This means using either RecoverPoint or array-based copies, but not both, for a given virtual volume. The VPLEX Clusters should not be undergoing a NDU while this command is being executed.

4. Detach the VPLEX device RAID-1 or Distributed RAID-1 mirror leg that will not be restored during the array-based restore processes. If the virtual volume is a member of a consistency group, in some cases the virtual volume may no longer have storage at one site which may cause the detach command to fail. In this case the virtual volume will need to be removed from the consistency group *before* it the mirror leg is detached. Use the detach-mirror command to detach the mirror leg(s):

```
device detach-mirror -m <device_mirror_to_detach> -d <distributed_device_name> -i -f
```

**Note:** Depending on the raid geometry for each leg of the distributed device, it may be necessary to detach both the local mirror leg and the remote mirror leg. This is because only 1 storage volume is being restored and there are up to 3 additional mirrored copies maintained by VPLEX (1 local and 1 or 2 remote). For example, if the VPLEX distributed device mirror leg being restored is, itself, a RAID-1 device then both the non-restored local leg and the remote leg must be detached.
5. Identify the copy (BCV/Clone/Snapshot) to source device pairings within the array(s). Follow the TimeFinder, SnapView, SRDF, MirrorView, XtremIO Snapshot restore procedure(s) to restore data to the desired source devices. See [http://support.emc.com](http://support.emc.com) for the latest EMC product documentation for TimeFinder, SnapView, SRDF, MirrorView, or XtremIO for more details.

6. Confirm the IO Status of storage volumes based on array-based clones is “alive” by doing a long listing against the storage volumes context for your cluster.

For example:

```
$ ls /clusters/cluster-1/storage-elements/storage-volumes/

Name            VFDE3 ID       Capacity  Use  Vendor IO Status
----------------- --------------- ---------- ------ -------- ------------
Symm1557dev_D620 VFDE3T1:6000097000192401554553003163194 100G     used  EMC  alive  normal
Symm1557dev_D621 VFDE3T1:6000097000192401554553003163195 100G     claimed EMC  alive  normal
Symm1557dev_D622 VFDE3T1:6000097000192401554553003163196 100G     claimed EMC  alive  normal
Symm1557dev_D623 VFDE3T1:6000097000192401554553003163197 100G     claimed EMC  alive  normal
Symm1557dev_D624 VFDE3T1:6000097000192401554553003163198 100G     claimed EMC  alive  normal
Symm1557dev_D625 VFDE3T1:6000097000192401554553003163199 100G     claimed EMC  alive  normal
```

In addition, confirm VPLEX back-end paths are healthy by issuing the “connectivity validate-be” command from the VPLEX CLI. Ensure that there are no errors or connectivity issues to the back-end storage devices. Resolve any error conditions with the back-end storage before proceeding.
Example output showing desired back-end status:

```
Vplexcli:/clusters/cluster-1> connectivity validate-be
Summary
Cluster cluster-2
  This cluster has 0 storage-volumes which are dead or unreachable
  This cluster has 0 storage-volumes which do not have dual paths
  This cluster has 0 storage-volumes which are not visible from all directors
Cluster cluster-1
  This cluster has 0 storage-volumes which are dead or unreachable
  This cluster has 0 storage-volumes which do not have dual paths
  This cluster has 0 storage-volumes which are not visible from all directors
```

7. Reattach the second mirror leg:

    **RAID-1:**
    
    device attach-mirror -m <2nd mirror leg to attach> -d /clusters/<local
    cluster name>/devices/<existing RAID-1 device>

    **Distributed RAID-1:**
    
    device attach-mirror -m <2nd mirror leg to attach> -d /clusters/<cluster
    name>/devices/<existing distributed RAID-1 device>

    **Note:** The device you are attaching in this step will be overwritten with the data from the newly restored source device.

8. For Pre 5.2 code, restore host access to the VPLEX volume(s).

    If the virtual volume is built from a local RAID 1 device:
    
    `/clusters/<local cluster name>/exports/storage-views> addvirtualvolume
    -v storage_view_name/ -o (lun#,device_Symm0191_065_1_vol/) -f`

    If the virtual volume is built from a distributed RAID 1 device:
    
    `/clusters/<remote cluster name>/exports/storage-views> addvirtualvolume
    -v storage_view_name/ -o (lun#, distributed_device_name_vol)-f`

    `/clusters/<local cluster name>/exports/storage-views> addvirtualvolume
    -v storage_view_name/ -o (lun#, distributed_device_name_vol)-f`

    The `lun#` is the previously recorded value from step 2 for each virtual volume.

    **Technical Note:** EMC recommends waiting at least 30 seconds after removing access from a storage view to restore access. This is done to ensure that the VPLEX cache has been cleared for the volumes. The array-based restore will likely will take 30 seconds, but if you are scripting be sure to add a pause.

    **Technical Note:** Some hosts and applications are sensitive to LUN numbering changes. Use the information you recorded in step 3 to ensure the same LUN numbering when you restore the virtual volume access.
Technical Note: Full mirror synchronization is not required prior to restoring access to virtual volumes. VPLEX will synchronize the second mirror leg in the background while using the first mirror leg as necessary to service reads to any unsynchronized blocks.

9. Rescan devices and restore paths (powermt restore) on hosts.
10. Mount devices (if mounts are used).
11. Restart applications.
Section 2: VPLEX Native Copy Capabilities

Scope and limitations
Since its initial release in May of 2010, VPLEX has had the ability to move and to mirror storage volumes both locally (VPLEX Local and Metro) and remotely (VPLEX Metro). From a pure technical point of view, this means that VPLEX is able to make a copy or clone of one storage volume onto another. Though not directly exposed as a full-featured copy mechanism, this functionality can become useful in many situations. As of VPLEX GeoSynchrony version 5.2, the disk copy capabilities provided by VPLEX are subject to the following limitations:

- Full synchronization and restore only (no incremental copy or restore)
- No multi-disk concurrent / multiple application consistent split capability across multiple virtual volumes unless the application(s) are quiesced.
- No Snapshots, but Thin to Thin copy is supported.
- Manual administration and identification of source and copy volumes.

Please consult with your local EMC support representative if you are uncertain as to the applicability of these procedures to your VPLEX environment.

Introduction
This section focuses on best practices and key considerations for leveraging VPLEX native copy technologies within VPLEX Local or Metro environments. These copy features and capabilities are complementary to, and not a replacement for, array-based copy technologies (e.g. TimeFinder, SnapView, SRDF, MirrorView, or XtremIO Snapshots). The VPLEX copy capabilities highlighted in this whitepaper are best suited for use cases that require one or more of the following:

- A copy of single-volume applications running on VPLEX Local or Metro platforms
- Single-volume copies between heterogeneous storage arrays
- A copy of a volume from one datacenter to be made available in a 2nd data center.
- Multi-volume copies where the application can be temporarily quiesced prior to splitting the copy.
- Crash consistent copies of sets of volumes
- Consistent copies of groups of volumes obtained by quiescing applications

Independent full disk copies provide important functionality in areas like application development and testing, data recovery, and data protection. As such, it is important to be aware of the copy features within VPLEX and where VPLEX provides advantages over physically and/or geographically constrained array-based copy technologies.
EMC VPLEX Native Copy Overview

Creating independent virtual volume copies with VPLEX can be accomplished using both the UniSphere for VPLEX UI and the VPLEX CLI. The high level steps for each are the same. Figures 9a and 9b below illustrate the steps in the process of creating a VPLEX native clone device.

Creating native copies with VPLEX consists of several steps:

1. Select the source virtual volume and its underlying device(s).
2. Select the copy (target) device with the desired characteristics.
3. Create a data mobility session consisting of the source device and the copy device. As part of this step VPLEX creates a temporary mirror to facilitate the copy.

Figure 9a: VPLEX native clone workflow
4. Synchronize the source and copy (target) devices.
   a. (Optional) Perform host and/or application procedures to ensure desired consistency level.

5. Confirm 100% synchronization and split source and copy device using the data mobility cancel command.

6. Make copy available to host(s).

*Figure 9b: VPLEX native copy workflow*
EMC VPLEX native copying with Unisphere for VPLEX

Creating native copies of VPLEX virtual volumes

VPLEX native copy can be accomplished from the Management Console using the Mobility Central tab located at the top right section of the main window. Even though the functionality used in this section is intended for mobility jobs, the underlying technology is creating full disk copies. The copies that are produced are identical to any other copy. The mobility procedures used to create copies with VPLEX are repeatable and fulfill the requirements of use cases mentioned earlier. Although the following example shows a single copy being created, multiple copies can also be concurrently created.

The steps to create a VPLEX native copy using the Unisphere for VPLEX UI are as follows:

1. Select the Mobility Central tab within the VPLEX Management Console.
2. Confirm the correct cluster name is shown in the For Cluster pull down.
3. Click on Create Device Mobility Job

The next few steps allow the user to specify either the source virtual volume or the corresponding source device that makes up the virtual volume that you wish to make a copy of.
4. (Optional) Select the source virtual volume you wish to make a copy of.

5. If you provided a source virtual volume in optional Step 3 then click the Add button. Confirm the source volume is listed in the Selected Virtual Volumes column.

6. Click Next.
7. Specify the source device you wish to create a copy of. This will be the underlying device for the virtual volume you wish to copy.
8. Click Add. Confirm the correct source device is listed in the Selected Devices column.
9. Click Next.
10. Select the target device to use as your copy. You will want to select a copy device with equal or greater capacity as the source device. In addition, you can select a copy device from the same array or a different array during this step.

11. Click Add Mapping. This step finalizes the relationship between the source devices and target devices you have selected.

12. Verify Source and Copy devices are as desired. Double check that the correct relationship is being created.

13. Click Next.

**Note:** The VPLEX mobility facility supports up to 25 concurrent device mobility sessions. More than 25 mobility jobs can be queued, but at any point in time only 25 will be actively synchronizing. In addition, once synchronization has been completed for each pair of devices, they will remain in sync until the next steps are performed by the script or by the administrator.
Now that the source and copy devices have mapped to one another, the next steps in the setup of the mobility job will allow us to track the synchronization progress, set the sync speed, monitor the completed sync state, and to initiate the split of the copy device from the source device.

14. Enter a descriptive name for the VPLEX native copy mobility job. Note, we are calling it VNC_Job_1 in this example.

15. Click Apply. This will place the name you selected in the Device Mobility Job Name column.

16. Set the Transfer Speed. This controls the speed of synchronization and impact of the copy synchronization process. It is recommended to start with one of the lower Transfer Speed settings and then work your way up as you become more familiar with the impact of the synchronization process on your application and are better able to assess the impact of this setting.

17. Click start to begin the synchronization between the source and copy devices you specified.

After clicking ok on the subsequent confirmation pop-up you will now see your VPLEX native copy mobility job listed in the In Progress section of the Mobility Central screen shown below. Sync progress can be monitored here.
EMC VPLEX Management Console

Mobility Central

Your central location for creating, viewing, and managing mobility jobs. **Learn more...**

**Show:** All Mobility Jobs

**For Cluster:**

- **Queued:**
  - **Name:** VNC Job 1.0
  - **Transfer Speed:** Highest
  - **Progress:** 0%
  - **Type:** Do

- **In Progress:**

- **Commit Pending:**
  - **Name:**
  - **Type:**

- **Errors:**
  - **Name:**
  - **Type:**

- **Canceled:**
  - **Name:**
  - **Type:**

- **Completed:**
  - **Name:**
  - **Type:**

**Create Entire Mobility Jobs**

**Create Device Mobility Jobs**
The next phase in the VPLEX Native Copy process is to split the copy device from the source device. This is accomplished by cancelling the data mobility job while it is in a Commit Pending state. Commit Pending is the state when both the source and target devices are in sync. This state is permanent until the job is cancelled. New writes to the source device will be mirrored to the copy device as long as this state is maintained. Once the mobility job is cancelled the virtual volume will revert to its original configuration and the copy device leg will be detached and placed in the device context. There will be no upper level virtual volume corresponding with the copy device which keeps it safe from new writes.

The individual steps to split the copy device are:

1. Confirm the mobility job is in the Commit Pending Column and that the Status shows ‘complete’.
2. When ready to split the copy device from the source device click the 'Cancel' button. Cancelling the mobility job reverts the source virtual volume back to its original state and splits of the copy device into a usable copy of the source device.
3. Confirm you are cancelling (splitting) the correct VNC mobility job.
4. Click OK to finalize and start the split process.
After the ok button is clicked there will be a confirmation pop-up showing the successful completion of your mobility job cancellation.

The third column of the Mobility Central window will show all cancelled mobility jobs. You can click on the hyperlink to get details on the about the source and copy devices that participated in this mobility job.
The copy device is now available in the **device** context in the Unisphere for VPLEX UI. If you wish to make the VPLEX Native copy device available to a host you must create a virtual volume on top of it. From the Devices context in the VPLEX Management Console click on **Create Virtual Volumes** as shown above and follow the normal steps to create and provision a virtual volume.
Restoring from VPLEX native copies

Restoring from VPLEX native copies can be accomplished from the Unisphere for VPLEX UI using the Mobility Central tab located at the top right section of the main window. It is important to be cognizant of the fact that the functionality used in this section is intended for mobility jobs and not specifically for copy operations. Even so, the results obtained are consistent, predictable, and in keeping with the use cases mentioned earlier.

Although our example shows a single source being restored from a copy, multiple source volumes can also be concurrently restored. The steps to restore from a VPLEX native copy using the Unisphere for VPLEX UI are identical to the copy creation process except the device to be restored becomes the target in the mobility process.

---

**Note:** The assumptions for this process are that the source device will have a virtual volume on top of it and that the host using the restored source virtual will not be impacted if the VPD ID of the virtual volume changes. If there is any uncertainty as to whether or not the host OS will be impacted by VPDID changes, this procedure should not be used until an impact determination made via testing and/or OS documentation review. If there is impact, then remediation steps to account for the VPD ID change should be added to the overall procedure. These steps will vary by OS and host platform.

The steps to perform a restore are:

1. Identify the copy virtual volumes and underlying devices you wish to restore from.
2. Identify the source virtual volumes and underlying devices you wish to restore to.
3. Quiesce any host applications running on the source virtual volume to be restored. It is recommended to unmount the source VPLEX virtual volume to force the host to flush its read cache.
4. Remove the source virtual volume from any storage views it is a member of. Make note of the corresponding lun number for future reference.
5. Remove the virtual volume on top of the source volume to be restored.
6. Create a data mobility job using the copy virtual volume’s underlying VPLEX device as the source device and the source device as the restore target.
7. Allow copy and source to synchronize and enter the commit pending state.
8. **Cancel** the mobility job.
9. Create a new virtual volume on top of the source device that was just restored.
10. Add the virtual volume to the original storage-view ensuring original lun # is used when necessary.
11. Scan for luns (if necessary).
12. Restart host applications.
   i. **Note:** If hosts are sensitive to VPD ID changes then please plan accordingly.
EMC VPLEX Native copies with the VPLEX CLI

Creating VPLEX Native copies

VPLEX Native Cloning can be accomplished from the VPLEX CLI using the data mobility command, ‘dm migration’. The functionality used in this section is intended for mobility jobs and not specifically for disk copy operations. Even so, the results obtained are consistent, predictable, and in keeping with use cases mentioned earlier.

Although our example shows a single copy being created, multiple copies can also be concurrently created. The ‘dm migration’ command is used to make VPLEX Native copies from the CLI. The usage of the ‘dm migration start’ command is as follows:

<table>
<thead>
<tr>
<th>migration start [options] &lt;name&gt; &lt;from&gt; &lt;to&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Line Parameter</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>-h, --help</td>
</tr>
<tr>
<td>-s, --transfer-size= &lt;arg&gt;</td>
</tr>
</tbody>
</table>

positional arguments (* = required):

* -n, --name= <arg> | The name of the new migration. |
* -f, --from= <arg> | The name of source extent or device for the migration. While this element is in use for the migration, it cannot be used for any other purpose, including another migration. |
* -t, --to= <arg> | The name of target extent or device (copy) for the migration. While this element is in use for the migration, it cannot be used for any other purpose, including another migration. |
Once the source virtual volume, source device, and clone device have been identified, the steps to create a VPLEX Native Copy (VNC) using the VPLEX CLI are as follows:

1. Start a new migration

Vplexcli:/> dm migration start --name VNC_Job_1 --from source_device1_1 --to copy_device1_2 --s 32M

This will start a new migration job named ‘VNC_Job_1’ that will create a native copy of source_device1_1 using copy_device1_2 with 32 MB Transfer Speed.

Example CLI output:

Vplexcli:/data-migrations/device-migrations>VNC_Job_1> dm migration start --n VNC_Job_1 --from device_Symm1554_0690_1 --to device_Symm1554_13E_1 --s 32M

Started device migration ‘VNC_Job_1’.

**Note:** The VPLEX mobility facility supports up to 25 concurrent device mobility sessions. More than 25 mobility jobs can be queued, but at any point in time only 25 will be actively synchronizing. In addition, once synchronization has been completed for each pair of devices, they will remain in sync until the next steps are performed by the script or by the administrator.

2. Wait for the migration to finish and enter the commit pending state.

Vplexcli:/data-migrations/device-migrations/VNC_Job_1> ll

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>from-cluster</td>
<td>Boston</td>
</tr>
<tr>
<td>percentage-done</td>
<td>100 (red underline)</td>
</tr>
<tr>
<td>source</td>
<td>device_Symm1554_0690_1</td>
</tr>
<tr>
<td>source-exported</td>
<td>false</td>
</tr>
<tr>
<td>start-time</td>
<td>Tue Nov 23 05:31:53 UTC 2010</td>
</tr>
<tr>
<td>status</td>
<td>complete</td>
</tr>
<tr>
<td>target</td>
<td>device_Symm1554_13E_1</td>
</tr>
<tr>
<td>target-exported</td>
<td>false</td>
</tr>
<tr>
<td>to-cluster</td>
<td>Boston</td>
</tr>
<tr>
<td>transfer-size</td>
<td>32M</td>
</tr>
<tr>
<td>type</td>
<td>Full</td>
</tr>
</tbody>
</table>

Confirm 100% synchronized.

The next phase in the VPLEX native copy process is to split the copy device from the source device. This is accomplished by cancelling the data mobility job while it is in a *Commit Pending* state. Commit Pending is the state when both the source and target devices are in sync. This state is permanent until the mobility job is cancelled. New writes to the source device will be mirrored to the copy device as long as this state is maintained. Once the mobility job is cancelled the virtual volume will revert to its original configuration and the copy device leg will be detached and placed in the device context. There will be no upper level virtual volume corresponding with the copy device, so it will be protected from new writes.
3. **Cancel** the migration. Once the status of the migration is 'complete' it can be **cancelled** in order to create a copy and leave the source device unchanged.

VPLEXcli:/> `dm migration cancel VNC_Job_1 --force`

```
VPLEXcli:/data-migrations/device-migrations> cancel -m VNC_Job_1/ -f
Cancelled 1 data migration(s) out of 1 requested migration(s).
```

**Note:** You can cancel a migration at any time unless you commit it. If it is accidentally committed then in order to return to using the original device you will need to create another mobility activity using the old source device.

4. If necessary, remove the migration record. You may wish to preserve this record to determine the source and target volumes at a later date.

VPLEXcli:/> `dm migration remove VNC_Job_1 --force`

This will remove the VNC_Job_1 migration context from the appropriate `/data-migrations` context.

The copy device is now available in the `/clusters/cluster name/devices/` context in the VPLEX CLI. If you wish to make the VPLEX Native Copy device available to a host you must create a virtual volume on top of it. Follow the normal steps to create and then provision a virtual volume using the copy device you just created.

**Optional Commands:**

To pause and then resume an in-progress or queued migration, use:

VPLEXcli:/> `dm migration pause VNC_Job_1`

VPLEXcli:/> `dm migration resume VNC_Job_1`
Restoring from VPLEX native copies

Restoring from VPLEX native copies can be accomplished from the VPLEX CLI using the `dm migrate` command.

**Note:** The functionality used in this section is intended for mobility jobs and not specifically for cloning operations.

The steps to restore from a VPLEX Native Clone using the VPLEX CLI are identical to the copy creation process except the device to be restored becomes the target in the mobility process. The assumption for this process is that the copy device has a virtual volume on top of it and that the target host will not be impacted if the VPD ID of the virtual volume changes post restore.

The steps to perform a restore are:

1. Identify the copy virtual volume and copy device you wish to restore from.
2. Identify the source virtual volume and device you wish to restore to.
3. Quiesce any host applications running on the source virtual volume to be restored. It is recommended to unmount the source VPLEX virtual volume to force the host to flush its read cache.
4. Remove the source virtual volume from any storage views it is a member of. Make note of the corresponding lun number for future reference.
5. Remove the virtual volume on top of the source volume to be restored.
6. Create a data mobility job with the copy virtual volume and the underlying copy device as the source device and the original production (source) device as the restore target.
7. Allow the copy and the source to synchronize and enter the commit pending state.
8. Cancel the mobility job.
9. Create a new virtual volume on top of source device that was just restored.
10. Add the virtual volume to the original storage-view ensuring original lun # is used when necessary.
11. Scan for luns (if necessary).
12. Restart host applications.
   
   i. Note: If host is sensitive to VPD ID changes then please plan accordingly
Conclusion

Array-based replication continues to provide business value when EMC VPLEX added to your existing storage infrastructure. While modifications to existing procedures are necessary, the changes are relatively simple to implement. In addition to using array-based copy technologies, VPLEX customers can also use native mirroring capabilities within VPLEX to create full disk copies of existing virtual volumes. These copies can be done within and across heterogeneous storage products in a single datacenter or across datacenters. This is a truly differentiating capability that individual frame based copy technologies cannot deliver. VPLEX UI, CLI, and RESTful API provide a variety of methods to continue to deliver high business value for array based copy technologies.

References

The following reference documents are available at Support.EMC.com:

- White Paper: Workload Resiliency with EMC VPLEX
- VPLEX 5.2 Administrators Guide
- VPLEX 5.2 Configuration Guide
- VPLEX Procedure Generator available from http://support.emc.com
- EMC VPLEX HA TechBook
- TechBook: EMC VPLEX Architecture and Deployment: Enabling the Journey to the Private Cloud