EMC® VPLEX™

GeoSynchrony® Release 5.2

Administration Guide

P/N 302-000-033-02

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 01748-9103
1-508-435-1000
www.EMC.com

Published June, 2013

Contents

Preface

As part of an effort to improve and enhance the performance and capabilities of its product line, EMC from time to time releases revisions of its hardware and software. Therefore, some functions described in this document may not be supported by all revisions of the software or hardware currently in use. Your product release notes provide the most up-to-date information on product features.

If a product does not function properly or does not function as described in this document, please contact your EMC representative.

About this guide

This guide is part of the VPLEX documentation set, and is intended for use by customers and service providers to configure and manage a storage environment.

Related documentation

Related documents (available on EMC Support Online) include:

- *EMC VPLEX Release Notes for GeoSynchrony Releases 5.2*
- *EMC VPLEX Product Guide*
- *EMC VPLEX Site Preparation Guide*
- *EMC VPLEX Hardware Installation Guide*
- *EMC VPLEX Configuration Worksheet*
- *EMC VPLEX Configuration Guide*
- *EMC VPLEX Security Configuration Guide*
- *EMC VPLEX CLI Guide*
- *EMC VPLEX Administration Guide*
- *VPLEX Management Console Help*
- *EMC VPLEX Element Manager API Guide*
- *EMC VPLEX Open-Source Licenses*
- *EMC Regulatory Statement for EMC VPLEX*
- Procedures provided through the Generator
- EMC Host Connectivity Guides

Conventions used in this document

EMC uses the following conventions for special notices.

Note: A note presents information that is important, but not hazard-related.

⚠ CAUTION

A caution contains information essential to avoid data loss or damage to the system or equipment.

IMPORTANT

An important notice contains information essential to operation of the software.

Typographical conventions

EMC uses the following type style conventions in this document:

Normal	Used in running (nonprocedural) text for: • Names of interface elements (such as names of windows, dialog boxes, buttons, fields, and menus) • Names of resources, attributes, pools, Boolean expressions, buttons, DQL statements, keywords, clauses, environment variables, functions, utilities • URLs, pathnames, filenames, directory names, computer names, filenames, links, groups, service keys, file systems, notifications
Bold	Used in running (nonprocedural) text for: • Names of commands, daemons, options, programs, processes, services, applications, utilities, kernels, notifications, system call, man pages Used in procedures for: • Names of interface elements (such as names of windows, dialog boxes, buttons, fields, and menus) • What user specifically selects, clicks, presses, or types
Italic	Used in all text (including procedures) for: • Full titles of publications referenced in text • Emphasis (for example a new term) • Variables
`Courier`	Used for: • System output, such as an error message or script • URLs, complete paths, filenames, prompts, and syntax when shown outside of running text
`Courier bold`	Used for: • Specific user input (such as commands)
`Courier italic`	Used in procedures for: • Variables on command line • User input variables
[ ]	Square brackets enclose optional values
|	Vertical bar indicates alternate selections - the bar means “or”
{ }	Braces indicate content that you must specify (that is, x or y or z)
...	Ellipses indicate nonessential information omitted from the example

Where to get help

EMC support and product information can be obtained as follows.

Product information — For documentation, release notes, software updates, or for information about EMC products, licensing, and service, go to the EMC Support website (registration required) at:

`http://support.EMC.com`

Technical support — For technical support, go to the EMC Support site. To open a service request, you must have a valid support agreement. Please contact your EMC sales representative for details about obtaining a valid support agreement or to answer any questions about your account.

Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Please send your opinion of this document to:

techpubcomments@EMC.com

1 Using the VPLEX CLI

This chapter describes how to use the VPLEX™ command line interface (CLI).

Log in to/log out from the CLI

Log in

1. Log in to the service account at the cluster's management server's public IP address using an SSH client (PuTTY or OpenSSH). Configure the SSH client as follows:

   - Port 22
   - SSH protocol version is set to 2
   - Scrollback lines set to 20000

   The log in prompt appears:

   ```
   login as:
   ```

2. Type **service** and press **ENTER**. A password prompt appears:

   ```
   Using keyboard-interactive authentication.
   Password:
   ```

3. Type the service password and press **ENTER**. The default password is **Mi@Dim7T**. A server prompt appears:

   ```
   service@ManagementServer:~>
   ```

4. Type the **vplexcli** command to connect to the VPLEX command line interface:

   ```
   service@ManagementServer:~> vplexcli
   ```

   Several messages are displayed, and a username prompt appears:

   ```
   Trying 127.0.0.1...
   Connected to localhost.
   Escape character is '^]'.

   Enter User Name:
   ```

5. Type **service** and press **ENTER**.

   ```
   Enter User Name: service
   ```

   A password prompt appears:

   ```
   Password:
   ```

6. Type the service password and press **ENTER**. The default password is **Mi@Dim7T**.

The VPLEX command line interface prompt appears:

```
creating logfile:/var/log/VPlex/cli/session.
log_service_localhost_T28921_20101020175912

VPlexcli:/>
```

Log out

Use the **exit** command to exit the VPLEX command line interface from any context.

For example:

```
VPlexcli:/clusters> exit
Connection closed by foreign host.
```

Password Policies

The VPLEX management server uses a Pluggable Authentication Module (PAM) infrastructure to enforce minimum password quality. For more information about technology used for password protection, refer to the *VPLEX Security Configuration Guide*.

Refer to "Log in to/log out from the CLI" on page 12 for information on the commands used to set password policies and the values allowed.

Note the following:

- Password policies do not apply to users configured using the LDAP server.
- Password policies do not apply to the **service** account.
- The Password inactive days policy does not apply to the **admin** account to protect the admin user from account lockouts.
- During the management server software upgrade, an existing user's password is not changed—only the user's password age information changes.
- You must be an **admin** user to configure a password policy.

Table 1 lists and describes the password policies and the default values.

Table 1 Default password policies

Policy name	Description	Default value
Minimum password length	The minimum number of characters used when creating or changing a password. The minimum number of characters includes numbers, upper-case and lower-case letters, and special characters.	14
Minimum password age	The minimum number of days a password can be changed after the last password change.	1
Maximum password age	The maximum number of days that a password can be used since the last password change. After the maximum number of days, the account is locked and the user must contact the admin user to reset the password.	90
Password expiry warning	The number of days before the password expires. A warning message indicating that the password must be changed is displayed.	15
Password inactive days	The number of days after a password has expired before the account is locked.	1

The password policy for existing admin and customer-created user accounts is updated automatically as part of the upgrade. See the *VPLEX Security Configuration Guide* for information about service user account passwords.

Valid Password Characters

The following characters are allowed in a VPlexcli password:

- A-Z
- a - z
- 0 - 9
- . ? / * @ ^ % # + = - _ ~ : space

Note: A space is allowed only between the characters in a password, not in the beginning or the end of the password

CLI context tree

The CLI is divided into command contexts. Some commands are accessible from all contexts, and are referred to as 'global commands'.

The remaining commands are arranged in a hierarchical context tree. These commands can only be executed from the appropriate location in the context tree.

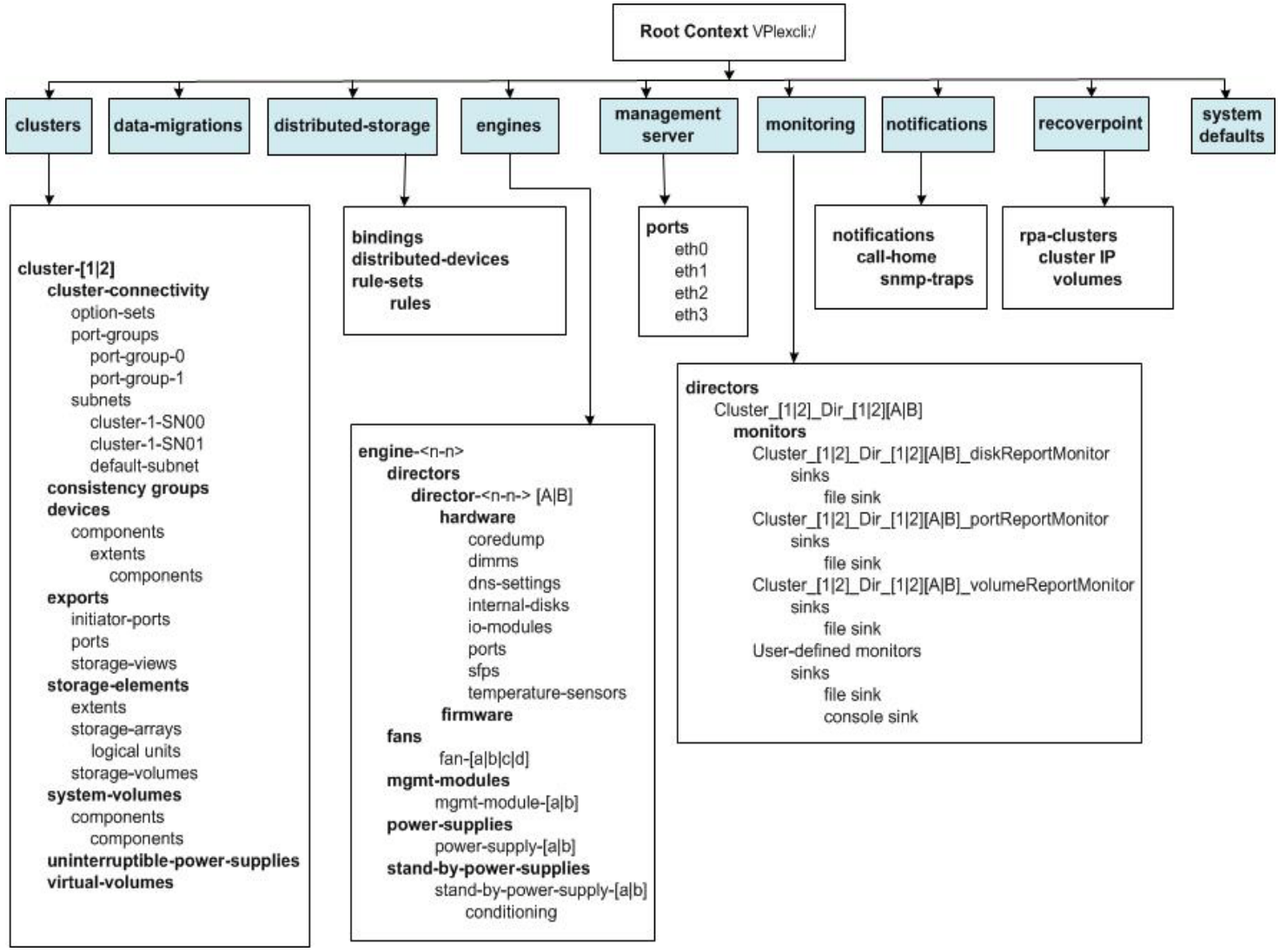
Understanding the command context tree is critical to using the VPLEX command line interface effectively.

The root context contains ten sub-contexts:

- **clusters/** - Create and manage links between clusters, devices, extents, system volumes and virtual volumes. Register initiator ports, export target ports, and storage views.
- **data-migrations**/ - Create, verify, start, pause, cancel, and resume data migrations of extents or devices.
- **distributed-storage/** - Create and manage distributed devices and rule sets.
- **engines/** - Configure and manage directors, fans, management modules, and power.
- **management-server/** - Manage the Ethernet ports.
- **monitoring/** - Create and manage performance monitors.
- **notifications/** - Create and manage call-home events.
- **recoverpoint/** - Manage RecoverPoint options.
- **security/** - Configure and view authentication password-policy settings. Create, delete, import and export security certificates. Set and remove login banners. The authentication sub context was added to the security context.
- **system-defaults/** - Display systems default settings.

Except for system-defaults/, each of the sub-contexts contains one or more sub-contexts to configure, manage, and display sub-components.

Command contexts have commands that can be executed only from that context. The command contexts are arranged in a hierarchical context tree. The topmost context is the root context, or "/".


Root Context VPlexcli:/
clusters
data-migrations
distributed-storage
engines
management server
monitoring
notifications
recoverpoint
system defaults
cluster-[1|2]
cluster-connectivity
option-sets
port-groups
port-group-0
port-group-1
subnets
cluster-1-SN00
cluster-1-SN01
default-subnet
consistency groups
devices
components
extents
components
exports
initiator-ports
ports
storage-views
storage-elements
extents
storage-arrays
logical units
storage-volumes
system-volumes
components
components
uninterruptible-power-supplies
virtual-volumes
bindings
distributed-devices
rule-sets
rules
ports
eth0
eth1
eth2
eth3
notifications
call-home
snmp-traps
rpa-clusters
cluster IP
volumes
engine-<n-n>
directors
director-<n-n-> [A|B]
hardware
coredump
dimms
dns-settings
internal-disks
io-modules
ports
sfps
temperature-sensors
firmware
fans
fan-[a|b|c|d]
mgmt-modules
mgmt-module-[a|b]
power-supplies
power-supply-[a|b]
stand-by-power-supplies
stand-by-power-supply-[a|b]
conditioning
directors
Cluster_[1|2]_Dir_[1|2][A|B]
monitors
Cluster_[1|2]_Dir_[1|2][A|B]_diskReportMonitor
sinks
file sink
Cluster_[1|2]_Dir_[1|2][A|B]_portReportMonitor
sinks
file sink
Cluster_[1|2]_Dir_[1|2][A|B]_volumeReportMonitor
sinks
file sink
User-defined monitors
sinks
file sink
console sink

Navigate the CLI context tree

Use the **cd** command to navigate between command contexts.

The current context is always displayed at the VPLEX command line interface prompt:

```
VPlexcli:/> cd /clusters/cluster-1/devices/

VPlexcli:/clusters/cluster-1/devices>
```

For example, to navigate from the root (/) context to the monitoring context for a specified director:

```
VPlexcli:/> cd /monitoring
VPlexcli:/monitoring> cd /directors/
VPlexcli:/monitoring/directors> cd /director-1-1-B
VPlexcli:/monitoring/directors/director-1-1-B> cd /monitors
VPlexcli:/monitoring/directors/director-1-1-B/monitors> cd
/director-1-1-B
VPlexcli:/monitoring/directors/director-1-1-B/monitors/director-1-1-B
> ls
Attributes:
Name              Value
---------------   --------------------------------
average-period    1s
bucket-count      64
.
.
.
```

Alternatively, type all the context identifiers in a single command. For example, the above navigation can be typed as:

```
VPlexcli:/> cd
/monitoring/directors/director-1-1-B/monitors/director-1-1-B_director

VPlexcli:/monitoring/directors/director-1-1-B/monitors/director-1-1-B
>
```

Use the **cd** command with no arguments or followed by three periods (**cd...**) to return to the root context:

```
VPlexcli:/engines/engine-1-1/fans> cd

VPlexcli:/>
```

Use the **cd** command followed by two periods (**cd..**) to return to the context immediately above the current context:

```
VPlexcli:/monitoring/directors/director-1-1-B> cd ..

VPlexcli:/monitoring/directors>
```

To navigate directly to a context from any other context use the **cd** command and specify the absolute context path. In the following example, the **cd** command changes the context from the data migrations/ extent-migrations context to the engines/engine-1/fans context:

```
VPlexcli:/data-migrations/extent-migrations> cd
/engines/engine-1-1/fans/

VPlexcli:/engines/engine-1-1/fans>
```

pushd and popd commands

- Use the **pushd** ***directory*** command to save the current directory, and jump to the specified directory.

  Once a directory is added to the pushd stack, use the **pushd** command with no argument to switch back to the previous directory.

  In the following example, **pushd** toggles between the engines and monitoring parent contexts:

```
VPlexcli:/engines/engine-1-1/directors/director-1-1-A> pushd
/monitoring/directors/director-1-1-A
[/monitoring/directors/director-1-1-A,
/engines/engine-1-1/directors/director-1-1-A,
/monitoring/directors/director-1-1-A]

VPlexcli:/monitoring/directors/director-1-1-A> pushd
[/engines/engine-1-1/directors/director-1-1-A,
/monitoring/directors/director-1-1-A,
/monitoring/directors/director-1-1-A]

VPlexcli:/engines/engine-1-1/directors/director-1-1-A> pushd
[/monitoring/directors/director-1-1-A,
/engines/engine-1-1/directors/director-1-1-A,
/monitoring/directors/director-1-1-A]

VPlexcli:/monitoring/directors/director-1-1-A>
```

- Use the **dirs** command to display to the current context stack:

```
VPlexcli:/clusters/cluster-1> dirs
[/clusters/cluster-1, /, /,
/engines/engine-1-1/directors/director-1-1-A/hardware/ports/A5-GE0
1, /]
```

- Use the **popd** command to remove the last directory saved by the **pushd** command and jump to the new top directory.

  In the following example, the **dirs** command displays the context stack saved by the **pushd** command, and the **popd** command removes the top directory, and jumps to the new top directory:

```
VPlexcli:/engines/engine-1-1/directors/director-1-1-A> dirs
[/engines/engine-1-1/directors/director-1-1-A,
/monitoring/directors/director-1-1-A]

VPlexcli:/engines/engine-1-1/directors/director-1-1-A> popd
[/engines/engine-1-1/directors/director-1-1-A]

VPlexcli:/monitoring/directors/director-1-1-A>
```

Where am I in the context tree?

The VPLEX CLI includes several features to help locate your current position in the context tree and determine what contexts and/or commands are accessible.

Note: The context tree displays only those objects associated with directors to which the management system is connected.

- The command prompt displays the current context:

```
VPlexcli:/> cd /monitoring/directors/director-1-1-B/monitors/

VPlexcli:/monitoring/directors/director-1-1-B/monitors>
```

- The **ls** command displays the sub-contexts immediately accessible from the current context:

```
VPlexcli:/> ls
clusters       data-migrations    distributed-storage
engines        management-server  monitoring
notifications  system-defaults
```

- The **ls -l** command displays more information about the current sub-contexts:

```
VPlexcli:/data-migrations> ls -l
Name               Description
-----------------  -------------------------------------
device-migrations  Contains all the device migrations in the system.
extent-migrations  Contains all the extent migrations in the system.
```

- For contexts where the next lowest level is a list of individual objects, the **ls** command displays a list of the objects:

```
VPlexcli:/clusters/cluster-1/exports/ports> ls
P000000003B2017DF-A0-FC00  P000000003B2017DF-A0-FC01
P000000003B2017DF-A0-FC02  P000000003B2017DF-A0-FC03
P000000003B3017DF-B0-FC00  P000000003B3017DF-B0-FC01
P000000003B3017DF-B0-FC02  P000000003B3017DF-B0-FC03
```

- The **cd** command followed by a <Tab> displays the same information as **ls** at the context level.

  For example, type **cd** and press <Tab> in the data-migrations context to display available options:

```
VPlexcli:/data-migrations> cd <Tab>

device-migrations/   extent-migrations/
```

- The **tree** command displays the immediate sub-contexts in the tree using the current context as the root:

```
VPlexcli:/ cd /clusters/cluster-1/devices/Symm_rC_3

VPlexcli:/clusters/cluster-1/devices/Symm_rC_3> tree
/clusters/cluster-1/devices/Symm_rC_3:
   components
     Symm_rC_3_extent_0
     Symm_rC_3_extent_1
```

- The **tree -e** command displays immediate sub-contexts in the tree and any sub-contexts under them:

```
VPlexcli:/clusters/cluster-1/devices/Symm_rC_3> tree -e
/clusters/cluster-1/devices/Symm_rC_3:
   components
     Symm_rC_3_extent_0
       components
         Symm0487_44C
           components
     Symm_rC_3_extent_1
       components
         Symm0487_44B
           components
```

Note: For contexts where the next level down the tree is a list of objects, the **tree** command displays the list. This output can be very long. For example:

```
VPlexcli:/clusters/cluster-1> tree
/clusters/cluster-1:
   cluster-connectivity
```

```
      cluster-links
        to-cluster-2
      proxy-servers
      static-routes
    devices
      base0
        components
          extent_CX4_lun0_1
            components
              CX4_lun0
                components
.
.
.
exports
      initiator-ports
        LicoJ006_hba0
        LicoJ006_hba1
.
.
.
ports
        P000000003CA00147-A0-FC00
        P000000003CA00147-A0-FC01
.
.
.
storage-views
        LicoJ009
        LicoJ013
    storage-elements
      extents
        extent_CX4_Logging_1
.
.
.
```

Using CLI commands

This section describes the following topics:

Display currently available commands

The commands that make up the CLI fall into two groups:

- Global commands that can be used in any context. For example: **cd**, **date**, **ls**, **exit**, **user**, and **security**.
- Context-specific commands that can be used only in specific contexts. For example, to use the **copy** command, the context must be /distributed-storage/rule-sets.

Use the **help** command to display a list of all commands (including the global commands) available from the current context.

Use the **help -G** command to display a list of available commands in the current context *excluding* the global commands:

```
VPlexcli:/notifications> help -G
Commands specific to this context and below:
call-home  snmp-trap
```

Some contexts "inherit" commands from their parent context. These commands can be used in both the current context and the context immediately above in the tree:

```
VPlexcli:/distributed-storage/bindings> help -G
Commands inherited from parent contexts:
dd  rule  rule-set  summary
```

Some commands are loosely grouped by function. For example, the commands to create and manage performance monitors start with the word "monitor".

Use the <Tab> key display the commands within a command group. For example, to display the commands that start with the word "monitor", type "monitor" followed by the <Tab> key:

```
VPlexcli:/> monitor <Tab>

add-console-sink   add-file-sink      collect          create
destroy            remove-sink
stat-list
```

Page output

For large configurations, output from some commands can reach hundreds of lines.

Paging displays long output generated by the **ll** and **ls** commands one page at a time:

To enable paging, add **-p** at the end of any command:

```
VPlexcli:/clusters/cluster-1/storage-elements>  ls storage-volumes -p
```

One page of output is displayed. The following message is at the bottom of the first page:

```
-- more --(TOP )- [h]elp
```

Press the spacebar to display the next page.

The message now indicates what percentage of the output has been displayed:

```
-- more --( 24%)- [h]elp
```

h - Displays instructions on how to move and search the output.

q - Exits paging mode.

Tab completion

Use the Tab key to:

- Complete a command
- Display valid contexts and commands
- Display command arguments

Complete a command

Use the Tab key to automatically complete a path or command until the path/command is no longer unique.

For example, to navigate to the UPS context on a single cluster (named cluster-1), type:

```
cd /clusters/cluster-1/uninterruptible-power-supplies/
```

To type the same command using tab completion:

1. Type **cd c** Tab

   Since 'clusters' is the only context starting with 'c' at the root level, the CLI auto-completes the selection:

   ```
   cd /clusters/
   ```

2. There is only one cluster (it is unique). Press Tab to automatically specify the cluster:

   ```
   cd /clusters/cluster-1/
   ```

3. Type a **u** to select the uninterruptible-power-supplies context and press Tab.

   The u is unique at the current context, and the CLI auto-completes the selection:

   ```
   cd /clusters/cluster-1/uninterruptible-power-supplies/
   ```

Display valid contexts and commands

Press Tab after typing a partial context path to display a list of valid commands and/or contexts for the current context:

```
VPlexcli:/> cd /clusters/cluster-1/ <Tab>

cluster-connectivity/             devices/
exports/                          storage-elements/
system-volumes/                   uninterruptible-power-supplies/
virtual-volumes/

VPlexcli:/> cd /clusters/cluster-1/
```

Display command arguments

Press Tab after typing a command name to display the command's arguments. For example:

```
VPlexcli:/> monitor <Tab>

add-console-sink     add-file-sink    collect
create               destroy          remove-sink stat-list
```

Wildcards

The VPLEX command line interface includes 3 wildcards:

- ***** - matches any number of characters.
- **?** - matches any single character.
- **[a|b|c]** - matches any of the single characters a or b or c.

* wildcard

Use the * wildcard to apply a single command to multiple objects of the same type (directors or ports). For example, to display the status of ports on each director in a cluster, without using wildcards:

```
ll engines/engine-1-1/directors/director-1-1-A/hardware/ports
ll engines/engine-1-1/directors/director-1-1-B/hardware/ports
ll engines/engine-1-2/directors/director-1-2-A/hardware/ports
ll engines/engine-1-2/directors/director-1-2-B/hardware/ports
.
.
.
```

Alternatively:

- Use one * wildcard to specify all engines, and
- Use a second * wildcard specify all directors:

```
ll engines/engine-1-*/directors/*/hardware/ports
```

** wildcard

Use the ** wildcard to match all contexts and entities between two specified objects. For example, to display all director ports associated with all engines without using wildcards:

```
ll /engines/engine-1-1/directors/director-1-1-A/hardware/ports
ll /engines/engine-1-1/directors/director-1-1-B/hardware/ports
.
.
.
```

Alternatively, use a ** wildcard to specify all contexts and entities between /engines and ports:

```
ll /engines/**/ports
```

? wildcard

Use the **?** wildcard to match a single character (number or letter).

```
ls /storage-elements/extents/0x1?[8|9]
```

Returns information on multiple extents.

(a|b|c) wildcard

Use the [a|b|c] wildcard to match one or more characters in the brackets.

```
ll engines/engine-1-1/directors/director-1-1-A/hardware/ports/A[0-1]
```

displays only ports with names starting with an A, and a second character of 0 or 1.

Names

Major components of the VPLEX are named as follows:

- Clusters - VPLEX Local™ configurations have a single cluster, with a cluster ID of cluster 1. VPLEX Metro™ and VPLEX Geo™ configurations have two clusters with cluster IDs of 1 and 2.

  ```
  VPlexcli:/clusters/cluster-1/
  ```

- Engines are named <*engine-n-n*> where the first value is the cluster ID (1 or 2) and the second value is the engine ID (1-4).

  ```
  VPlexcli:/engines/engine-1-2/
  ```

- Directors are named <*director-n-n-n*> where the first value is the cluster ID (1 or 2), the second value is the engine ID (1-4), and the third is A or B.

  ```
  VPlexcli:/engines/engine-1-1/directors/director-1-1-A
  ```

For objects that can have user-defined names, those names must comply with the following rules:

- Can contain uppercase and lowercase letters, numbers, and underscores
- No spaces
- Cannot start with a number
- No more than 63 characters

Command globbing

Command globbing combines wildcards and context identifiers in a single command. Globbing can address multiple entities using a single command.

Example 1

In the following example, a single command enables ports in all engines and all directors (A and B) whose name include 0-FC and 1-FC:

```
set /engines/*/directors/*/hardware/ports/*[0-1]-FC*:: enabled true
```

- First * wildcard — All engines in the cluster.
- Second * wildcard — All directors in the cluster.
- Third * wildcard — All A-side ports and all B-side ports.
- The [0-1] limits the selections to all port numbers that start with A0, A1, B0, or B1.
- Fourth * wildcard — All ports whose numbers start with A0-FC, A1-FC, B0-FC, or B1-FC.

Example 2

To display the status of all the director ports on a large configuration using no wildcards, type:

```
ll /engines/engine-1-<Enclosure_ID>/directors/<director_
name>/hardware/ports
```

for each engine and director.

Using the * wildcard reduces this task to a single command:

```
ll /engines/engine-1-*/directors/*/hardware/ports
```

Using the ** wildcard simplifies the command even more:

```
ll /**/ports
```

Positional command arguments

Most commands require arguments.

Some command arguments are *positional*. That is, the argument can be typed without an identifier IF it is entered in the position specified by the command syntax.

For example, the **alias** command has two arguments in the following order (syntax):

```
alias
   [-n|--name] <alias name>
   [-t|to] <"string of commands in quotes">
```

Type the command with the arguments *with identifiers* in any order (not as specified by the syntax):

```
VPlexcli:/> alias --to "cd clusters" --name cdc
```

or,

Type the command with the arguments *without identifiers* in the order specified by the command syntax:

```
VPlexcli:/> alias cdc "cd clusters"
```

--verbose argument

The **--verbose** argument displays additional information for some commands. For example, without **--verbose** argument:

```
VPlexcli:/> connectivity validate-be

Summary

Cluster cluster-1
    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
```

With **--verbose** argument:

```
VPlexcli:/> connectivity validate-be --verbose
Storage volumes that are dead or unreachable:

Cluster    Dead or Unreachable Storage Volumes
---------  ------------------------------------
cluster-2  VPD83T3:60004530000000080007f16e9512a2b1
cluster-1  VPD83T3:60004530000000010007f16e9512a2a5
           VPD83T3:60004530000000010007f16e9512a2a7
           VPD83T3:60004530000000010007f16e9512a2a9

Summary
Cluster cluster-2
    This cluster has 1 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 3 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
```

Search command history

- To display the last commands typed, press the up arrow key.

- To search for a command typed in the current CLI session, press **Ctrl-r**.

  The reverse search prompt is displayed:

  ```
  (reverse-i-search)'':
  ```

  Type the first letter of the command to search for. After the first letter is typed, the search tool displays a list of possible matches.

View command history

Use the "up arrow" key to display the last command typed.

Use the "up arrow" key, multiple times to display recent command history.

Use the **history** command to display a complete list of commands executed in the current session:

```
VPlexcli:/engines/engine-0-0/directors> history
VPlexcli:/> history
0 cd engines/engine-0-0/directors
1 extent unclaim *
2 ls
3 ls -l
4 extent claim *
5 ls
6 ls -l
7 ls -la
```

Use the **history** *nn* command to display the last *nn* entries in the list:

```
VPlexcli:/clusters/cluster-1> history 22
478 ls storage-volumes -p
479 cd clusters/cluster-1/
480 ls storage-volumes
481 cd storage-elements/
482  ls storage-volumes -p
```

Get help

- Use the **help** or **?** command with no arguments to display all the commands available, the current context, including global commands.
- Use the **help** or **?** command with **-G** argument to display all the commands available, the current context, excluding global commands:

  ```
  VPlexcli:/clusters> help -G
  Commands specific to this context and below:

  add  cacheflush  configdump  expel  forget  shutdown  summary  unexpel
  ```

- Use the **help** *command* or *command* **-help** to display help for the specified command.

2

CLI Workspace and User Accounts

This chapter describes how to use the VPLEX command line interface (CLI) to configure the CLI workspace and to manage user accounts.

Configure the CLI workspace

The workspace is the appearance and behavior of a CLI session. Use the procedures described in this section to control the output of commands, the level of logging messages sent to the console, and to search the command history of the current CLI session.

Set/remove the login banner

You can customize the login banner for the VPLEX management servers.

Use the **security set-login-banner** command to apply the contents of a text file as the login banner.

The change takes effect at the next login to the management server.

The formatting of the text in the specified text file is replicated in the banner.

There is no limit to the number of characters or lines in the specified text file.

The text file must be saved in a directory on the management server.

In the following example, a text file "login-banner.txt" containing the following lines is specified as the login banner:

VPLEX cluster-1/Hopkinton

Test lab 3, Room 6, Rack 47

Metro with RecoverPoint CDP

```
VPlexcli:/> security set-login-banner -b
   /home/service/login-banner.txt
The text provided in the specified file will be set as the Login banner
   for this management server.
Any previously applied banner will be overwritten.
Do you want to proceed ? (Yes/No) Yes
```

At next login to the management server, the new login banner is displayed:

```
login as: service
VPLEX cluster-1/Hopkinton
Test lab 3, Room 6, Rack 47
Metro with RecoverPoint CDP
Password:
```

Use the **security remove-login-banner** command to remove the login banner.

```
VPlexcli:/> security remove-login-banner

The login banner of this management server will be removed.
Do you want to proceed ? (Yes/No) yes
```

Set the threshold for console logging

The console logger displays messages received from directors on the console.

By default, the console displays only emergency (level 0) messages.

Messages are categorized into 8 severities (0-7), with 0 being the most severe:

7 - **debug** (debug-level messages)

6 - **info** (informational messages)

5 - **notice** (normal but significant messages)

4 - **warning** (warning messages)

3 - **err** (error messages)

2 - **crit** (critical messages)

1 - **alert** (messages that must be handled immediately)

0 - **emerg** (messages notifying the system as unusable)

To enable messages with lower severity to appear on the console, change the threshold of the logging filter for the console.

1. Use the **log filter list** command to display existing log filters.

   ```
   VPlexcli:/> log filter list
   1. [Threshold='>0'] Destination='null' Consume='true'
   2. [Message matches 'Family and Fru Id Mismatch Retrieved']
   Destination='null' Consume='true'
   ```

2. Determine the ID of the filter controlling the display of messages to the console. The console filter has the following attributes:

   Threhold='>=0'
   Destination= 'null'
   Consume='true'

3. Use the **log filter destroy** command to delete the existing console logging filter.

   ```
   VPlexcli:> log filter destroy 1
   ```

4. Use the **log filter create** command to create a new filter for the console with the required threshold:

   ```
   VPlexcli:> log filter create --threshold <n> --component "logserver"
   ```

   where *n* is 0-7.

Note: The threshold value filters all messages with greater or equal severity.
To see critical (2) and above (0 and 1), set the threshold at 3.
To see error (3) and above (0, 1, and 2) set the threshold at 4.

Set window width to 100

Output from many commands is more than 80 columns wide. EMC recommends that the command window in which VPlexcli is running be expanded to at least 100 columns in width.

Managing User Accounts

Add a user account

1. Log in to the admin account of the VPLEX CLI.
2. Type the **user add** <*username*> command.

   Usernames may be up to 1-32 characters, and contain numbers, letters and special characters, and no spaces. For example:

   ```
   VPlexcli:/> user add TestUser
   ```

   A prompt for the Administrator password appears:

   ```
   admin password:
   ```

3. Type the password for the Administrator username.

   A prompt for the new password for the username being added appears:

   ```
   New password:
   ```

4. Type the password for the new username. Passwords must be at least eight characters, and may contain numbers, letters, and special characters. No spaces. No dictionary words.

   A prompt to confirm the new password appears:

   ```
   Confirm password:
   ```

5. Retype the password.
6. Repeat steps 2-5 for each new user account.
7. Use the **user list** command to verify the new account(s):

   ```
   VPlexcli:/> user list
   Username
   --------
   admin
   monitoruser
   service
   TestUser
   ```

First login to a new user account

In order to login to the CLI, newly created user accounts must change their password on the management server.

The new password must be at least 14 characters long.

Example of a first login after an account is created:

```
login as: newuser
Using keyboard-interactive authentication.
Password: old-password
Using keyboard-interactive authentication.
Password change requested. Choose a new password.
Old Password: old-password
Using keyboard-interactive authentication.
New Password: my-new-password
Using keyboard-interactive authentication.
Reenter New Password: my-new-password
```

```
Password changed.

localuser@ManagementServer:~>
```

After this initial login is completed, subsequent logins behave as described in "Log in" on page 12

Delete a user account.

1. Log in to the VPLEX CLI as an Administrator user.
2. Use the **user list** command to display all accounts:

```
VPlexcli:/> user list
Username
--------
admin
monitoruser
service
TestUser
```

3. Type the **user remove** <*username*> command:

```
VPlexcli:/> user remeove TestUser
```

A prompt for the Administrator password appears:

```
admin password:
```

4. Type the password for the Administrator username.

The specified user username is removed.

5. Use the **user list** command to verify the deletion.

Change user account password

Allows all users to change the password only for their own username.

1. Login to the CLI using the account whose password needs to be changed.
2. Use the **user passwd** <*username*> command to change the password of an existing user.

For example:

```
VPlexcli:/> user passwd monitoruser
```

A prompt for the current password appears:

```
old password:
```

A prompt for the new password appears:

```
New password:
```

3. Type the new password. Passwords must be at least 14 characters long, and must not be dictionary words.

A prompt to confirm the new password appears:

```
Confirm password:
```

4. Retype the password.

Reset a user account

1. Login as an Administrator user.
2. Type the **user reset --username** <*username*> command:

   ```
   VPlexcli:/> user reset --username TestUser
   ```

   A prompt for the Administrator password appears:

   ```
   admin password:
   ```

3. Type the password for the Administrator username.

   A prompt for new password for the username being reset appears:

   ```
   New password:
   ```

4. Type a new password for the username.

   A prompt to confirm the new password appears:

   ```
   Confirm password:
   ```

5. Re-type the password.

3

Meta-volumes

This chapter describes the procedures to manage metadata and meta-volumes using the VPLEX CLI:

About meta-volumes

VPLEX metadata includes virtual-to-physical mappings, data about devices, virtual volumes, and system configuration settings.

Metadata is stored in cache and backed up on specially designated external volumes called meta-volumes.

Meta-volumes are created during system setup.

When a cluster is initially configured, the meta-volume must be the first storage presented to VPLEX. This prevents the meta-volume from being accidentally overwritten.

After the meta-volume is configured, updates to the metadata are written to both the cache and the meta-volume when the VPLEX configuration is modified.

Backup meta-volumes are point-in-time snapshots of the current metadata, and provide extra protection before major configuration changes, refreshes, or migrations.

Metadata is read from the meta-volume only during the boot of each director.

Meta-volume backups are created:

- Before migrating to a new array
- Before a major update.

Meta-volumes differ from standard storage volumes in that:

- A meta-volume is created without first being claimed
- Meta-volumes are created directly on storage volumes, not extents.

Refer to the *VPLEX Configuration Guide* for more details about the criteria to select storage used for meta-volumes.

CAUTION

If the meta-volume is configured on a CLARiiON array, it must not be placed on the vault drives of the CLARiiON.

Meta-volume performance and availability requirements

Performance is not critical for meta-volumes. The minimum performance allowed is 40 MB/sec and 100 4 K IOP/second.

The physical spindles for meta-volumes should be isolated from application workloads.

EMC recommends the following for meta-volumes:

- Read caching should be enabled
- A hot spare meta-volume be pre-configured in case of a catastrophic failure of the active meta-volume.

Availability is *critical* for meta-volumes. The meta-volume is essential for system recovery. The best practice is to mirror the meta-volume across two or more back-end arrays to eliminate the possibility of data loss. Choose the arrays used to mirror the meta-volume such that they are not required to migrate at the same time.

DANGER

Do not create a new meta-volume using volumes from a single storage array. Single array meta-volumes are not a high availability configuration and are a single point of failure.

If VPLEX temporarily loses access to all meta-volumes, the current metatdata in cache is automatically written to the meta-volumes when access is restored.

If VPLEX permanently loses access to both meta-volumes, it will continue to operate based on the metadata in memory. Configuration changes are suspended until a new meta-volume is created.

Note: If the VPLEX loses access to all meta-volumes, and all directors either fail or are re-booted, changes made to the meta-data (the VPLEX configuration) after access was lost cannot be recovered.

Create a meta-volume

To create a meta-volume:

1. Use the **configuration show-meta-volume- candidates** command to display possible candidates:

Note: The following example output is truncated.

```
VPlexcli:/> configuration show-meta-volume-candidates
Name                                      Capacity...Array Name
----------------------------------------  --------  ------------------------
VPD83T3:6006048000019010054753303036453 9  187G .....EMC-SYMMETRIX-190100547
VPD83T3:6000097000019260170753303133313 2  98.5G.....EMC-SYMMETRIX-192601707
VPD83T3:6000097000019260170753303133313 3  98.5G.....EMC-SYMMETRIX-192601707
VPD83T3:6000097000019260170753303133313 4  98.5G.....EMC-SYMMETRIX-192601707
VPD83T3:6000097000019260170753303133313 5  98.5G.....EMC-SYMMETRIX-192601707
VPD83T3:6000097000019260170753303133313 6  98.5G.....EMC-SYMMETRIX-192601707
VPD83T3:6000097000019260170753303133313 7  98.5G.....EMC-SYMMETRIX-192601707
VPD83T3:6000097000019260170753303133313 8  98.5G.....EMC-SYMMETRIX-192601707
VPD83T3:6006016049e02100442c66c8890ee011  80G ......EMC-CLARiiON-FNM00083800068
.
.
.
The log summary for configuration automation has been captured in
/var/log/VPlex/cli/VPlexconfig.log


The task summary and the commands executed for each automation task has been captured in
/var/log/VPlex/cli/VPlexcommands.txt


The output for configuration automation has been captured in
/var/log/VPlex/cli/capture/VPlexconfiguration-session.txt
```

2. Use the **meta-volume create** command to create a new meta-volume. The syntax for the command is:

```
meta-volume create --name meta-volume_name --storage-volumes
storage-volume_1,storage-volume_2,storage-volume_3
```

IMPORTANT

Specify two or more storage volumes. Storage volumes must be:
- unclaimed
- on different arrays

```
VPlexcli:meta-volume create --name ICO_META_1_1_Metadata -storage-volumes
VPD83T3:6000097000019260170753303133313 6, VPD83T3:6006048000019030048753303034344 5
```

3. Navigate to the system volume context.
4. Use the **ll** command to display the new meta-volume's status:

```
VPlexcli:/clusters/cluster-1/system-volumes> ll ICO_META_1_1_Metadata

/clusters/cluster-1/system-volumes/ICO_META_1_1_Metadata:

Attributes:
Name                  Value
--------------------  --------------------
active                true
```

```
application-consistent  false
block-count             24511424
block-size              4K
capacity                79.6G
free-slots              31968
geometry                raid-1
.
.
.
```

Verify that the **active** attribute shows a value of **true**.

5. Use the **cluster status** command to display the cluster status:

```
VPlexcli:/clusters/cluster-1/system-volumes> cluster status
Cluster cluster-1
        operational-status:        ok
        transitioning-indications:
        transitioning-progress:
        health-state:              ok
        health-indications:
```

Wait for the **operational status** field to transition to **ok** (while the meta-volume synchronizes with the mirror) before proceeding with other tasks.

Back up the meta-volume

Backup creates a point-in-time copy of the current in-memory metadata without activating it. The new meta-volume is named:

*current-metadata-name*backup_yyyyMMMdd_HHmms

Create a backup meta-volume:

- As part of an overall system health check before a major migration or update
- If VPLEX permanently loses access to both meta-volumes

Back up the meta-volume: VPLEX Local

Before you begin

Identify two or more storage volumes to which to backup the metadata. Target storage volumes must be:

- Unclaimed
- 78 GB or larger

To back up the metadata for a single cluster configuration:

1. Use the **ll** command in device migration and extent migration contexts to verify that there are no active migrations:

   ```
   VPlexcli:/data-migrations/device-migrations> ll

   VPlexcli:/data-migrations/extent-migrations> ll
   ```

   If any migrations are in-progress or queued:

   - Allow the migrations to complete, and commit them (see "Commit a completed migration"); or
   - Cancel the migrations (see "Cancel a migration (optional)") and remove them (see "Clean a migration").

2. Use the following commands to verify the overall health of VPLEX:
   - **validate-system-configuration** - Performs a basic system configuration check
   - **cluster status** - Displays a cluster's operational-status and health-state
   - **export storage-view summary** -Lists each view, and the number of volumes and initiators that it contains (identifies failed devices)
   - **connectivity show** - Displays the communication protocol endpoints that can see each other
   - **export port summary** - Summarizes any unhealthy ports
3. Optionally, use the **cluster configdump** command to dump the cluster configuration in an XML format.

WARNING

Using the cluster configdump command to dump a large configuration may take a long time.

The information collected by the **cluster configdump** command can be useful to identify problems in case of a failure. Administrators must weigh the value of the information collected against the amount of time required to dump a large configuration when deciding whether to perform a configdump.

IMPORTANT

No modifications should be made to VPLEX during the remainder of the backup procedure. Make sure that all other users are notified.

4. Use the **ll** command in the system-volumes context to verify that the meta-volume is **Active** and its **Ready** state is true.

   For example:

```
VPlexcli:/clusters/cluster-1/system-volumes> ll
```

Name	Volume Type	Operational Status	Health State	Active	Ready	Geometry	Block Count	Block Size	Capacity	Slots
logging_1_vol	logging-volume	ok	ok	-	-	raid-0	20971520	4K	80G	-
metadata_1	meta-volume	ok	ok	true	true	raid-1	23592704	4K	90G	32000

5. Use the **meta-volume backup** command to back up the meta-volume

   meta-volume backup --storage-volumes *storage-volumes*

   For the *storage-volumes* value, type the system ID for two or more storage volumes identified in "Before you begin".

   For example:

```
VPlexcli:meta-volume backup -storage-volumes
VPD83T3:6006048000019030048753303035 4636,
VPD83T3:6006048000019030048753303034 3445
```

Back up the meta-volume: VPLEX Metro or Geo

Before you begin

Identify two or more storage volumes at each cluster to which to back up the metadata. Target storage volumes must be:

- Unclaimed
- 78 GB or larger

Open a second Putty session to each cluster to display the client log files at /var/log/Vplex/cli directory. Use these sessions to watch for call home events.

To back up the meta-volume for a two-cluster VPLEX Metro or Geo:

1. Log in to each cluster.
2. At each cluster, use the **ll** command in device migration and extent migration contexts to verify that there are no active migrations:

   ```
   VPlexcli:/data-migrations/device-migrations> ll
   VPlexcli:/data-migrations/extent-migrations> ll
   ```

   If any migrations are in-progress or queued:

   - Allow the migrations to complete, and commit them (see "Commit a completed migration"); or
   - Cancel the migrations (see "Cancel a migration (optional)") and remove them (see "Remove migration records").
3. Use the following commands to verify the overall health of VPLEX:
   - **validate-system-configuration** - Performs a basic system configuration check
   - **cluster status** - Displays a cluster's operational-status and health-state
   - **export storage-view summary** -Lists each view, and the number of volumes and initiators that it contains (identifies failed devices)
   - **connectivity show** - Displays the communication protocol endpoints that can see each other
   - **export port summary** - Summarizes any unhealthy ports
4. Optionally, use the **cluster configdump** command to dump the cluster configuration in an XML format.

WARNING

Using the cluster configdump command to dump a large configuration may take a long time.

The information collected by the **cluster configdump** command can be useful to identify problems in case of a failure. Administrators must weigh the value of the information collected against the amount of time required to dump a large configuration when deciding whether to perform a configdump.

IMPORTANT

No modifications should be made to VPLEX during the remainder of the backup procedure. Make sure that all other users are notified.

5. At each cluster, use the **ll** command in the system-volumes context to verify that the status of the cluster's meta-volume is **Active** and **Ready** state is true.

For example:

```
VPlexcli:/clusters/cluster-1/system-volumes> ll
```

Name	Volume Type	Operational Status	Health State	Active	Ready	Geometry	Block Count	Block Size	Capacity	Slots
new_metal	meta-volume	ok	ok	true	true	raid-1	20447744	4K	78G	32000
new_metal_backup_2010May24_163810	meta-volume	ok	ok	false	true	raid-1	20447744	4K	78G	32000

6. Use the **meta-volume backup** command to back up the meta-volume at each cluster:

```
meta-volume backup --storage-volumes storage-volumes --cluster cluster
```

For the *storage-volumes* value, type the system ID of one or more storage volumes identified in "Before you begin".

Type the storage volume IDs separated by commas.

For example, at cluster-1:

```
VPlexcli:/clusters/cluster-1/system-volumes>  meta-volume backup -storage-volumes
VPD83T3:60000970000194900383533030454342,VPD83T3:60000970000194900383533030454341 --cluster
cluster-1
```

IMPORTANT

Perform backup of the meta-volumes at the two clusters in quick succession.

7. Use the **ll** command to display the new meta-volume at each cluster:

```
VPlexcli:/clusters/cluster-1/system-volumes> ll
```

Name	Volume Type	Operational Status	Health State	Active	Ready	Geometry	Block Count	Block Size	Capacity	Slots
new_meta1	meta-volume	ok	ok	true	true	raid-1	20447744	4K	78G	32000
new_meta1_backup_2010May24_163810	meta-volume	ok	ok	false	true	raid-1	20447744	4K	78G	32000

8. The default name assigned to the backup meta-volume includes a timestamp. Verify that the timestamp for the backup meta-volumes at the two clusters are in quick succession.
9. Use the second Putty session to verify that no call home events were sent during the backups.

   If a CallHome event was sent, use the **meta-volume destroy** command to delete the new meta-volume on each cluster and start over at Step 2 .

```
VPlexcli:/clusters/cluster-1/system-volumes> meta-volume destroy
new_meta_data_backup_2010May24_163810
```

Move a meta-volume

To move a meta-volume from one storage volume to another:

1. Use the **ll** command to display a list of storage volumes on the cluster:

```
VPlexcli:/> ll /clusters/cluster-1/storage-elements/storage-volumes

/clusters/cluster-1/storage-elements/storage-volumes:
Name           VPD83 ID                                 Capacity  Use    Vendor IO    Type
-------------- ---------------------------------------- --------  -------  -----  ------  ------  ----------
Clar0068_LUN71 VPD83T3:6006016049e02100281ebe77852cdf11  78G       meta-data  DGC    alive   traditional
Clar0068_LUN74 VPD83T3:6006016049e02100291ebe77852cdf11  78G       meta-data  DGC    alive   traditional
Clar0068_LUN75 VPD83T3:6006016049e02100c064c78a852cdf11  78G       unclaimed  DGC    alive   normal
Clar0068_LUN76 VPD83T3:6006016049e02100c164c78a852cdf11  78G       unclaimed  DGC    alive   normal
.
.
.
```

2. Identify 2 storage volumes that are:
   - Unclaimed
   - 78 GB or larger
   - On different arrays
3. Use the **meta-volume create** command to create a new meta-volume.

   Specify the storage volumes identified in Step 2 .

```
VPlexcli:/engines/engine-1-1/directors> meta-volume create --name meta_dmx --storage-volumes
VPD83T3:60060160372022009 66da1373865de11,
VPD83T3:60060160372022009 66da1373865de12
```

   See "Create a meta-volume" on page 36.

4. Use the **meta-volume move** command to move the existing in-memory metadata to the new meta-volume:

```
VPlexcli:/engines/engine-1-1/directors> meta-volume move --target-volume meta_dmx
```

Rename a meta-volume

By default, meta-volume names are based on a timestamp. To change the name, do the following:

1. Navigate to the /clusters/*cluster*/system-volumes/ context:

   ```
   VPlexcli:/> cd clusters/cluster-2/system-volumes/

   VPlexcli:/clusters/cluster-2/system-volumes>
   ```

2. Use the **ll** command to display the names of the meta-volumes.
3. Navigate to the /clusters/*cluster*/system-volumes/ *target-meta-volume* context.

   For example:

   ```
   VPlexcli:/clusters/cluster-1/system-volumes> cd
   new_meta1_backup_2010May24_163810
   ```

4. Use the **set name** ***new_meta-volume_name*** command to change the name.

   For example:

   ```
   VPlexcli:/clusters/cluster-1/system-volumes/new_meta1_backup_2010M
   ay24_163810> set name backup_May24_pre_refresh
   ```

Delete a meta-volume

IMPORTANT

A meta-volume must be *inactive* in order to be deleted. Attempts to delete an active meta-volume fail with an error message.

To delete a meta-volume, do the following:

1. Navigate to the target volume's context.

   For example:

   ```
   cd clusters/cluster-1/system-volumes/metadata_1/
   ```

2. Use the **ll** command to verify that the volume is not active.

   For example:

   ```
   /clusters/cluster-1/system-volumes/metadata_1> ll

   Attributes:
   Name                    Value
   ----------------------  -----------
   active                  false
   application-consistent  false
   block-count             23592704
   block-size              4K
   .
   .
   .
   ```

3. Use the **meta-volume destroy --meta-volume** *meta-volume* command to delete the specified meta-volume.

   For example:

   ```
   meta-volume destroy --meta-volume metadata_1
   ```

   A warning message appears:

   ```
   Meta-volume 'metadata_1' will be destroyed. Do you wish to continue?
   (Yes/No)
   ```

4. Type **y**.

Display meta-volume

Use the **ll** command to display status for a meta-volume:

```
VPlexcli:/clusters/cluster-1/system-volumes/ICO_META_1_1_Metadata> ll

/clusters/cluster-1/system-volumes/ICO_META_1_1_Metadata:

Attributes:
Name                     Value
-----------------------  -------------
active                   true
application-consistent   false
block-count              24511424
block-size               4K
capacity                 79.5G
component-count          2
free-slots               31968
geometry                 raid-1
health-indications       []
health-state             ok
locality                 local
operational-status       ok
ready                    true
rebuild-allowed          true
rebuild-eta              -
rebuild-progress         -
rebuild-status           done
rebuild-type             full
slots                    32000
stripe-depth             -
system-id                ICO_META_1_1_Metadata
transfer-size            2M
volume-type              meta-volume


Contexts:
Name        Description
----------  ------------------------------------------------------------------
components  The list of components that support this device or system virtual
            volume.
```

Use the **ll components/** command to display the component volumes of the meta-volume:

```
VPlexcli:/clusters/cluster-2/system-volumes/ICO_META_1_1_Metadata> ll components/

/clusters/cluster-2/system-volumes/clus2_MetaVol/components:
Name                                      Slot    Type            Operational  Health  Capacity
                                          Number  --------------  Status       State   --------
----------------------------------------  ------  --------------  -----------  ------  --------
VPD83T3:60000970000192601707533031333136  0       storage-volume  ok           ok      78G
VPD83T3:60060480000190300487533030343445  1       storage-volume  ok           ok      78G
```

Table 2 meta-volume display fields

Field	Description
active	Indicates whether this is the currently-active metadata volume. The system has only one active metadata volume at a time.
application-consistent	Whether or not this storage-volume is application-consistent.
block-count	The number of blocks in the volume.

Table 2 meta-volume display fields

Field	Description
capacity	The size of the meta-volume.
component-count	The number of mirrors in this raid-1 meta-data volume.
free-slots	The number of free slots for storage-volume headers in this meta-volume.
geometry	Indicates the geometry or redundancy of this device. Will always be raid-1.
health-indications	If health-state is not “ok”, additional information.
health-state	**ok** - The storage volume is functioning normally. **degraded** - The storage volume may be out-of-date compared to its mirror. (This state applies only to a storage volume that is part of a RAID-1 Metadata Volume.) **unknown** - VPLEX cannot determine the storage volume's Health state, or the state is invalid. **non-recoverable error** - The storage volume may be out-of-date compared to its mirror (applies only to a storage volume that is part of a RAID-1 Metadata Volume), and/or VPLEX cannot determine the Health state. **critical failure** - VPLEX has marked the storage volume as hardware-dead.
locality	Locality of the supporting device. **local** - The volume is local to the enclosing cluster. **remote** - The volume is made available by a different cluster than the enclosing cluster, and is accessed remotely. **distributed** - The virtual volume either has, or is capable of having, legs at more than one cluster. *}
operational status	**ok** - The storage volume is functioning normally. **degraded** - The storage volume may be out-of-date compared to its mirror. (This state applies only to a storage volume that is part of a RAID-1 Metadata Volume.) **unknown** - VPLEX cannot determine the storage volume's Health state, or the state is invalid. **error** - VPLEX has marked the storage volume as hardware-dead. **starting** - The storage volume is not yet ready. **lost-communication** - The storage volume is unreachable.
ready	Indicates whether this metadata volume is ready or not.
rebuild-allowed	Whether or not this device is allowed to rebuild.
rebuild-eta	If a rebuild is in progress, the estimated time remaining for the current rebuild to complete.
rebuild-progress	If a rebuild is in progress, the percentage of this device that has been rebuilt.
rebuild-status	The rebuild status of this device. **done** - Rebuild is complete.
rebuild-type	The rebuild type. **full** - A full copy of all the blocks. **incremental** - An incremental copy uses a checksum differencing algorithm to transfer only those blocks that are different. **comparison** - A comparison copy. **resync** - A resync rewrites blocks that may have been affected by a director failure, guaranteeing that the mirror legs are identical.
slots	The total number of slots for storage-volume headers in this meta-volume.
stripe-depth	The depth of a stripe in bytes when 'geometry' is 'raid-0'.

Table 2 meta-volume display fields

Field	Description
system-id	Name assigned to the meta-volume.
transfer-size	The transfer size during rebuild in bytes. See "About transfer-size" on page 121.
volume-type	For meta-volumes, this is always 'meta-volume'.

4

System Management

This chapter describes how to use the VPLEX CLI to manage battery conditioning, call-home notifications and system reporting, event log locations, and hardware acceleration with VAAI.

SPS battery conditioning

A standby power supply (SPS) battery conditioning cycle consists of a 5 minute period of on-battery operation followed by 6 hours of recharge.

Battery conditioning verifies the health of the batteries and extends their operational life.

Each SPS battery in a VPLEX system is automatically conditioned once a month.

Battery conditioning is enabled by default, but can be disabled for some activities (maintenance, system upgrades) where both SPS units are required.

In addition to the monthly automatic conditioning cycles, manually requested conditioning cycles can be scheduled and cancelled.

Battery conditioning cycle calendar

The conditioning cycle effectively moves the target battery into discharge state. The automatic conditioning cycle is scheduled so as not to have more than one active battery conditioning cycle (and thus more than one battery in discharge state) at any one time.

In addition to automatic battery conditioning cycles, additional cycles can be manually requested.

Automatic battery conditioning cycles are "checker boarded" into 6 hour windows:

- An SPS on the A side is scheduled to run in the first window, followed by
- A window that allows manual tests on the A side, followed by
- An SPS on the B side, followed by
- A window that allows manual tests on the B side SPS.

Time windows for manual tests allow only one side (A or B) to run conditioning cycles in a given period.

Figure 1 shows the conditioning cycle calendar for a typical month:

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
	30	31	1	2	3	4	5
12:00 AM							
6:00 AM	Manual A-side	Manual A-side	Manual A-side	Manual A-side	Manual A-side	Manual A-side	Manual A-side
12:00 PM							
6:00 PM	Manual B-side	Manual B-side	Manual B-side	Manual B-side	Manual B-side	Manual B-side	Manual B-side
	6	7	8	9	10	11	12
12:00 AM		**SPS 1-A**					
6:00 AM	Manual A-side	Manual A-side	Manual A-side	Manual A-side	Manual A-side	Manual A-side	Manual A-side
12:00 PM		**SPS 1-B**					
6:00 PM	Manual B-side	Manual B-side	Manual B-side	Manual B-side	Manual B-side	Manual B-side	Manual B-side
	13	14	15	16	17	18	19
12:00 AM		**SPS 2-A**					
6:00 AM	Manual A-side	Manual A-side	Manual A-side	Manual A-side	Manual A-side	Manual A-side	Manual A-side
12:00 PM		**SPS 2-B**					
6:00 PM	Manual B-side	Manual B-side	Manual B-side	Manual B-side	Manual B-side	Manual B-side	Manual B-side
	20	21	22	23	24	25	26
12:00 AM		**SPS 3-A**					
6:00 AM	Manual A-side	Manual A-side	Manual A-side	Manual A-side	Manual A-side	Manual A-side	Manual A-side
12:00 PM		**SPS 3-B**					
6:00 PM	Manual B-side	Manual B-side	Manual B-side	Manual B-side	Manual B-side	Manual B-side	Manual B-side
	27	28	29	30	1	2	3
12:00 AM		**SPS 4-A**					
6:00 AM	Manual A-side	Manual A-side	Manual A-side	Manual A-side	Manual A-side	Manual A-side	Manual A-side
12:00 PM		**SPS 4-B**					
6:00 PM	Manual B-side	Manual B-side	Manual B-side	Manual B-side	Manual B-side	Manual B-side	Manual B-side

Figure 1 **Typical monthly battery conditioning schedule**

When it is safe to start a conditioning cycle

CAUTION

If an SPS is put on battery power when it is unsafe to do so, the risk of data unavailability and data loss is increased.

There are several conditions that must be met in order to safely start a conditioning cycle:

- The SPS must have 6 hours to fully charge before the allotted conditioning time expires. Conditioning cycles (including manually requested cycles) start at the beginning of their scheduled time slot.
- The SPS must not have failed a previous conditioning cycle or have any internal failures.
- All power components in the engine related to the SPS must be healthy.

- No maintenance, hardware replacement, or system upgrade can occur during a conditioning cycle.

  Starting a conditioning cycle during maintenance or system upgrades could disrupt these operations.

When to stop a conditioning cycle

Stop a conditioning cycle once it has started when:

- A power component in the associated engine becomes unhealthy.

  This could be a hardware fault in one of the engine's director power supplies or a power loss in the peer SPS.

  A power disruption automatically aborts any ongoing SPS conditioning.

- Manual intervention is required due to unscheduled maintenance or an impending disaster.

  If an SPS that is currently being conditioned loses AC power, the engine will behave normally and continue to be powered by the peer SPS.

Additional documentation

Refer to the *VPLEX CLI Guide* for information about the CLI commands related to battery conditioning:

- **battery-conditioning set-schedule** - Sets the battery conditioning schedule (day of week) for backup battery units on a cluster.
- **battery-conditioning enable** - Enables conditioning on the specified backup battery unit(s).
- **battery-conditioning disable** - Disables battery conditioning on the specified backup battery unit(s).
- **battery-conditioning manual-cycle request** - Manually requests a battery conditioning cycle on the specified backup battery unit.
- **battery-conditioning manual-cycle cancel-request** - Cancels a manually requested battery conditioning cycle on the specified backup battery unit.
- **battery-conditioning summary** - Displays a summary of the battery conditioning schedule for all devices, grouped by cluster.

Refer to the VPLEX generator for the procedures to:

- Set the battery conditioning schedule
- Enable battery conditioning
- Disable battery conditioning
- Manually request an additional conditioning cycle
- Cancel a manually requested cycle

Call-home notifications and system reporting

About call-home notifications

Call-home notifications are messages sent automatically from VPLEX to EMC Customer Service and/or customer personnel when a serious problem occurs. Call-home notifications enable EMC to pro-actively engage the relevant personnel, or use a configured ESRS gateway to resolve the problem.

There are 4 levels of system events. Call-home notifications are sent only for three levels:

Table 3 Event severity and Call-home notifications

Severity	Definition	Impact on Performance or Availability	Call-home
Critical (1)	A DU or DL is either highly probable or has occurred.	System unavailable. Severe performance degradation.	Yes
Error (2)	Possible DU or DL. Requires service intervention.	Limited performance impact. Loss of redundancy. Moderate risk of DU/DL.	Yes
Warning (3)	Service attention required. No urgency.	No performance impact. Loss of redundancy. No risk of DU/DL.	Yes
Info (4)	Informational event. No action is required.	None.	No

Refer to the VPLEX generator *Troubleshooting Procedures > Events and Messages* for a list of all events.

Many maintenance activities (such as hardware replacements) generate a flurry of call-home events. Many such procedures include steps to temporarily disable call-home during the operation.

If the same event on the same component occurs repeatedly, a call-home is generated for the first instance of the event, and not again for 8 hours (480 minutes).

For example, if event E1 occurs on a director D1 at the time T1, a call-home is generated. If the same event E1 is generated on the same component D1 at the time T1 + N minutes, where N < 480, no call-home is generated.

The interval N is tracked by the management server. If the management server fails, the counter is reset to 8 hours. After recovery from a management server failure, a call-home event is sent for the same event/component, even though 8 hours may not have elapsed since the first call-home for that event/component.

About customized call-home events

You can import an EMC-provided .xml file to customize events severity level and text messages.

There are two types of .xml event files:

- EMC-generic events are modifications recommended by EMC.

  EMC provides an .xml file containing commonly requested modifications to the default call-home events.

- Customer-specific events are events modified to meet a specific customer requirement.

EMC provides a custom events file developed by EMC engineering and applied by EMC Technical Support.

Call-home behaviors changes immediately when the modified events file is applied.

If a customized events file is already applied, applying a new file overrides the existing file.

If the same event is modified in the customer-specific and EMC-generic file, the modification specified for that event in the customer-specific file is applied.

If call-home is disabled when the custom events file is applied, the modified events are saved and applied when call-home is enabled.

About system (SYR) reporting

System reporting (SYR) collects and transmits two types of information from VPLEX systems to EMC:

- **System reports** - Sent once weekly to the EMC System Reports database. System reports include information about the configuration and state of the system.
- **System alerts** - Sent in real-time through a designated SMTP server to the EMC. Alerts are filtered as to whether a service request should be opened with EMC Customer Service. If a service request is required, it is opened automatically.

SYR is enabled by default, but can be disabled at any time through the GUI or CLI.

Modify call-home and SYR

Call-home and SYR settings are typically configured during system set-up.

Use the to **configuration event-notices-reports-config** CLI command to configure the call-home and/or SYR settings if they were not configured during the initial installation.

The command runs an interview script that prompts for the required information. If either call-home or SYR is not configured, interview questions to configure the service that is not configured are displayed.

If both call-home and SYR are already configured, the current configuration information is displayed.

Before you begin

You will need the following information to complete configuration of call-home and SYR reporting:

- IP address of the primary SMTP server used to forward reports to EMC. EMC recommends using your ESRS gateway as the primary connection address.
- (Optional) One or more IP addresses of secondary SMTP server(s) used to forward reports to EMC if the primary server fails. These addresses must be different than the address for the primary SMPTP server.
- (Optional) One or more e-mail addresses of personnel who should receive e-mail notifications when events occur.

Additional documentation

Refer to the VPLEX generator for the procedure to configure SYR:

Refer to the *VPLEX CLI Guide* for information about the CLI commands related to call-home notifications and SYR reporting:

- **configuration event-notices-reports-config** - Configure call-home and SYR settings after the initial configuration of VPLEX,
- **configuration event-notices-reports-reset** - Resets the current event notification and reporting configuration.
- **notifications call-home import-event-modification** - Imports and applies modified call-home events.
- **notifications call-home remove-event-modifications** - Removes customized call-home events files, including customer-specific modifications and modifications recommended by EMC.
- **notifications call-home view-event-modifications** - Displays any customized call-home events.

Event log locations

VPLEX includes services, processes, components, and operating systems that write entries to various logs.

Logs are collected for:

- Scheduled activities: SYR collection
- On-demand utilities: collect-diagnostics
- Call-home events

The locations of various logs on the VPLEX management server are listed in Table 4:

Table 4 VPLEX log file locations

Log name	Description and location
Firmware log	Includes all entries from the entire VPLEX system. Messages are expanded. On a running management server: `/var/log/VPlex/cli/firmware.log*` In collect-diagnostics output: `smsDump_<datestamp>-<timestamp>\clilogs\`

Call home log	On a running management server:
	`/opt/emc/connectemc/logs/ConnectEMC.log`
	`/opt/emc/connectemc/archive/*.xml`
	`/opt/emc/connectemc/failed/*.xml`
	`/opt/emc/connectemc/poll/*.xml`
	In collect-diagnostics output:
	`smsDump_<datestamp>-<timestamp>\connectemc\`

ZPEM Log	Zephyr Power and Environmental Monitoring log. ZPEM sends events to a syslog-ng service on the director. The syslog-ng service writes ZPEM entries to log files in /var/log and also them to EMC Common Object Manager (ECOM), which streams the log entries to the cluster management server to be written to the firmware log.
	On a running director:
	`/var/log/zpem.log`
	In collect-diagnostics output:
	`\<director_name>-<datestamp>-<timestamp>\var\log`

NSFW log	GeoSynchrony log. NSFW sends events to a syslog-ng service on the director. The syslog-ng service writes NSFW entries to log files in /var/log and also them to EMC Common Object Manager (ECOM), which streams the log entries to the cluster management server to be written to the firmware log.
	On a running director:
	`/var/log/nsfw.log`
	In collect-diagnostics output:
	`\<director_name>-<datestamp>-<timestamp>\var\log`

DMI log	On a running director: accessed through the zdt utility In collect-diagnostics output: `\<director_name>-diagnostics-<datestamp>-<timestamp>.log`
ZPEM Trace log	ECOM writes trace logs to a cimom and ecomofl log files. ZPEM writes trace logs to a ZTrace log. These trace logs are not part of the event logging system. In collect-diagnostics output: `\<director_name>-<datestamp>-<timestamp>\var\log`

Hardware acceleration with VAAI

VMware API for Array Integration (VAAI) allows you to:

- Offload storage operations from compute side to storage hardware,
- Shift provisioning and snapshotting from hypervisor to VPLEX,
- Dedicate hypervisor memory and processing resources to other functions.

On VPLEX, VAAI is implemented using two SCSI commands:

- "Compare and Write" offloads coordination of powering virtual machines (VMs) on/off, and moving them between ESX servers.
- "WriteSame (16)" offloads copying data to and from the array through the hypervisor.

Compare and Write

The CompareAndWrite (CAW) SCSI command is used to coordinate VMware operations such as powering-on/off VMs, moving VMs from one ESX to another without halting applications (VMotion), and Distributed Resource Scheduler (DRS) operations.

CAW is used by VMWare ESX servers to relieve storage contention, which may be caused by SCSI RESERVATION in distributed VM environments. CAW assists storage hardware acceleration by allowing ESX servers to lock a region of disk instead of entire disk.

ESX 5.0 servers use this strategy to increase the number of VMs an ESX servers can host, and to increase the performance of those VMs.

Starting in GeoSynchrony 5.1, VPLEX support for CAW is enabled by default.

Enabling/disabling CAW

CAUTION

CAW can be enabled/disabled on VPLEX only by EMC Technical Support personnel.

VMware servers discover whether the CAW SCSI command is supported:

- During initial storage scanning
- When the VMFS3.HardwareAcceleratedLocking value on the ESX host is enabled (or toggled if it is enabled)

Note: To toggle the value: In the vSphere client, toggle host > Configuration > Software > Advanced Settings > VMFS3.HardwareAcceleratedLocking value to 0 and then 1.

If CAW is not supported or support is disabled, VPLEX returns CHECK CONDITION, ILLEGAL REQUEST, and INVALID OP-CODE. The ESX server reverts to using SCSI RESERVE and the VM operation continues.

VM operations may experience significant performance degradation if CAW is not enabled.

VPLEX enables CAW to be enabled/disabled for all storage associated with VPLEX, using a single command. When CAW is disabled on VPLEX, VPLEX storage volumes, do not include CAW support information in their responses to inquiries from hosts.

To mark storage CAW disabled:

- VMFS3.HardwareAcceleratedLocking must be toggled, or
- Hosts may need to rescan their storage.

CAUTION

Enabling/disabling CAW functionality supports exceptional situations such as assisting EMC Technical Support personnel to diagnose a problem. CAW is enabled by default and should be disabled only by EMC Technical Support.

Support for CAW can be enabled or disabled at two levels:

- **storage-view** - Enabled or disabled for all existing storage views. A storage view created after CAW is enabled/disabled at the storage view level inherits the system default setting. EMC recommends maintaining uniform CAW setting on all storage views in VPLEX. If CAW must be disabled for a given storage view, it must be disabled on all existing and future storage views. To make future storage views to reflect the new setting, change the system default (described below).
- **system default** - Enabled or disabled as a system default. A storage view created after CAW is enabled/disabled at the system default level inherits the system default setting. If the system default is enabled, CAW support for the new storage view is also enabled

Display CAW setting

Use the **ls** command in /clusters/*cluster*/exports/storage-views context to display whether CAW is enabled at the storage view level. For example:

```
VPlexcli:/> ll /clusters/cluster-2/exports/storage-views/*

/clusters/cluster-2/exports/storage-views/FE-Logout-test:
Name                        Value
--------------------------  ------------------------------------------------------------------
caw-enabled                 false
.
.
.
/clusters/cluster-2/exports/storage-views/default_quirk_view:
Name                        Value
--------------------------  ------------------------------------------
caw-enabled                 false
.
.
.
```

Use the **ls** command in /clusters/*cluster* context to display the CAW system default setting:

```
VPlexcli:/> ls /clusters/cluster-1

/clusters/cluster-1:

Attributes:
Name                        Value
--------------------------  ------------------------------------------
allow-auto-join             true
auto-expel-count            0
auto-expel-period           0
auto-join-delay             0
cluster-id                  1
connected                   true
default-cache-mode          synchronous
default-caw-template        true
.
.
```

.

Enable/disable CAW for a storage view

Use the **set** command in /clusters/*cluster*/exports/storage-views/*storage-view* context to enable or disable CAW for the storage view.

To enable CAW for a storage view:

```
VPlexcli:/clusters/cluster-1/exports/storage-views/recoverpoint_vols>
   set caw-enabled true
```

To disable CAW for a storage view:

```
VPlexcli:/clusters/cluster-1/exports/storage-views/recoverpoint_vols>
   set caw-enabled false
```

Enable/disable CAW as system default

Use the **set** command in /clusters/*cluster* context to enable or disable CAW for the entire cluster.

To enable CAW as the cluster system default:

```
VPlexcli:/clusters/cluster-1> set default-caw-template true
```

To disable CAW as the cluster system default:

```
VPlexcli:/clusters/cluster-1> set default-caw-template false
```

CAW statistics

CAW performance statistics are included for front-end volume (fe-lu), front-end port (fe-prt), and front-end director (fe-director) targets.

See Front-end volume (fe-lu) statistics on page 265, Front-end port (fe-prt) statistics on page 266, and Front-end director (fe-director) statistics on page 265

Statistics for fe-director targets are collected as a part of the automatically created perpetual monitor.

You can create a monitor to collect CAW statistics, which can be especially useful for fe-lu targets (because there can be very large numbers of volumes involved, these statistics are not always collected). See Example: Send CAW statistics to the management server on page 248

WriteSame (16)

The WriteSame (16) SCSI command provides a mechanism to offload initializing virtual disks to VPLEX. WriteSame (16) requests the server to write blocks of data transferred by the application client multiple times to consecutive logical blocks.

WriteSame (16) is used to offload VM provisioning and snapshotting in vSphere to VPLEX.

WriteSame (16) enables the array to perform copy operations independently without using host cycles. The array can schedule and execute the copy function much more efficiently.

VPLEX support for WriteSame (16) is enabled by default.

Enabling/disabling WriteSame (16)

CAUTION

WriteSame (16) can be enabled/disabled on VPLEX only by EMC Technical Support personnel.

VMware servers discover whether the WriteSame (16) SCSI command is supported:

- During initial storage scanning
- When the VMFS3.HardwareAcceleratedLocking value on the ESX host is enabled (or toggled if it is enabled)

Note: To toggle the value: In the vSphere client, toggle host > Configuration > Software > Advanced Settings > VMFS3.HardwareAcceleratedLocking value to 0 and then 1.

VM operations may experience significant performance degradation if WriteSame (16) is not enabled.

VPLEX allows WriteSame (16) to be enabled/disabled for all storage associated with VPLEX, using a single command. When WriteSame (16) is disabled on VPLEX, VPLEX storage volumes, do not include WriteSame (16) support information in their responses to inquiries from hosts.

Support for WriteSame (16) can be enabled or disabled at two levels:

- **storage-view** - Enabled or disabled for all existing storage views. A storage view created after WriteSame (16) is enabled/disabled at the storage view level inherits the system default setting. EMC recommends maintaining uniform WriteSame (16) setting on all storage views in VPLEX.

  If WriteSame (16) must be disabled for a given storage view, it must be disabled on all existing and future storage views. To make future storage views to reflect the new setting, change the system default (described below).

- **system default** - Enabled or disabled as a system default. A storage view created after WriteSame (16) is enabled/disabled at the system default level inherits the system default setting. If the system default is enabled, WriteSame (16) support for the new storage view is also enabled.

CAUTION

To disable the Write Same 16 default template, you MUST disable Write Same 16 for all existing views, and disable Write Same 16 template so all future views will be Write Same 16 disabled.

To enable the Write Same 16 default template, you MUST enable Write Same 16 for all existing views, and enable Write Same 16 template so all future views will be Write Same 16 enabled.

Display WriteSame (16) setting

Use the **ls** command in /clusters/*cluster*/exports/storage-views context to display whether WriteSame (16) is enabled at the storage view level. For example:

```
VPlexcli:/> ll /clusters/cluster-2/exports/storage-views/*

/clusters/cluster-2/exports/storage-views/FE-Logout-test:
Name                        Value
--------------------------  ------------------------------------------------------------------
```

```
caw-enabled                 false
.
.
.
/clusters/cluster-2/exports/storage-views/default_quirk_view:
Name                        Value
--------------------------  ----------------------------------------
.
.
.
write-same-16-enabled       false
```

Use the **ls** command in /clusters/*cluster* context to display the WriteSame (16) system default setting:

```
VPlexcli:/> ls /clusters/cluster-1

/clusters/cluster-1:

VPlexcli:/clusters/cluster-1> ls

Attributes:
Name                              Value
-------------------------------
   ------------------------------------------------
allow-auto-join                   true
auto-expel-count                  0
auto-expel-period                 0
auto-join-delay                   0
cluster-id                        1
connected                         true
default-cache-mode                synchronous
default-caw-template              true
default-write-same-16-template    false
.
.
.
```

Enable/disable WriteSame (16) for a storage view

Use the **set** command in /clusters/*cluster*/exports/storage-views/*storage-view* context to enable or disable WriteSame (16) for the storage view.

To enable WriteSame (16) for a storage view:

```
VPlexcli:/clusters/cluster-1/exports/storage-views/recoverpoint_vols>
   set write-same-16-enabled true
```

To disable WriteSame (16) for a storage view:

```
VPlexcli:/clusters/cluster-1/exports/storage-views/recoverpoint_vols>
   set write-same-16-enabled false
```

Enable/disable WriteSame (16) as system default

Use the **set** command in /clusters/*cluster* context to enable or disable CAW for the entire cluster.

To enable CAW as the cluster system default:

```
VPlexcli:/clusters/cluster-1> set
   default-write-same-16-enabled-template true
```

To disable CAW as the cluster system default:

```
VPlexcli:/clusters/cluster-1> set
   default-write-same-16-enabled-template false
```

5

Distributed devices

This chapter provides procedures to manage distributed devices using VPLEX CLI.

Additional documentation

- Refer to the *EMC VPLEX CLI Guide* for detailed information about the CLI commands to create and manage distributed devices.
- Refer to the *EMC VPLEX Product Guide* for general information about distributed devices.
- Refer to the VPLEX generator for the following procedures:
  - Expand a distributed GeoSynchrony volume
  - Move a logging volume
  - Convert a local synchronous volume to an asynchronous DR1

About distributed devices

Distributed devices have their underlying storage arrays located at both clusters in a VPLEX Metro or VPLEX Geo.

Distributed devices support virtual volumes, that are presented to a host through a storage view. From the host, the virtual volumes appear as single volumes located on a single array.

You can configure up to 8000 distributed devices in a VPLEX system. That is, the total number of distributed virtual volumes plus the number of top-level local devices must not exceed 8000.

All distributed devices must be associated with a logging volume. During a link outage, the logging volume is used to map the differences between the legs of a DR1.

When the link is restored, the legs are resynchronized using the contents of their logging volumes.

All distributed devices must have a detach rule-set to determine which cluster continues I/O when connectivity between clusters is lost.

Logging volumes

This section describes the following topics:

About logging volumes

Logging volumes are required at each cluster before a distributed device can be created.

Logging volumes keep track of blocks written during:

- An inter-cluster link outage, or
- When one leg of a DR1 becomes unreachable and then recovers.

After the inter-cluster link or leg is restored, the VPLEX system uses the information in logging volumes to synchronize the mirrors by sending only changed blocks across the link.

Logging volumes also track changes during loss of a volume when that volume is one mirror in a distributed device.

Note: Logging volumes are not used to optimize re-syncs on local RAID-1s.

Single-cluster systems and systems that do not have distributed devices do not require logging volumes.

During and after link outages, logging volumes are subject to high levels of I/O. Thus, logging volumes must be able to service I/O quickly and efficiently.

EMC recommends:

- Place logging volumes on the fastest storage available.
- Stripe logging volumes across several disks to accommodate the high level of I/O that occurs during and after link outages.
- Mirror logging volumes across two or more back-end arrays, as they are critical to recovery after the link is restored.

  Arrays for mirroring logging volumes should be chosen such that they will not need to migrate at the same time.

Logging volume size

Logging volumes must be sized as follows:

One bit for every page of distributed storage space.

For example, for a VPLEX Metro configuration with 160 TB of distributed devices, the logging volumes at each cluster should be sized ~ 10 GB.

Create a logging volume

To create a logging volume, first claim the storage volumes that will be used, and create extents from those volumes.

- Use the **ll /clusters/*cluster*/storage-elements/ storage-volumes** command to display the available storage volumes on the cluster.
- Use the **storage-volume claim -n *storage-volume_name*** command to claim one or more storage volumes.
- Use the **extent create -d *storage-volume_name*, *storage-volume_name*** command to create an extent to use for the logging volume.

  Repeat this step for each extent to be used for the logging volume.

Use the **logging-volume create** command to create a logging volume. The syntax for the command is:

```
logging-volume create --name name --geometry [raid-0 |raid-1]
   --extents context-path --stripe-depth
```

--name - The name for the new logging volume.

--geometry - The geometry for the new volume. Valid values are **raid-0** or **raid-1.**

--extents - Context paths to one or more extents to use to create the logging volume. Target extents must not be empty, and must all reside at the specified cluster.

--stripe-depth - Required if --geometry is raid-0. Stripe depth must be:

- Greater than zero, but not greater than the number of blocks of the smallest element of the RAID 0 device being created
- A multiple of 4 K bytes

A depth of 32 means 128 K (32 x 4K) is written to the first disk, and then the next 128 K is written to the next disk.

Concatenated RAID devices are not striped.

Best practice regarding stripe depth is to follow the best practice of the underlying array.

For example:

```
VPlexcli:/> cd clusters/cluster-1/system-volumes/
VPlexcli:/clusters/cluster-1/system-volumes> logging-volume create --name cluster_1_log_vol
--geometry raid-0 --extents extent_1 , extent_2 --stripe-depth 1
```

Add a mirror to a logging volume

To display the name of the logging volume for the current cluster, navigate to the system volume context and type **ll**:

```
VPlexcli:/> cd clusters/cluster-1/system-volumes/

VPlexcli:/clusters/cluster-1/system-volumes> ll

Name      Volume Type     Operational  Health  Active  Ready  Geometry  Block     Block  Capacity  Slots
--------  --------------  Status       State   ------  -----  --------  Count     Size   --------  -----
--------  --------------  -----------  ------  ------  -----  --------  --------  -----  --------  -----
log1_vol  logging-volume  ok           ok      -       -      raid-1    20448000  4K     78G       -
meta_1    meta-volume     ok           ok      true    true   raid-1    20447744  4K     78G       3200
```

Use the **logging-volume add-mirror** command to add a mirror to the specified logging volume. The syntax for the command is:

logging-volume add-mirror --logging-volume ***logging-volume*** **--mirror** ***mirror***

--logging-volume - Name of logging volume to which to add the mirror.

--mirror - The name or context path of the device or storage-volume extent to act as a mirror. Must be a top-level device or a storage-volume extent.

For example:

```
VPlexcli:/clusters/cluster-1/system-volumes>logging-volume add-mirror --logging-volume
log1_vol --mirror /clusters/cluster-2/ storage-elements/extents/extent_Symm2194_20F0_1
```

Delete a logging volume

Use the **logging-volume destroy --logging-volume** *logging-volume* command to delete the specified logging volume.

For example:

```
VPlexcli:/clusters/<cluster-n>/system-volumes>logging-volume destroy --logging-volume
cluster_6_log_vol
```

Rule-sets

This section describes the following topics:

About rule-sets

Rule-sets are predefined rules that determine which cluster continues I/O when connectivity between clusters is lost. Rule-set apply to devices that are not members of consistency groups.

A cluster loses connectivity to its peer cluster when:

- The inter-cluster link fails,
- The peer cluster fails.

Note: The **cluster expel** command also causes clusters to lose connectivity to one another.

When a loss of connectivity occurs, VPLEX:

- Suspends I/O to all distributed devices on both clusters, and
- Starts a delay timer.

If connectivity is not restored when the timer expires (default is 5 seconds), VPLEX:

- Resumes I/O on the leg of the distributed device (the "winning" cluster) as determined by the device's rule-set
- Keeps I/O suspended on the other leg (the "losing" cluster).

Writes to the distributed device from the losing cluster are suspended until connectivity is restored.

When the inter-cluster link is restored, the I/O written to the winning cluster is re-synchronized to the losing cluster.

The rules for determining the number of seconds to wait, which leg of a distributed device is resumed, and which remains suspended are contained in rule-sets.

A rule-set consists of a container (the rule-set) and one rule.

Rules have two attributes:

- **Delay** - The number of seconds between the link outage and when the actions defined by the rule-set (resume I/O to the winning cluster, keep I/O to the losing cluster suspended) begin. The default is 5 seconds.

- **Cluster** - The cluster on which I/O resumes (the winning cluster).

All distributed devices must have a rule-set. A cluster may be the "winning" cluster for some distributed devices, and the "losing" cluster for other distributed devices.

Most I/O workloads require specific sets of virtual volumes to resume on one cluster and remain suspended on the other cluster.

VPLEX includes two default rule-sets:

Table 5 Default rule-sets

Rule-set name	Cluster-1	Cluster-2
cluster-1-detaches	Services I/O	Suspends I/O
cluster-2-detaches	Suspends I/O	Services I/O

EMC recommends that only the two default rule-sets be applied to distributed devices.

The default value of a distributed device's rule-set is determined by the management server on which the device was created.

If a device is created on the management server for cluster-1, the default rule-set for that device is cluster-1-detaches.

Rule-sets are located in the distributed-storage/rule-sets context.

Rules are located under their rule-set context.

Auto-resume

When cluster connectivity is restored, subsequent behavior is determined by the auto-resume attribute for the distributed device.

- If auto-resume is set to false, the distributed device on the losing cluster remains suspended until I/O is manually resumed by the system administrator (using the device resume-link-up command).
- If auto-resume is set to true, I/O may start immediately after the link is restored.

After I/O is resumed to the mirror leg on the losing cluster, any data written to the device on the winning cluster during the outage is resynchronized from the leg on the winning cluster.

Use the **set** command to configure a device's auto-resume attribute.

Force I/O to resume on a suspended leg

Two commands allow administrators to force I/O to resume on a suspended mirror leg:

- **device resume-link-down** - Used when the inter-cluster link is down. Allows suspended mirror legs to resume I/O immediately.

  For example, use this command if the peer cluster is the winning cluster but is known to have failed.

- **device resume-link-up** - Used after the link has been restored.

  For example, use this command to the restart I/O to the mirror leg on the losing cluster when the auto-resume flag is set to false.

Rule-set best practices

- Use the provided pre-configured rule-sets whenever possible.

- For custom rule-sets, leave the detach delay timer at the default value of 5 seconds.

  Setting the detach delay lower than 5 seconds can result in unnecessary cluster detaches during periods of network instability. Multiple cluster detaches in a short period of time can result in unnecessary data rebuilds and reduced performance.

  Consult EMC Customer Support if a lower detach time is required.
- Configure detach rules based on the cluster/site that is expected to continue I/O during an outage.
- Avoid conflicting detach configurations. It is important to understand which cluster is configured to automatically detach.
- Document how rule sets are applied.

  It is acceptable for different distributed devices to resume I/O on different clusters during an outage.

CAUTION

If a host application uses more than one distributed device, all distributed devices for that application should have the same rule-set (to resume I/O on the same cluster).

Caution: conflicted detach

The major problem to avoid when creating and using rule-sets is a scenario referred to as a conflicted detach. Conflicted detach happens during a network outage when:

- Both legs of a distributed device perform I/O independently.
- Both clusters write to the different legs of the same virtual volume.
- When connectivity is restored, the administrator picks the winner cluster, meaning that one of the legs is used as the source to rebuild.
- Logging synchronizes the losing cluster to match the winning cluster.
- Any data written to the losing cluster during the network communication outage is overwritten.

Rule-sets and manual detaches must not result in conflicting detaches. Conflicting detaches result in lost data (on the losing cluster), a full rebuild and degraded performance during the full rebuild.

Rule-sets and local devices

A rule-set can be applied to a local device when the device's visibility is set to global.

Attaching a rule-set to a local device:

- Enables the cluster to detach in the event of a link outage
- Allows I/O to continue at the detached cluster.

VPLEX islands

Rule-sets introduce the term "island". An island is a set of communicating clusters.

A single cluster can be an island by itself.

For 2 clusters, there are 3 possible combinations, each of which can be an island.

- cluster-1
- cluster-2

- cluster-1 and cluster-2

For VPLEX Metro and Geo configurations, islands are mostly synonymous with clusters.

Manage rule-sets

The **ds rule-set create** command creates custom rule-sets. When a new rule-set is created, the VPLEX system creates a sub-context under the rule-sets context.

Rules themselves are added to these new sub-contexts.

After a rule is added to a rule-set, the rule can be applied (attached) to a distributed device.

Create a rule-set

Use the **ds rule-set create** ***rule-set-name*** command to create a new rule-set. The new rule-set is empty upon creation.

Note: The **ds rule-set create** command automatically creates a new sub-context, with the same name as the new rule-set.

In the following example, a new rule-set named TestRuleSet is created:

```
VPlexcli:/> ds rule-set create --name TestRuleSet

Name            PotentialConflict  UsedBy
--------------  -----------------  ------
TestRuleSet     false

VPlexcli:/>
```

Add rules to a rule-set

Use the **ds rule island-containing** command to add a rule that describes when to resume I/O on all clusters in the island containing the specified cluster.

Note: The **island-size rule** command creates rules for configurations with more than two clusters and is not supported in the current release.

1. Navigate to the target rule-set.

   For example:

```
VPlexcli:/> cd /distributed-storage/rule-sets/TestRuleSet

VPlexcli:/distributed-storage/rule-sets/TestRuleSet>
```

2. Use the **rule island-containing** command to add a rule to describe when to resume I/O on all clusters in the island containing the specified cluster. The syntax for the command is:

   rule island-containing --clusters ***context-path,context-path*** **--delay** ***delay***

   --clusters ***context-path,context-path*** - Context path(s) to the cluster or clusters to which this rule applies. Separated by commas.

   --delay ***delay*** - Sets the delay after a link outage before the rule is applied. Value must be a positive integer and end with one of the following units:

s - delay timer is seconds (default unit)

min - delay timer is seconds

h - delay timer is hours

In the following example, the **rule island-containing** command creates a rule that dictates the following behavior:

1. The VPLEX system waits 10 seconds after a link outage
2. Resumes I/O to the island containing cluster-1
3. Detaches any other islands.

```
VPlexcli:/distributed-storage/rule-sets/cluster1_Active> rule island-containing --clusters
cluster-1 --delay 10s
```

3. Use the **ll** command to display the new rule-set.

   For example:

```
VPlexcli:/distributed-storage/rule-sets/TestRuleSet> ll

Attributes:
Name                Value
------------------  ------------------------
key                 ruleset_5537985253109250
potential-conflict  false
used-by             []


Contexts:
Name   Description
-----  ----------------------------------
rules  The list of rules for this rule set.
```

4. Navigate to the rules context under the new rule-set context.

   For example:

```
VPlexcli:/distributed-storage/rule-sets/TestRuleSet> cd rules
```

5. Use the **ll** command to display the rules.

   For example:

```
VPlexcli:/distributed-storage/rule-sets/TestRuleSet/rules> ll
RuleName  RuleType          Clusters   ClusterCount  Delay  Relevant
--------  ----------------  ---------  ------------  -----  --------
rule_1    island-containing  cluster-1  -             5s     true
```

Note: To apply a rule-set to a distributed device when the device is created, see "Create a distributed device" on page 82.

To modify the rule-set applied to a distributed device, see "Modify the rule-set attached to a distributed device" on page 74.

Attach a rule-set to a distributed device

WARNING

If a host application uses more than one distributed device, configure all distributed devices for that application to resume I/O on the same cluster.

1. Navigate to the distributed devices context:

```
VPlexcli:/ cd/distributed-storage/distributed-devices
VPlexcli:/distributed-storage/distributed-devices>
```

2. Optionally, use the **ll** command to display the names of the distributed devices:

```
VPlexcli:/distributed-storage/distributed-devices> ll
Name   Status   Operational  Health  Auto    Rule Set Name       Transfer
-----  -------  Status       State   Resume  ------------------  Size
-----  -------  -----------  ------  ------  ------------------  --------
dd_00  running  ok           ok      true    cluster-1-detaches  2M
dd_01  running  ok           ok      true    cluster-1-detaches  2M
dd_02  running  ok           ok      true    cluster-1-detaches  2M
dd_03  running  ok           ok      true    cluster-1-detaches  2M
dd_04  running  ok           ok      true    cluster-1-detaches  2M
dd_05  running  ok           ok      true    cluster-1-detaches  2M
dd_06  running  ok           ok      true    cluster-1-detaches  2M
dd_07  running  ok           ok      true    cluster-1-detaches  2M
dd_08  running  ok           ok      true    cluster-2-detaches  2M
.
.
.
```

3. Use the **set** *device*::**rule-set-name** *rule-set-name* command to attach the specified rule-set to the specified device.

   In the following example, the **set** command attaches a rule-set named cluster1_Active to the device dd_00:

```
VPlexcli:/distributed-storage/distributed-devices> set dd_00::rule-set-name cluster1_Active
```

4. Use the **ll** command to verify the change:

```
VPlexcli:/distributed-storage/distributed-devices> ll
Name   Status   Operational  Health  Auto    Rule Set Name       Transfer
-----  -------  Status       State   Resume  ------------------  Size
-----  -------  -----------  ------  ------  ------------------  --------
dd_00  running  ok           ok      true    cluster1_Active     2M
dd_01  running  ok           ok      true    cluster-1-detaches  2M

.
.
.
```

Modify the rule-set attached to a distributed device

1. Navigate to the rule-set context:

```
VPlexcli:/> cd distributed-storage/rule-sets
VPlexcli:/distributed-storage/rule-sets>
```

2. Use the **ll** command to display available rule-sets.

3. Navigate to the context of the target distributed device:

```
VPlexcli:/distributed-storage/rule-sets>cd /distributed-storage/distributed-devices/dd_07
```

4. Use the **set rule-set-name** *rule-set-name* command to set or change its rule-set.

   For example:

```
VPlexcli:/distributed-storage/distributed-devices/dd_07> set rule-set-name cluster-1-detaches
```

5. Use the **ll** command to verify the change:

```
VPlexcli:/distributed-storage/distributed-devices/dd_07> ll

Attributes:
Name                    Value
```

```
----------------------  ----------------------
application-consistent  false
.
.
.
rule-set-name           cluster-1-detaches
.
.
.
```

Attach a rule-set to a local device

1. Navigate to the rule-set context:

```
VPlexcli:/> cd distributed-storage/rule-sets/
```

2. Use the **ll** command to display available rule-sets:

```
VPlexcli:/distributed-storage/rule-sets> ll
Name                 PotentialConflict  UsedBy
-------------------  -----------------  ----------------------------------------
cluster-1-detaches   false              dd_00, dd_01, dd_02, dd_03, dd_04,
                                        dd_05, dd_06, dd_07, dd_16, dd_17, ...
                                        (20 total)
cluster-2-detaches   false              dd_08, dd_09, dd_10, dd_11, dd_12,
                                        dd_13, dd_
```

3. Navigate to the context for the target device.

```
VPlexcli:/distributed-storage/rule-sets>cd
   /clusters/cluster-1/devices/base0

VPlexcli:/clusters/cluster-1/devices/base0>
```

4. Use the **set rule-set-name** command to attach a rule-set:

```
VPlexcli:/clusters/cluster-1/devices/base0> set rule-set-name
   cluster-2-detaches
```

5. Use the **ll** command to verify the change:

```
VPlexcli:/clusters/cluster-1/devices/base0> ll

Attributes:
Name                    Value
----------------------  ---------
application-consistent  false
.
.
.
rule-set-name           cluster-2-detaches
.
```

Verify a rule-set's configuration

Use the **ds rule-set what-if** command to describe if and when I/O is resumed at the individual clusters based on the current configuration of a given rule-set. The syntax for the command is:

ds rule-set what-if --islands "<list of cluster IDs in the island separated by commas>**" --rule-set** *rule-set-name*

Cluster IDs must be separated by commas, islands by a space.

For example:

```
VPlexcli:/distributed-storage/rule-sets> ds rule-set what-if --islands "cluster-1,cluster-2"
--rule-set cluster-2-detach

IO does not stop.
```

Detach a rule-set from a device

Use the **set** *device*::**rule-set-name ""** command to remove a rule-set from a device.

1. Navigate to the context for the target device:

```
VPlexcli:/distributed-storage/rule-sets> cd /distributed-storage/distributed-devices/dd_23

VPlexcli:/distributed-storage/distributed-devices/dd_23>
```

2. Use the **ll** command to display the rule-set attached to the device:

```
VPlexcli:/distributed-storage/distributed-devices/dd_23> ll

Attributes:
Name                    Value
----------------------  ---------
application-consistent  false
.
.
.
rule-set-name           cluster-2-detaches
```

3. Use the **set** *device*::**rule-set-name ""** command to remove the rule-set:

```
VPlexcli:/distributed-storage/distributed-devices/dd_23> set rule-set-name ""
```

A message and prompt appear:

```
Removing the rule-set from device 'dd_23' could result in data being unavailable during a WAN
link outage.  Do you wish to proceed ? (Yes/No)
```

4. Type **Yes**.
5. Use the **ll** command to display the change:

```
VPlexcli:/distributed-storage/distributed-devices/dd_23> ll

Attributes:
Name                    Value
----------------------  ----------------------
.
.
.
rule-set-name           -
.
.
.
```

Remove all rules from all distributed devices

WARNING

There is NO undo for this procedure.

From any context, use the **ds dd remove-all-rules** command to remove all rules from all distributed devices.

For example:

```
VPlexcli:/distributed-storage/distributed-devices/dd_23> remove-all-rules

All the rules in distributed-devices in the system will be removed. Continue? (Yes/No) yes
```

Copy a rule-set

Use the **ds rule-set copy** command to copy a rule-set. The syntax for the command is:

ds rule-set copy --source *source* **--destination** *destination*

For example:

```
VPlexcli:/distributed-storage/rule-sets> ll
Name                 PotentialConflict  UsedBy
-------------------  -----------------  ---------------------------------------
TestRuleSet          false
.
.
.
VPlexcli:/distributed-storage/rule-sets> rule-set copy --source TestRuleSet --destination
CopyOfTest
VPlexcli:/distributed-storage/rule-sets> ll
Name                 PotentialConflict  UsedBy
-------------------  -----------------  ---------------------------------------
CopyOfTest           false
TestRuleSet          false
.
.
```

Delete a rule-set

Use the **ds dd destroy** command to delete a specified rule-set:

1. Navigate to the rule-set context:

```
VPlexcli:/> cd distributed-storage/rule-sets/
```

2. Use the **ll** command to display the rule-sets:

```
VPlexcli:/distributed-storage/rule-sets> ll
Name                 PotentialConflict  UsedBy
-------------------  -----------------  ---------------------------------------
CopyOfTest           false
TestRuleSet          false
.
.
.
```

3. Use the **destroy** ***rule-set-name*** command to delete the specified rule-set:

```
VPlexcli:/distributed-storage/rule-sets> destroy CopyOfTest
```

4. Use the **ll** command to verify the deletion:

```
VPlexcli:/distributed-storage/rule-sets> ll
Name                 PotentialConflict  UsedBy
-------------------  -----------------  ---------------------------------------
TestRuleSet          false
```

Configure distributed devices

The general procedure to configure a distributed device, and a virtual volume on that device includes the following steps:

- "Create a logging volume" on page 67
- "Identify available storage"
- "Create extents from selected storage"
- "Create local devices"
- "Create a distributed device"
- "Attach a rule-set to a distributed device"
- "Create a virtual volume on a distributed device"

Identify available storage

To identify storage volumes on each cluster available to be combined into the distributed device, do the following:

1. Use the **ll -p **/storage-volumes** command to display the storage volumes on each cluster:

   Note: VPD83 IDs are truncated in the following example.

```
VPlexcli:/> ll -p **/storage-volumes

/clusters/cluster-1/storage-elements/storage-volumes:
Name                          VPD83 ID         Capacity  Use         Vendor  IO      Type          Thin
                                                                               Status  ------------  Rebuild
----------------------------- ---------------- --------  ----------  ------  ------  ------------  -------
Sym_20gb_vol_1                VPD83T3:nnnn      20G       used        EMC     alive   normal        false
Sym_20gb_vol_10               VPD83T3:nnnn      20G       used        EMC     alive   normal        false
Sym_20gb_vol_100              VPD83T3:nnnn      20G       used        EMC     alive   normal        false
Sym_20gb_vol_101              VPD83T3:nnnn      20G       used        EMC     alive   normal        false
.
.
.
/clusters/cluster-2/storage-elements/storage-volumes:
.
.
.
```

2. Identify claimed storage volumes to add to the distributed device.

Note: To prevent creating distributed devices with unusable "leftover" storage, total capacities of the selected storage at both clusters should be identical.

Create extents from selected storage

To create extents using the storage volumes identified at each cluster in Identify available storage, do the following:

1. From the storage-volumes context, use the **extent create** command to create the extents to be added to the distributed device.

   The syntax for the **extent create** command is:

```
extent create --storage-volumes <storage-volumes> --size <size>
   --block-offset <offset> --num-extents <integer>
```

--**storage-volumes** - List of claimed storage volumes to be added to the extent.

--**size** - Size of each extent. If no size is specified, the largest contiguous range of blocks is used to create the specified number of extents.

--block-offset - The block offset on the underlying storage volume. If no block offset is specified, it is set automatically.

--num-extents - Number of extents to create per specified storage-volume. If no number is specified, it creates one extent per storage-volume.

In the following example, two 16 GB storage volumes are used to create an extent on cluster-1:

```
VPlexcli:/clusters/cluster-1/storage-elements/storage-volumes> extent create
VPD83T3:6000097000019260185253303041414 3,VPD83T3:600009700001926 01852533030414230
```

2. Navigate to the storage-volumes context on the second cluster and repeat **step 1** to create one or more extents to be added to the distributed device:

```
VPlexcli:/clusters/cluster-1/storage-elements/storage-volumes> cd /clusters
/cluster-2/storage-elements/storage-volumes/

VPlexcli:/clusters/cluster-2/storage-elements/storage-volumes> extent create
VPD83T3:6000097000019260185253303042423 8,VPD83T3:600009700001926 01852533030424243
```

3. Navigate to the storage-volume context for each of the extented storage volumes.
4. Use the **ll** command to display the amount of free space and the largest free chunk size:

```
VPlexcli:/> cd clusters/cluster-1/storage-elements/storage-volumes/
VPD83T3:6000097000019260185253303041423 4
VPlexcli:/clusters/cluster-1/storage-elements/storage-volumes/VPD83T3:600009700001926018525330
030414234> ll
Name                     Value
-----------------------  ------------------------------------------------------------
application-consistent   false
block-count              4195200
block-size               4K
capacity                 16G
description              -
free-chunks              []
health-indications       []
health-state             ok
io-status                alive
itls                     0x5000144240014720/0x50000972081cf15d/64,
                         0x5000144240014720/0x50000972081cf165/64,
                         0x5000144240014730/0x50000972081cf165/64,
                         0x5000144240014722/0x50000972081cf158/64,
                         0x5000144240014732/0x50000972081cf158/64,
                         0x5000144240014730/0x50000972081cf15d/64,
                         0x5000144240014732/0x50000972081cf160/64,
                         0x5000144240014722/0x50000972081cf160/64
largest-free-chunk       0B
locality                 -
operational-status       ok
storage-array-name       EMC-SYMMETRIX-192601707
storage-volumetype       normal
system-id                VPD83T3:6000097000019260185253303041423 4
thin-rebuild             false
total-free-space         0B
use                      used
used-by                  [extent_Symm1852_AB4_1]
vendor-specific-name     EMC
```

Create local devices

To create a local device from extents, do the following:

1. Use the **ll - p **/extents/** command to display the extents created on each cluster in "Create extents from selected storage" on page 78:

```
VPlexcli:/> ll -p **/extents/

/clusters/cluster-1/storage-elements/extents:
Name                    StorageVolume  Capacity  Use
----------------------  -------------  --------  -------
extent_CX4_logging_1    CX4_logging    80G       claimed
extent_CX4_lun0_1       CX4_lun0       1G        used
extent_CX4_lun0_2       CX4_lun0       1G        used
.
.
.
extent_Symm1852_21C_1   Symm1852_21C   16G       used
extent_Symm1852_AB4_1   Symm1852_AB4   16G       claimed
extent_Symm1852_AB8_1   Symm1852_AB8   16G       claimed

/clusters/cluster-2/storage-elements/extents:
Name                                  StorageVolume                Capacity  Use
------------------------------------  ---------------------------  --------  -------
extent_CX4_logging_2_1                CX4_logging_2                80G       claimed
extent_CX4_Test_Lun_5_1               CX4_Test_Lun_5               10G       claimed
extent_CX4_Test_Lun_6_1               CX4_Test_Lun_6               10G       claimed
extent_CX4_Test_Lun_9_1               CX4_Test_Lun_9               10G       claimed
extent_Cluster_2_VMware_Datastore_1 Cluster_2_VMware_Datastore  200G      claimed
.
.
.
```

2. Use the **local-device create** command to create a local device with the specified name.

   The syntax for the **local-device create** command is:

   local-device create --name *name* **--geometry** *geometry* **--extents** *extents* **--stripe-depth** *depth*

   --name - Name for the new device. Must be unique across all clusters. Devices on different clusters that have the same name cannot be combined into a distributed device.

   --geometry - Geometry for the new device. Valid values are "raid-0", "raid-1", or "raid-c".

   --extents - List of pathnames of claimed extents to be added to the device, separated by commas. Can also be other local devices (to create a device of devices).

   --stripe-depth - Required for devices with a geometry of raid-0. Specifies the stripe depth in 4 K byte blocks. The resulting stripe is sized using the following formula:

   <stripe-depth> * <the block size on the source storage extents>

   In the following example, the **ll** command displays the available (claimed) extents, and the **local-device create** command is used to create a 16 GB RAID 1 device on cluster-1:

```
VPlexcli:/clusters/cluster-1/storage-elements/extents> ll
Name                    StorageVolume  Capacity  Use
----------------------  -------------  --------  -------
```

```
.
.
.
extent_Symm1852_AAC_1  Symm1852_AAC   16G       claimed
extent_Symm1852_AB0_1  Symm1852_AB0   16G       claimed
extent_Symm1852_AB4_1  Symm1852_AB4   16G       claimed
extent_Symm1852_AB8_1  Symm1852_AB8   16G       claimed
```

```
VPlexcli:/clusters/cluster-1/storage-elements/extents> local-device create --name
TestDevCluster1 --geometry raid-1 --extents /clusters/cluster-1/
storage-elements/extents/extent_Symm1852_AAC_1,/clusters/cluster-1/storage-
elements/extents/extent_Symm1852_AB0_1
```

3. Navigate to the extents context on the second cluster:

```
VPlexcli:/clusters/cluster-1/storage-elements/extents> cd /clusters/cluster-2/
storage-elements/extents/
```

4. Use the **local-device create** command to create a local device with the same capacity.

   In the following example, the **ll** command displays the available (claimed) extents, and the **local-device create** command is used to create a 16 GB RAID 1 device on cluster-2:

```
VPlexcli:/clusters/cluster-2/storage-elements/extents> ll
Name                                 StorageVolume                Capacity  Use
-----------------------------------  ---------------------------  --------  -------
.
.
.
extent_Symm1852_BB8_1                Symm1852_BB8                 16G       claimed
extent_Symm1852_BBC_1                Symm1852_BBC                 16G       claimed
extent_base_volume_1                 base_volume                  2G        used
extent_base_volume_2                 base_volume                  2G        used
```

```
VPlexcli:/clusters/cluster-2/storage-elements/extents> local-device create --name
TestDevCluster2 --geometry raid-c --extents /clusters/cluster-2/
storage-elements/extents/extent_Symm1852_BB8_1,/clusters/cluster-2/storage-elements/extents/e
xtent_Symm1852_BBC_1
```

5. Return to the root context and use the **ll **/devices/** command to display the new devices on both clusters:

```
VPlexcli:/clusters/cluster-2/storage-elements/extents> cd
VPlexcli:/> ll -p **/devices

/clusters/cluster-1/devices:
Name            Operational  Health  Block    Block  Capacity  Geometry  Visibility  Transfer  Virtual
                Status       State   Count    Size   --------  --------  ----------  Size      Volume
--------------  -----------  ------  -------  -----  --------  --------  ----------  --------  ---------
TestDevCluster1 ok           ok      4195200  4K     16G       raid-1    local             2M        -
base0           ok           ok      262144   4K     1G        raid-0    local       -         base0_vol
base1           ok           ok      262144   4K     1G        raid-0    local       -         base1_vol
base2           ok           ok      262144   4K     1G        raid-0    local       -         base2_vol
base3           ok           ok      262144   4K     1G        raid-0    local       -         base3_vol

/clusters/cluster-2/devices:
Name            Operational  Health  Block    Block  Capacity  Geometry  Visibility  Transfer  Virtual
                Status       State   Count    Size   --------  --------  ----------  Size      Volume
--------------  -----------  ------  -------  -----  --------  --------  ----------  --------  ----------
TestDevCluster2 ok           ok      8390400  4K     32G       raid-c    local       -         -
```

```
base01          ok          ok      524288   4K     2G        raid-0    local       -         base01_vol
base02          ok          ok      524288   4K     2G        raid-0    local       -         base02_vol
base03          ok          ok      524288   4K     2G        raid-0    local       -         base03_vol
```

Create a distributed device

Local devices on each cluster constitute the two *legs* of the distributed device. The individual local devices may include any underlying type of storage volume or geometry (RAID 0, RAID 1, or RAID C), but they should be the same capacity.

If a distributed device is configured with local devices of different capacities:

- The resulting distributed device will be only as large as the smaller local device.
- The leftover capacity on the larger device will ***not*** be available.

To create a distributed device without wasting capacity, choose local devices on each cluster with the same capacity.

CAUTION

If there is pre-existing data on a storage-volume, and the storage-volume is not claimed as being application-specific, converting an existing local RAID device to a distributed RAID will NOT initiate a rebuild to copy the data to the other leg. Data will exist at only one cluster.

To prevent this, do one of the following:

- Claim the disk with data using the **application-consistent** flag, or
- Create a single-legged RAID 1 or RAID 0 and add a leg using the device attach-mirror command.

To create a distributed device from two local devices with the same capacity:

1. Use the **ll --p ** /devices** command to display the available (no virtual volume configured) local devices.

```
VPlexcli:/> ll -p **/devices

/clusters/cluster-1/devices:
Name             Operational  Health  Block    Block  Capacity  Geometry  Visibility  Transfer  Virtual
---------------  Status       State   Count    Size   --------  --------  ----------  Size      Volume
---------------  -----------  ------  -------  -----  --------  --------  ----------  --------  ---------
TestDevCluster1  ok           ok      4195200  4K     16G       raid-1    local       2M        -
base0            ok           ok      262144   4K     1G        raid-0    local       -         base0_vol
base1            ok           ok      262144   4K     1G        raid-0    local       -         base1_vol
base2            ok           ok      262144   4K     1G        raid-0    local       -         base2_vol
base3            ok           ok      262144   4K     1G        raid-0    local       -         base3_vol


/clusters/cluster-2/devices:
Name             Operational  Health  Block    Block  Capacity  Geometry  Visibility  Transfer  Virtual
---------------  Status       State   Count    Size   --------  --------  ----------  Size      Volume
---------------  -----------  ------  -------  -----  --------  --------  ----------  --------  ---------
TestDevCluster2  ok           ok      4195200  4K     16G       raid-1    local       2M        -
base01           ok           ok      524288   4K     2G        raid-0    local       -         base01_vol
base02           ok           ok      524288   4K     2G        raid-0    local       -         base02_vol
base03           ok           ok      524288   4K     2G        raid-0    local       -         base03_vol
```

2. Use the **ds dd create** command to create a distributed device.

   The syntax for the command is:

   ds dd create --name *name* **--devices** *devices* **--logging-volume** *logging-volumes* **--rule-set** *rule-set*

--name <name> - The name must be unique across the entire VPLEX configuration.

-- devices - List of pathnames to local devices to add to the distributed device. Select devices that have the same capacities. Separate entries in the list by commas.

--logging-volume - List of pathnames to one or more logging volumes to use with the new device.

If no logging volume is specified, a logging volume is automatically selected from any available logging volume that has sufficient space for the required entries. If no available logging volume exists, an error message is returned. See"Logging volumes" on page 66.

--rule-set - Attaches the specified rule-set to the device. If no rule-set is specified, the cluster that is local to the management server is assumed to be the winner in the event of an inter-cluster link outage. See "Manage rule-sets" on page 72.

In the following example, the **ds dd create** command creates a distributed device, and the default rule-set behavior is applied to the new device:

```
VPlexcli:/> ds dd create --name TestDevice --devices
/clusters/cluster-1/devices/TestDevCluster1,/clusters/cluster-2/devices/TestDevCluster2

Distributed-device 'TestDisDevice' is using rule-set 'cluster-2-detaches'.
```

3. Use the **ll **/***new device name* command to display the new distributed device:

```
VPlexcli:/> ll **/TestDisDevice

/distributed-storage/distributed-devices/TestDevice:

Attributes:
Name                    Value
----------------------  ----------------------
application-consistent  false
auto-resume             -
block-count             4195200
block-size              4K
capacity                16G
clusters-involved       [cluster-1, cluster-2]
geometry                raid-1
.
.
.
```

Create a virtual volume on a distributed device

The virtual volume is the logical storage construct presented to hosts. A virtual volume can be created with restricted access to specified users or applications, providing additional security.

To create a virtual volume on a distributed device, do the following;

1. Use the **ll -p **/distributed-devices** command to display a list of distributed devices on all clusters:

```
VPlexcli:/> ll -p **/distributed-devices

/distributed-storage/distributed-devices:
Name           Status   Operational  Health  Auto    Rule Set Name       Transfer
-------------  -------  Status       State   Resume  ------------------  Size
-------------  -------  -----------  ------  ------  ------------------  --------
-------------  -------  -----------  ------  ------  ------------------  --------
TestDisDevice  running  ok           ok      -       cluster-2-detaches  2M
dd_00          running  ok           ok      true    cluster-1-detaches  2M
dd_01          running  ok           ok      true    cluster-1-detaches  2M
dd_02          running  ok           ok      true    cluster-1-detaches  2M
.
.
.
```

2. Use the **virtual volume create** command to create a virtual volume on a specified distributed device.

   The syntax for the command is:

   virtual-volume create --device *device* **--set-tier** {1|2}

   --device - Pathname of the device on which to configure the virtual volume.

   --set-tier - Storage tier for the new virtual volume. This value is displayed to the host as part of the virtual volume's product ID.

   Use storage tiers to logically group storage. For example, specify the tier for Symmetrix systems as tier 1 storage, and for CLARiiON systems as tier 2.

   For example:

```
VPlexcli:/> virtual-volume create --device
   /distributed-storage/distributed-devices/TestDevice --set-tier 1
```

3. Navigate to the new virtual volume's context, and use the **ll** command to display its attributes:

```
VPlexcli:/> cd clusters/cluster-1/virtual-volumes/TestDevice_vol/

VPlexcli:/clusters/cluster-1/virtual-volumes/TestDevice_vol> ll
Name                  Value
--------------------  -----------------
block-count           4195200
block-size            4K
cache-mode            synchronous
capacity              16G
consistency-group     -
expandable            false
health-indications    []
health-state          ok
locality              distributed
operational-status    ok
scsi-release-delay    0
service-status        unexported
```

```
storage-tier          1
supporting-device     TestDevice
system-id             TestDevice_vol
volume-type           virtual-volume
```

Note: Virtual volume names are assigned automatically based on the device name and a sequential virtual volumes number.

Expose a virtual volume to hosts

A virtual volume can be exposed to one or more hosts.

To expose a virtual volume to a host:

1. Use the **ls** command in the storage-views context to display a list of all storage views:

```
VPlexcli:/clusters/cluster-1/exports/storage-views> ls
LicoJ009  Win206
```

2. Optionally, use the **ll **//***storage-view-name* command to display virtual volumes already in the target storage view:

```
VPlexcli:/> ll **//LicoJ010

/clusters/cluster-2/exports/storage-views/LicoJ010:
Name                       Value
-------------------------  ------------------------------------------------------------
controller-tag             -
initiators                 [LicoJ010_hba0, LicoJ010_hba1, LicoJ010_hba2, LicoJ010_hba3]
operational-status         ok
port-name-enabled-status   [P000000003CA000E6-A0-FC00,true,ok,
                           P000000003CA000E6-A1-FC00,true,ok,
                           P000000003CA001CB-A0-FC00,true,ok,
                           P000000003CA001CB-A1-FC00,true,ok,
                           P000000003CB000E6-B0-FC00,true,ok,
                           P000000003CB000E6-B1-FC00,true,ok,
                           P000000003CB001CB-B0-FC00,true,ok,
                           P000000003CB001CB-B1-FC00,true,ok]
ports                      [P000000003CA000E6-A0-FC00, P000000003CA000E6-A1-FC00,
                           P000000003CA001CB-A0-FC00, P000000003CA001CB-A1-FC00,
                           P000000003CB000E6-B0-FC00, P000000003CB000E6-B1-FC00,
                           P000000003CB001CB-B0-FC00, P000000003CB001CB-B1-FC00]
virtual-volumes            (0,base01_vol,VPD83T3:6000144000000010a000e68dc5f76188,2G),
                           (1,dd_00_vol,VPD83T3:6000144000000010a0014760d64cb21f,128G),
                           (2,dd_01_vol,VPD83T3:6000144000000010a0014760d64cb221,128G),
                           (3,dd_02_vol,VPD83T3:6000144000000010a0014760d64cb223,128G),
.
.
.                     ... (25 total)
```

3. Use the **export storage-view addvirtual volume** command to add the virtual volume to the storage view.

   The syntax for the command is:

   export storage-view addvirtualvolume --view *<storage-view>* **--virtual-volumes** *<virtual-volumes>* **--force**

   --view - Context path of the storage view to which to add the specified virtual volume.

   --virtual-volumes - List of virtual volumes or LUN-virtual-volume pairs. Mixing of virtual volumes and LUN-virtual-volume pairs is allowed. If only virtual volumes are specified, the LUN is automatically assigned. Entries must be separated by commas.

   --force - Required to expose a distributed device's volume to more than one host.

   For example:

```
VPlexcli:/> export storage-view addvirtualvolume --view LicoJ009 --virtual-volumes
TestDisDevice_vol/
```

   If the virtual volume has already been exposed to a host, a error message appears:

```
VPlexcli:/> export storage-view addvirtualvolume -view lsca5230view --virtual-volume
ExchangeDD_vol --force

WARNING: Volume 'ExchangeDD_vol' is already assigned to view 'lsca3207view'
```

4. Re-scan the disks on each host to ensure that each host can access the virtual volume.

Expose a virtual volume to a remote host

To enable a host to access a virtual volume on a remote cluster, expose the virtual volume on one cluster to a host on a different cluster.

To expose a virtual volume to a remote host:

1. Navigate to the device context of the target device:

```
VPlexcli:/> cd clusters/cluster-1/devices/base0
```

2. Use the **set visibility global** command to set the device's visibility to global:

```
VPlexcli:/clusters/cluster-1/devices/base0> set visibility global
```

3. Optionally, use the **ll** command to verify that change:

```
VPlexcli:/clusters/cluster-1/devices/base0> ll

Attributes:
Name                     Value
-----------------------  ---------
application-consistent   false
.
.
.
transfer-size            -
virtual-volume           base0_vol
visibility               global
.
.
.
```

4. Use the **export storage-view addvirtualvolume** command to expose the virtual volume to the remote host.

   For example:

```
VPlexcli:/clusters/cluster-1//devices/base0> export storage-view addvirtualvolume --view
E_209_view --virtual-volumes Symm1254_7BF_1_vol
```

5. Re-scan the disks on each host to ensure that each host can access the virtual volume.

Note: When a local-device is exported, it is automatically assigned the rule-set for its enclosing cluster.

Add a local mirror to distributed device

Attach a mirror to the local leg of a distributed device to increase data protection.

Use the **device attach-mirror** command to attach a local mirror to a distributed device.

In the following example, the distributed device TestDevDevice is composed of:

- TestDevCluster1 (device on cluster-1)
- TestDevCluster2 (device on cluster-2)

To add a local mirror to a component device of a distributed device:

1. Navigate to the cluster-*n* context on the cluster where the mirror will be added:

```
VPlexcli:/> cd clusters/cluster-1/
```

2. Use the **ll **/extents** command to display a list of extents:

```
VPlexcli:/clusters/cluster-1> ll **/extents

/clusters/cluster-1/storage-elements/extents:
Name                    StorageVolume  Capacity  Use
----------------------  -------------  --------  -------
extent_CX4_logging_1    CX4_logging    80G       claimed
extent_CX4_lun0_1       CX4_lun0       1G        used
extent_CX4_lun0_2       CX4_lun0       1G        used
extent_CX4_lun0_3       CX4_lun0       1G        used
extent_CX4_lun0_4       CX4_lun0       1G        used
extent_CX4_lun0_5       CX4_lun0       1G        claimed
extent_CX4_lun0_6       CX4_lun0       1G        claimed
```

3. Identify one or more claimed extents whose combined capacity matches the distributed device.
4. Use the **local-device create** command to create a device of the same capacity as the distributed device.

   For example:

```
VPlexcli:/clusters/cluster-1> local-device create --name TestDevCluster1mirror --geometry
raid-0 --extents extent_Symm1254_7BC_2, extent_Symm1254_7BE_2
```

5. Use the **device attach-mirror** command to attach the new device to the local (same cluster as the current context) leg of the distributed device. The syntax for the command is:

   device attach-mirror --device <*device*> **--mirror** <*mirror*> **--rule-set** <*rule-set*> **--force**

   -- device - Name or context path of the device to add the mirror to. The target device must not have a virtual volume configured. If the name of a device is used, verify that the same device name is not used by any local-device in a different cluster.

   --mirror - Name or context path of the device to add as a mirror. It must be a top-level device. Verify that the same device name is not used by any local-device in a different cluster.

   --force - Forces a rule-set with a potential conflict to be applied.

   --rule-set - The rule-set applied to the device.

   If the **--rule-set** option is omitted, a default rule-set is assigned as follows:

- If the parent device has a volume, the device inherits the rule-set of the parent.
- If the parent device does not have a volume, the cluster that is local to the management server is the winner.

Note: The VPLEX system displays a message indicating which rule-set has been assigned.

For example:

```
VPlexcli:/clusters/cluster-2> device attach-mirror --device
   /clusters/cluster-2/devices/TestDevCluster2 --mirror
   TestDevCluster2 Mirror
```

A rebuild is automatically started to synchronize the mirror.

Note: Use the **rebuild status** command to monitor the rebuild.

Remove a local mirror from a distributed device

To remove the local mirror from a distributed device, do the following:

Use the **device detach-mirror** command to remove a mirror from a local device. The syntax of the command is:

```
device detach-mirror
   [-d|--device] [context path|name]
   [-m|--mirror] [context path|name]
   [-s|--slot] slot number
   [-i|--discard]
   [-f|--force]
```

[-d | --device] ***context path*** or ***name*** - * Name or context path of the device from which to detach the mirror. Does not have to be a top-level device. If the device name is used, verify that the name is unique throughout the VPLEX, including local devices on other clusters.

[-m | --mirror] ***context path*** or ***name*** - * Name or context path of the mirror to detach. Does not have to be a top-level device. If the device name is used, verify that the name is unique throughout the VPLEX, including local devices on other clusters.

[-s | --slot] ***slot number*** - Optional argument. Slot number of the mirror to be discarded. Applicable only when the **--discard** argument is used.

[-i | --discard] - Optional argument. When specified, discards the mirror to be detached. The data is not discarded.

[-f | --force] -Force the mirror to be discarded. Must be used when the **--discard** argument is used.

For example:

```
VPlexcli:/clusters/cluster-2> device detach-mirror --device
/clusters/cluster-2/devices/TestDevCluster2 --mirror /clusters/cluster-2/devices/
TestDevCluster2Mirror
Detached mirror TestDevCluster2Mirror.
Mirror TestDevCluster2Mirror is below /clusters/cluster-2/devices.
```

The mirror is removed from the cluster, but the distributed device is left intact and functional.

Create a distributed device from an exported volume

The **device attach-mirror** command can be used to create a distributed device using:

- A local, exported volume as the source
- A device on the remote cluster as the mirror

Adding a mirror using this method expands the local device into a distributed device without impacting host I/O.

To add a remote mirror to an exported volume:

1. Use the **local-device create** command to create a local device on the remote cluster.

   For example:

```
VPlexcli:/clusters/cluster-2> local-device create --name RemoteMirrorCluster2 --geometry
raid-0 --extents extent_Symm1254_7BC_3, extent_Symm1254_7BE_3 --stripe-depth 1
```

2. Use the **device attach-mirror** command to attach the local device on the remote cluster to the leg on local cluster used as the basis for the distributed device:

```
VPlexcli:/clusters/cluster-1/devices> device attach-mirror --device
/clusters/cluster-1/devices/SourceVolumeCluster1 --mirror
/clusters/cluster-2/devices/RemoteMirrorCluster2
```

   The device on the remote cluster is added as a mirror to the exported volume on the local cluster.

   A rebuild is automatically started to synchronize the two devices.

Note: Use the **rebuild status** command to monitor the rebuild.

   While the rebuild is in progress, I/O on the volume being mirrored continues without host access being affected.

Display/enable/disable automatic device rebuilds

By default, automatic device rebuilds are enabled on all devices. For configurations with limited bandwidth between clusters, it may be useful to disable automatic rebuilds.

Use the **set** command to enable/disable automatic rebuilds on the distributed device. The rebuild setting is immediately applied to the device.

- Set rebuild-allowed to **true** to start or resume a rebuild if the mirror legs are out of sync.
- Set rebuild-allowed set to **false** to stop any rebuild in progress.

  When automatic rebuild is re-enabled on a device where it has been disabled, the rebuild starts again from the place where it stopped.

Display device's rebuild setting

To display the automatic rebuild setting on a device, navigate to the context of the target distributed device:

```
VPlexcli:/>cd distributed-storage/distributed-devices/TestDevDevice
```

Use the **ll** command to display detailed information about the device, including its rebuild-allowed setting:

```
VPlexcli:/distributed-storage/distributed-devices/TestDevDevice> ll

Attributes:
Name                    Value
----------------------  ----------------------
.
.
.
rebuild-allowed         true
```

Enable automatic rebuilds

To allow automatic rebuilds after a failed inter-cluster link has been restored:

1. Navigate to the to the context of the target distributed device:

```
VPlexcli:/cd distributed-storage/distributed-devices/TestDevDevice
```

2. Use the **set rebuild-allowed true** command to allow automatic rebuilds after a failed inter-cluster link has been restored:

```
VPlexcli:/distributed-storage/distributed-devices/TestDevDevice> set rebuild-allowed true
```

3. Use the **ll** command to display the change.

Disable automatic rebuilds

To prevent automatic rebuild after a failed inter-cluster link has been restored:

1. Navigate to the context of the target distributed-device:

```
VPlexcli:/cd distributed-storage/distributed-devices/TestDisDevice
```

2. Use the **set rebuild-allowed false** command to prevent automatic rebuilds after a failed inter-cluster link has been restored:

```
VPlexcli:/distributed-storage/distributed-devices/TestDevDevice> set rebuild-allowed false
```

3. Use the **ll** command to display the change.

Configure I/O resumption after a network outage

This section describes the procedures to

- "Enable/disable I/O auto-resume on a losing cluster" on page 95
- "Resume I/O on a losing cluster (auto-resume is false)" on page 95
- "Resume I/O on a cluster during link outage" on page 96
- "Resume I/O for a conflicting detach" on page 97

By default, the cluster local to the management server used to create a distributed device is the "winner" in the event of a network outage.

Rule-sets determine which cluster "wins" when a network outage occurs.

Use the **set auto-resume** command to determine what happens at the losing cluster after the link is restored.

Use the **device resume-link-up** command to manually resume I/O on the losing cluster when the auto-resume flag is false.

Use the **device resume-link-down** command to manually resume I/O on a suspended volume while the link is down.

Enable/disable I/O auto-resume on a losing cluster

Use the **set auto-resume true** command to configure a distributed device to automatically resume I/O on the ***losing*** cluster after the network outage is restored between clusters.

The syntax of the command is:

```
set auto-resume {true|false}
```

For example:

```
VPlexcli:/cd distributed-storage/distributed-devices/TestDevDevice
VPlexcli:/distributed-storage/distributed-devices/TestDevDevice> set auto-resume true
VPlexcli:/distributed-storage/distributed-devices/TestDevDevice> ll

Attributes:
Name                    Value
----------------------  ----------------------
application-consistent  false
auto-resume             true
.
.
.
```

Resume I/O on a losing cluster (auto-resume is false)

When a distributed device's **auto-resume** is set to false, use the **device resume-link-up** command to manually resume I/O to the mirror leg on the losing cluster after the link outage recovers.

Use the **device resume-link-up** command to resume I/O on:

- All suspended virtual volumes of the specified top-level distributed devices
- The given distributed virtual volumes,
- All the distributed virtual volumes in the system.

The syntax of the command is:

```
device resume-link-up --force {--virtual-volume
   <virtual-volume>|--devices <devices>|--all}
```

--virtual-volume - List of context paths to one or more virtual volumes, separated by commas. Resume I/O on only the specified virtual volume.

--devices - List of context paths to one or more top-level devices, separated by commas. Resume I/O on only the specified top level device.

--all - Resume I/O on all virtual volumes on the losing clusters.

--force - Forces I/O to resume.

For example:

```
VPlexcli:/distributed-storage/distributed-devices> device
   resume-link-up --devices DDte_4a --force
```

Resume I/O on a cluster *during* link outage

WARNING

The device resume-link-down command causes I/O to resume on the local cluster regardless of any rule-sets applied to the device. Verify that rules and any manual detaches do not result in conflicting detaches (cluster-1 detaching cluster-2, and cluster-2 detaching cluster-1). Conflicting detaches will result in lost data on the losing cluster, a full rebuild, and degraded access during the time of the full rebuild.

Use the **device resume-link-down** command to resume I/O to the mirror leg of distributed devices *during* a link outage.

CAUTION

Be careful not to introduce a conflicted detach by allowing both legs of distributed devices to independently resume I/O.

The syntax for the command is:

device resume-link-down {**--all-at-island** | **--cluster** <*context-path*> | **--devices** <*context-path,context-path*>} **--force**

--all-at-island - Resumes I/O on all devices on the chosen winning cluster *and* the clusters with which it is in communication.

--cluster - Context path to a cluster. Resumes I/O on the specified cluster and the clusters it is in communication with during a link outage. Necessary only when the **all-at-island** flag or distributed devices are specified. Not required for local devices with global visibility.

--devices - List of context paths to one or more top-level devices, separated by commas. Resumes I/O for the specified devices.

--force - Forces I/O to resume.

For example:

```
VPlexcli:/distributed-storage/distributed-devices> device
   resume-link-down --all-at-island --cluster --devices DD_5d --force
```

Resume I/O for a conflicting detach

When a conflicting detach occurs, use the **ds dd declare-winner** command to manually designate the winning cluster.

When the link is restored, the leg on losing cluster is rebuilt from the leg on the winning cluster.

The syntax for the command is:

```
ds dd declare-winner --distributed-device <distributed-device>
   --cluster <cluster> --force
```

--distributed-device - Specifies the distributed-device for which to declare a winning cluster.

--cluster - Specifies the winning cluster.

--force - Forces the 'declare-winner' command to be issued.

For example:

```
VPlexcli:/distributed-storage/distributed-devices> ds dd
   declare-winner --distributed-device DDtest_4 --cluster cluster-2
```

6

Volume expansion

This chapter describes the following topics:

Overview

A VPLEX virtual volume is created on a device or a distributed device, and is presented to a host through a storage view. For a number of reasons, you may want to expand the capacity of a virtual volume.

If the volume supports expansion, VPLEX detects the capacity gained by expansion. Then, you determine the available expansion method: either storage-volume (the preferred method) or concatenation (RAID-C expansion). VPLEX can also detect the available expansion method.

Not all virtual volumes can be expanded. See "Determine volume expansion-method" on page 101 for more details.

If the volume supports expansion, VPLEX detects the capacity gained by expansion. Then, you determine the available expansion method: either storage-volume (the preferred method) or concatenation (RAID-C expansion).

You perform volume expansion using a simple, non-disruptive procedure:

1. Expand the storage volume associated with the virtual volume on the underlying storage array.
2. Allow VPLEX to rediscover the underlying storage array.
3. Expand the virtual volume using the CLI or GUI.

Additional documentation

- *EMC VPLEX CLI Guide* - Run the **virtual-volume expand** command.
- Unisphere for VPLEX Online Help - Use the GUI to expand the virtual-volume.
- The generator - Expand a distributed virtual volume using GeoSynchrony. and configure storage arrays for VPLEX.

Determine volume expansion-method

VPLEX recommends the best expansion method based on the geometry of the underlying device, using the **expansion-method** attribute.

Possible values for the **expansion-method** attribute are:

- **storage-volume** - VPLEX expands the underlying storage-volume (the corresponding LUNs on the back-end array).
- **concatenation** (or RAID-C expansion) - The virtual volume is expanded by adding only specified extents or devices, as required.
- **not supported** - VPLEX cannot expand the virtual volume because the volume did not meet one or more prerequisites. See "Limitations" on page 106 for details.

List the **expansion-method** attribute using the CLI or GUI.

List expansion-method attribute using CLI

In this example, the **expansion-method** attribute for **Test_volume** is displayed by listing the **virtual-volumes** context using the CLI.

```
VPlexcli:> ll /clusters/cluster-1/virtual-volumes/ Test_volume
Name                  Value
--------------------  --------------
.
.
.
capacity              0.5G
consistency-group     -
expandable            true
expandable-capacity   0.0G
expansion-method      storage-volume
expansion-status      -
```

Note the **expansion-method** attribute value **storage-volume**, indicating VPLEX will use the storage-volume method, by default, to expand this virtual volume.

List expansion-method attribute using GUI

When using the GUI, as shown in Figure 2, you display the properties of the virtual volume you want to expand by clicking on the virtual volume name.

In the example below, the properties for device_BASIC_vnx-1912_LUN146_1_vol indicate the recommended expansion method is **storage-volume**. VPLEX will use the **storage-volume** method, by default, to expand this virtual volume.

For more information on using the GUI, see the Help available on the VPLEX Management Server.

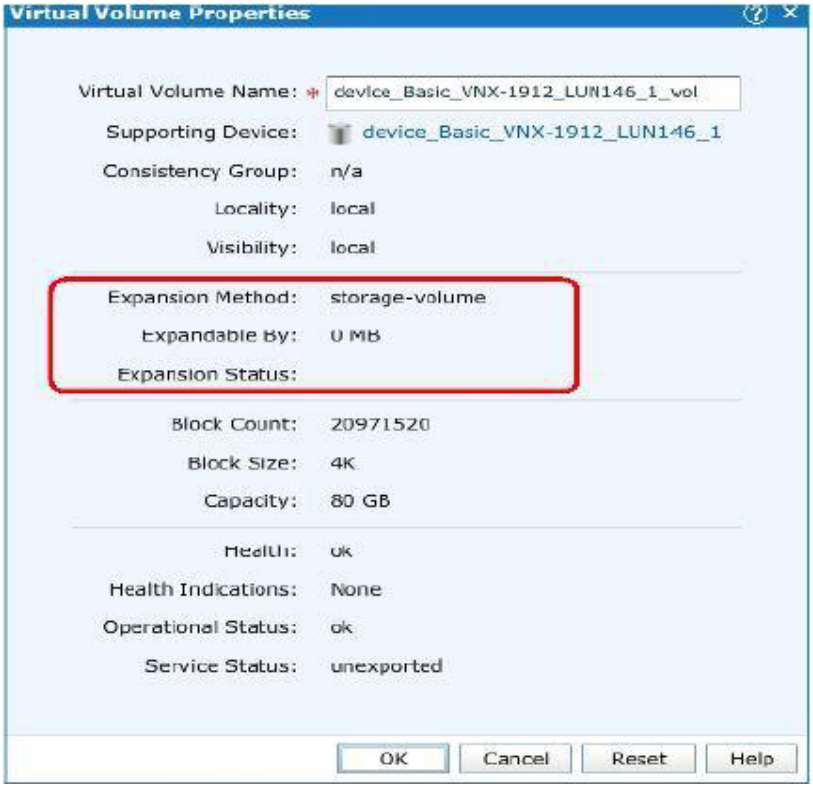


Figure 2 **Virtual volume expansion properties**

For information about other attributes and how to use them as you expand your virtual volume, see "Expand the virtual volume" on page 103.

Expand the virtual volume

Storage-volume expansion method

Use the following guidelines when expanding the virtual volume using the storage-volume method.

Overview

The storage volume method of expansion supports simple, fast expansion on a variety of device geometries. Three of the most common are described here.

1:1 virtual volume to storage volume

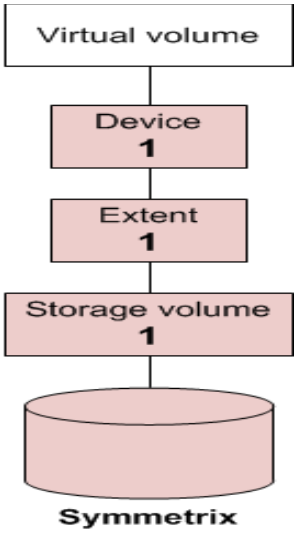


Figure 3 Common geometries: 1:1 virtual volume to storage volume

In the 1:1 virtual volume to storage volume geometry, the virtual volume is built on a single extent. The extent is built on a single storage volume.

Dual-legged RAID-1

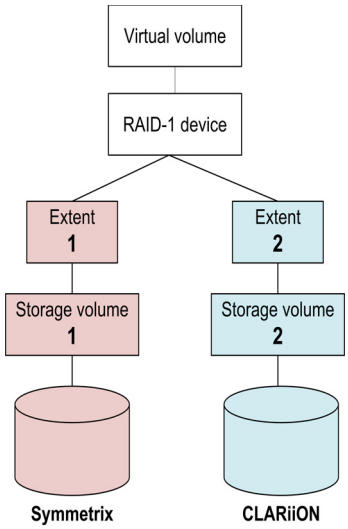


Figure 4 Common geometries: dual-legged RAID-1

This geometry is similar to the dual-legged RAID-1, but uses a distributed RAID-1 device (DR1) versus a RAID-1 device. DR1 devices have physical volumes at both clusters in a VPLEX Metro or VPLEX Geo configuration for simultaneous active/active read/write access using AccessAnywhere™.

Distributed RAID-1

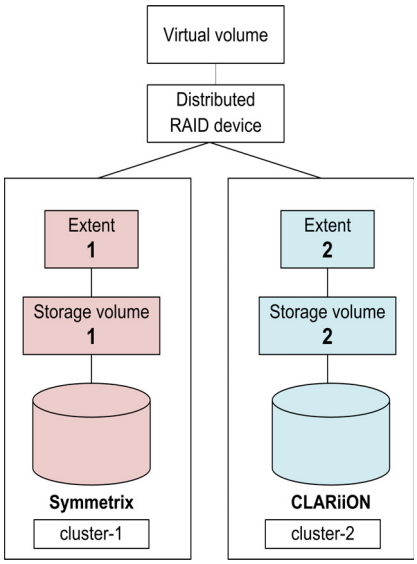


Figure 5 **Common geometries: distributed RAID-1**

Storage-volume expansion method prerequisites

In order to expand a device or add a target for expansion using the storage-volume expansion method, the VPLEX virtual volume geometry must meet one of the following criteria:

- The virtual volume is mapped 1:1 to the underlying storage volume.
- The virtual volume is a multi-legged RAID-1 or RAID-0 volume, and each of its smallest extents is mapped 1:1to a back end storage volume.
- The virtual volume is a RAID-C (expansion through the last in-use extent or any following extent, and only through extents 1:1 mapped to storage volumes).
- Volume geometry is a combination of any of the geometries listed previously.

Note: Storage volumes that are not mapped 1:1 onto extents cannot have the virtual volumes built on them expanded using this command. To expand a virtual volume whose underlying storage-volume is not mapped 1:1 onto extents, perform an extent migration to an extent that is 1:1 mapped to a storage volume. Alternatively, migrate to a larger device and use the **virtual-volume expand** CLI command to expand the volume to the extra available capacity.

Virtual volumes in RecoverPoint consistency groups

If a virtual volume is a member of a VPLEX RecoverPoint consistency group, then you must perform additional steps in order to use the storage volume expansion method:

- Remove the volume from RecoverPoint use.

  Refer to the "Remove a VPLEX Virtual Volume from use in RecoverPoint" procedure in the generator.

- Expand the virtual volume using the storage volume expansion method.

- Add the volume back to RecoverPoint use.

  Refer to this procedure in the generator.

For prerequisites specific to expansion using the storage-volume technique, see "Storage-volume expansion method prerequisites" on page 104.

Plan for volume expansion

List the **expandable-capacity** attribute (in the CLI) or the **Expandable By** field (in the GUI) to plan capacity of your back-end storage devices.

- **expandable-capacity**/**Expandable By**: - For virtual volumes that can be expanded using the storage-volume method of expansion, this value is the capacity added to the back-end storage-volume, but not yet exposed to the host by the virtual volume.

  This capacity is available for expanding the VPLEX virtual volume using the **storage-volume** method of expansion.

  - **0** (zero) **-** A value of Zero indicates that there is no **expandable-capacity** for the volume. Reference the expansion-method attribute to determine if storage-volume based expansion is supported.
  - Non-zero value - A non-zero value indicates capacity available to expand the VPLEX virtual volume. Reference the expansion-method attribute to determine if storage-volume based expansion is supported.

Volume Expansion

Perform volume expansion using one of the following techniques:

- The **virtual-volume expand** CLI command. Refer to the *EMC VPLEX CLI Guide* for detailed information about this command.
- Expand a virtual volume using the VPLEX GUI. Refer to the Unisphere for VPLEX Online Help for complete steps.
- Refer to the generator for procedures to expand a distributed virtual volume using GeoSynchrony:

During volume expansion, using the storage-volume method, keep the following in mind:

CAUTION

Performing a major host operation (such as a LIP reset, for example) in order to detect a change in volume size presents risk to volumes accessed by the host. It is best to avoid such resource intensive operations during volume expansion.

- Expansion initialization traffic occurs on disk areas not performing host I/O. In addition, the amount of time taken to initialize the newly added capacity depends on the performance of the array hosting the storage volumes. The expected performance is still faster, however, than that to rebuild a volume.
- Across distributed RAID-1 devices, the initialization process does not consume WAN data bandwidth as each cluster performs its initialization locally.
- On RAID-1 and distributed RAID-1 devices, VPLEX ensures that all RAID-1 legs have consistent information on the expanded space.
- The level of redundancy on RAID-1 and distributed RAID-1 device geometries is maintained through the expansion and initialization process.
- The newly expanded virtual volume capacity will be available to use by hosts when the initialization process has finished.

- If VPLEX has claimed the storage volumes as thinly provisioned, the initialization process will not affect the underlying provisioning of the additional capacity reported to VPLEX.

Check status of volume expansion

Query the status of your volume expansion by listing the value of the following attributes in the **virtual-volumes** context using the CLI or GUI.

- **expansion-status** - Status of virtual volume expansion. Indicates whether a virtual volume expansion is underway or has failed.

  The attribute will have one of the following values:

  - **in-progress** - Expansion is in progress.
  - **failed** - Most recent **in-progress** expansion has failed and expansion needs to be retried by the user. If expansion is not retried, then this state persists for up to two days. If two days pass with no fix, the **failed** state clears and the volume will be assumed fixed.
  - **unknown** - Status was unable to be determined. This may be because of a communication error or because of an internal programming error.
  - **-** (dash character) - None of the above states apply.
- **expansion-summary -** If there are no **in-progess** or **failed** expansions, and no virtual volumes with a non-zero e**xpandable-capacity**, then the virtual volume summary command displays `No expansion activity` in **expansion-summary.**

Limitations

The following are general limitations for expanding virtual volumes.

Some virtual volumes cannot be expanded under specific circumstances or at all. Volumes can not be expanded if any of the following conditions are true:

- **Migration or rebuilding is occurring** - Expansion is blocked during migration or rebuilds.
  - If you are rebuilding volumes, wait until the rebuild is complete before attempting expansion.
  - If you are migrating data, wait until the migration is complete. Alternatively, cancel or commit the migration, and then perform the expansion.
- **Upgrade is occurring** - Volume expansion is blocked during Non-Disruptive Upgrade (NDU).
- ***health-check* command reports problems** - The health-check command returns problems relating to the cluster, storage volumes, or virtual volume being expanded.
- **Volume to expand belongs to a RecoverPoint-enabled consistency group** - A volume cannot be expanded if it belongs to a RecoverPoint-enabled consistency group. See "Virtual volumes in RecoverPoint consistency groups" on page 104 for additional steps you need to take before you can expand a volume that is a member of a RecoverPoint-enabled consistency group.
- **Volume is a metadata volume** - Metadata volumes cannot be expanded.

Limitations with storage-volume expansion

The following limitations apply to the storage volume expansion method:

- For virtual volumes built on RAID-1 or distributed RAID-1 devices, a maximum of 1000 initialization processes can run concurrently per cluster. If this limit is reached on a cluster, then no new expansions can be started on virtual volumes with these geometries until some of the previously started initialization processes finish on that cluster.

Virtual volumes not containing RAID-1 or distributed RAID-1 devices are not affected by this limitation.

Troubleshooting and health-indications

When a volume expansion fails, information as to why it failed is added to the *health-indications* attribute. Note that when an expansion fails it does **not** degrade overall health, operational-status, or service-status of a virtual volume.

Note: The VPLEX Troubleshooting section of the SolVe Desktop contains procedures for recovering from an error with volume expansions.

Rediscover the array

You may need to rediscover the array after expansion. Depending on the type and configuration of the back-end array, the storage array may not support auto-discovery by VPLEX.

Best practice

If VPLEX does not automatically detect the change in the storage volume, use the **array-rediscover** command to force VPLEX to recognize the back-end expansion.

If you are performing multiple storage-volume expansions on the array, complete all storage-volume expansions, and re-discover the array only once to force VPLEX to detect all the expansions.

Some arrays need specific system settings to enable support of auto-discovery.

Refer to the generator for the procedures to configure storage arrays for VPLEX.

Note: Review the applicable best practices for host and array connectivity and configuration in the generator. Some arrays require specific flag settings for automatic detection.

CAUTION

Array re-discoveries may consume excessive resources and can be disruptive to I/O. Re-discover arrays only when necessary.

Concatenation expansion method

Some devices do not support the storage-volume method of expansion. In this case, use the concatenation method.

Best Practice

Before selecting extents and local devices to expand a virtual volume using concatenation, ensure the following:

- For volumes created on RAID-0 and RAID-C devices:
  - Select only one extent or local device. This extent or local device can be the same storage volume from which the virtual volume is created, or it can be a different storage volume.
- For volumes created on RAID-1 devices:
  - Select two or more extents or local devices.
  - Select a number of extents or local devices greater than or equal to the number of components (legs) in the RAID-1 device.
  - Create the same redundancy as the original RAID-1 device on which the virtual volume is created.
  - Select extents or local devices from different arrays to avoid a single point of failure.

7

Data migration

This chapter describes the following topics:

About data migrations

There are two types of data migrations:

- **One time migrations** - Begin an extent or device migration immediately when the **dm migration start** command is used.
- **Batch migrations** - Are run as batch jobs using re-usable migration plan files. Multiple device or extent migrations can be executed using a single command.

One time migrations

One time migrations include:

- **Extent migrations** - Extent migrations move data between extents in the same cluster. Use extent migrations to:
  - Move extents from a "hot" storage volume shared by other busy extents
  - Defragment a storage volume to create more contiguous free space
  - Perform migrations where the source and target have the same number of volumes with identical capacities
- **Device migrations** - Devices are RAID 0, RAID 1, or RAID C devices built on extents or on other devices.

  Device migrations move data between devices on the same cluster or between devices on different clusters. Use device migrations to:
  - Migrate data between dissimilar arrays
  - Relocate a "hot" volume to a faster array
  - Relocate devices to new arrays in a different cluster

Limitations

- Device migrations between distributed devices are not supported.
- Devices must be removed from consistency groups before they can be migrated.

Batch migrations

Batch migrations migrate multiple extents or devices. Create batch migrations to automate routine tasks:

- Use batched *extent* migrations to migrate arrays within the same cluster where the source and destination have the same number of LUNs and identical capacities.
- Use batched *device* migrations to migrate to dissimilar arrays (user must configure the destination's capacities to match the capacity and tier of the source array), and to migrate devices between clusters in a VPLEX Metro or VPLEX Geo.

Up to 25 local and 25 distributed migrations can be in progress at the same time. Any migrations beyond those limits are queued until an existing migration completes.

Limitations

- Devices must be removed from consistency groups before they can be migrated.

General procedure to perform data migration

Use the following general steps to perform extent and device migrations:

1. **Create and check a migration plan** (batch migrations only).
2. **Start** the migration.
3. **Monitor** the migration's progress.

4. **Pause, resume, or cancel** the migration (optional).
5. **Commit** the migration. Commit transfers the source virtual volume/device/extent to the target.

   If the virtual volume on top of a device has a system-assigned default name, committing a device migration renames the virtual volume after the target device.
6. **Clean up** (optional).

   For extent migrations: dismantle the source devices or destroy the source extent and unclaim its storage-volume.
7. **Remove** the record of the migration.

Prerequisites for target devices/extents

The target device or extent must:

- Be the same size or larger than the source device or extent

  If the target is larger in size than the source, the extra space cannot be utilized.

  For example, if the source is 200 GB, and the target is 500 GB, only 200 GB of the target can be used after a migration. The remaining 300 GB cannot be claimed.
- Not have any existing volumes on it.

WARNING

Device migrations are not recommended between clusters. All device migrations are synchronous. If there is I/O to the devices being migrated, and latency to the target cluster is equal to or greater than 5ms, significant performance degradation may occur.

About rebuilds

Rebuilds synchronize data from a source drive to a target drive. When differences arise between legs of a RAID, a rebuild updates the out-of-date leg.

There are two types of rebuild behavior:

- A **full** rebuild copies the entire contents of the source to the target.
- A **logging** rebuild copies only changed blocks from the source to the target.

Local mirrors are updated using a full rebuild (local devices do not use logging volumes).

In VPLEX Metro and Geo configurations, all distributed devices have an associated logging volume. Logging volumes keep track of blocks written during an inter-cluster link outage. After a link or leg is restored, the VPLEX system uses the information in logging volumes to synchronize mirrors by sending only changed blocks across the link.

Logging volume rebuilds also occur when a leg of a disaster recovery RAID 1 (DR1) becomes unreachable, but recovers quickly.

If a logging volume is unavailable at the time that a leg is scheduled to be marked out-of-date (via the log), the leg is marked as fully out-of-date, causing a full rebuild.

The unavailability of a logging volume matters both at the time of recovery (when the system reads the logging volume) and at the time that a write failed on one leg and succeeded on another (when the system begins writes to the logging volume).

CAUTION

If no logging volume is available, an inter-cluster link restoration will cause a full rebuild of every distributed device to which there were writes while the link was down.

See "Logging volumes" on page 66.

Rebuilds for thin provisioned storage

Thin provisoning allows storage to migrate onto a thinly provisioned storage volumes while allocating the minimal amount of thin storage pool capacity.

Thinly provisioned storage volumes can be incorporated into RAID 1 mirrors with similar consumption of thin storage pool capacity.

VPLEX preserves the unallocated thin pool space of the target storage volume by detecting zeroed data content before writing, and suppressing the write for cases where it would cause an unnecessary allocation. VPLEX requires the user to specify thin provisoning for each back-end storage volume. If a storage volume is thinly provisioned, the "thin-rebuild" attribute must be to "true" either during or after claiming.

CAUTION

If a thinly provisioned storage volume contains non-zero data before being connected to VPLEX, the performance of the migration or initial RAID 1 rebuild is adversely affected. System volumes are supported on thinly provisioned LUNs, but these volumes must have thin storage pool resources available, at maximum capacity. System volumes must not compete for this space with user-data volumes

in the same pool. If the thin storage allocation pool runs out of space and this is the last redundant leg of the RAID 1, further writing to a thinly provisioned device causes the volume to lose access to the device, resulting in data unavailability.

Performance considerations

To improve overall VPLEX performance, disable automatic rebuilds or modify the rebuild transfer size:

- Disable automatic rebuilds to avoid a flood of activity when re-attaching two clusters. See "Display/enable/disable automatic device rebuilds" on page 93

CAUTION

Disabling automatic rebuilds prevents DR1s from synchronizing. Child devices will be out of date, increasing the likelihood of remote reads.

- Modify the rebuild transfer size. See "About transfer-size" on page 121.

One-time data migrations

A one-time data migration moves data between the specified source and targets as soon as the **dm start migration** command is used. No reusable migration plan file is created as in "Batch migrations" on page 119.

Start a one-time device or extent data migration

To start a one-time device or extent migration:

1. Use the **drill down** command to display the components of the source, (including extents) of a view, virtual volume or device, down to the storage-volume level:

```
VPlexcli:/clusters/cluster-1> drill-down -o
virtual-volumes/Symm1254_7B7_1_vol

virtual-volume: Symm1254_7B7_1_vol (cluster-1)
   local-device: Symm1254_7B7_1 (cluster-1)
      extent: extent_Symm1254_7B7_1
         storage-volume: Symm1254_7B7
```

2. Identify the extent or device used by the source storage volume.
3. Use the **ll /clusters/*cluster*/storage-elements/extents** or **ll /clusters/*cluster*/devices** command to display available extents or devices.
4. Identify an unused extent or device as the destination.
5. Navigate to the appropriate migration context.

   For device migrations, navigate to device migration context:

```
VPlexcli:/> cd data-migrations/device-migrations
```

   For extent migrations, navigate to extent migration context:

```
VPlexcli:/> cd data-migrations/extent-migrations
```

6. Use the **dm migration start** command to start a migration. The syntax for the command is:

   dm migration start --name *name of migration*
   --transfer-size [*40 KB-128 MB*]
   --from s*ource extent or device*
   --to *destination extent or device*

   - The --**name** argument is used to track the migration's progress, and to manage (cancel, commit, pause, resume) the migration.
   - Use the optional **--transfer-size** argument to specify the size of the area set aside in cache to service the migration. A bigger transfer size means smaller space available for host I/O.

     Range: 40 K - 128 M. Default: 128 K. Must be a multiple of 4K.
   - Specify the **--from** device or extent by name if that name is unique in the global namespace. Otherwise, specify a full pathname.
   - Specify the **--to** device or extent by name if that name is unique in the global namespace. Otherwise, specify a full pathname.

   For example:

```
VPlexcli:/data-migrations/device-migrations> dm migration start --name migrate_012 --from
device_012 --to device_012a --transfer-size 12M
```

CAUTION

Setting too large a transfer size may result in data unavailability. Only vary from the default when performance implications are fully understood.

CAUTION

If host I/O activity is high, setting a large transfer size may impact host I/O.

See "About transfer-size" on page 121.

Monitor a migration's progress

Use the **ls** command to display the migration's status.

```
VPlexcli:/> ls data-migrations/device-migrations/ migrate_012
Name              Value
----------------  ----------------------------
from-cluster      cluster-1
percentage-done   10
source            device_012
source-exported   false
start-time        Fri May 28 13:32:23 MDT 2010
status            in progress
target            device_012a
target-exported   false
to-cluster        cluster-2
transfer-size     12M
type              full
```

Table 6 Migration status (1 of 2)

Field	Description
from-cluster	Cluster ID of the source extent, device, or devices in consistency group.
percentage-done	Percentage completion of the migration. 100 if migration is complete or committed.
source	Source extent or device.
source-exported	Whether the source device was exported during the migration. Applicable if the migration is an inter-cluster device migration and the device was not already exported. Devices are exported to a remote cluster in order to be visible at that cluster and can be used as a leg in a temporary distributed RAID 1 during the migration. **false** - Source device was not exported. **true -** Source device was exported.
start-time	Date and time migration was started.
status	Status of the migration. **committed** - Migration is committed. **complete** - Migration is complete (but not committed). **error** - Error condition, including source or target unreachable. **in-progress** - Migration is in progress. **partially-cancelled** - Attempt to cancel the migration failed. Retry the cancel. **partially-committed** - Attempt to commit the migration failed. Retry the commit. **paused** - The migration is paused. **queued** - The migration is in the queue.
target	Destination extent or device.

Table 6 Migration status (2 of 2)

Field	Description
target-exported	Whether the target device was exported during the migration. **false** - Target device was not exported. **true** - Target device was exported.
to-cluster	Cluster ID of the destination extent or device.
transfer-size	Size of the region in cache used to service the migration. 40 KB-128 MB.
type	Type of rebuild. **full** - Copies the entire contents of the source to the target. **logging** - Copies only changed blocks from the source to the target.

Pause/resume a migration (optional)

Active migrations (a migration that has been started) can be paused and then resumed at a later time.

Pause an active migration to release bandwidth for host I/O during periods of peak traffic.

Use the **dm migration pause --migrations** ***migration-name*** command to pause a migration.

Specify the ***migration-name*** by name if that name is unique in the global namespace. Otherwise, specify a full pathname.

For example:

- Pause a device migration:

```
VPlexcli:/data-migrations/device-migrations> dm migration pause --migrations migrate_012
```

Use the **dm migration resume --migrations** ***migration-name*** command to resume a paused migration.

Specify the ***migration-name*** by name if that name is unique in the global namespace. Otherwise, specify a full pathname.

For example:

- Resume a paused device migration:

```
VPlexcli:/data-migrations/device-migrations> dm migration resume --migrations migrate_012
```

Cancel a migration (optional)

Migrations can be canceled in the following circumstances:

- The migration is in progress or paused. The migration is stopped, and any resources it was using are freed.
- The migration has not been committed. The source and target devices or extents are returned to their pre-migration state.

Use the **dm migration cancel --force --migrations** ***migration-name*** command to cancel a migration.

Specify the ***migration-name*** by name if that name is unique in the global namespace. Otherwise, specify a full pathname.

For example:

```
VPlexcli:/data-migrations/device-migrations> dm migration cancel --force --migrations
migrate_012
```

Commit a completed migration

The migration process inserts a temporary RAID 1 structure above the source device/extent with the target device/extent as an out-of-date leg of the RAID 1. The migration can be understood as the synchronization of the out-of-date leg (the target).

After the migration is complete, the commit step detaches the source leg of the RAID 1, and removes the RAID 1.

The virtual volume, device, or extent is identical to the one before the migration except that the source device/extent is replaced with the target device/extent.

A migration must be committed in order to be cleaned.

CAUTION

Verify that the migration has completed successfully before committing the migration.

Use the **dm migrations commit --force --migrations** ***migration-name*** command to commit a migration.

Note: You must use the **--force** flag to commit a migration.

For example:

- Commit a device migration:

```
VPlexcli:/data-migrations/device-migrations> dm migration commit --force --migrations
migrate_012

Committed 1 data migration(s) out of 1 requested migration(s).
```

Clean a migration

Device migrations

For device migrations, cleaning dismantles the source device down to its storage volumes. The storage volumes no longer in use are unclaimed.

For device migrations only, use the **--rename-target** argument to rename the target device after the source device. If the target device is renamed, the virtual volume on top of it is also renamed if the virtual volume has a system-assigned default name.

Without renaming, the target devices retain their target names, which can make the relationship between volume and device less evident.

Extent migrations

For extent migrations, cleaning destroys the source extent and unclaims the underlying storage-volume if there are no extents on it.

Use the **dm migration clean --force --migrations** ***migration-name*** command to clean a migration.

Specify the ***migration-name*** by name if that name is unique in the global namespace. Otherwise, specify a full pathname.

For example:

```
VPlexcli:/data-migrations/device-migrations> dm migration clean --force --migrations
migrate_012

Cleaned 1 data migration(s) out of 1 requested migration(s).
```

Remove migration records

Note: Migrations must be canceled or committed before they can be removed.

Use the **dm migration remove --force --migrations** ***migration-name*** command to remove the records of the migration.

Specify the ***migration-name*** by name if that name is unique in the global namespace. Otherwise, specify a full pathname.

For example:

```
VPlexcli:/data-migrations/device-migrations> dm migration remove --force --migrations
migrate_012

Removed 1 data migration(s) out of 1 requested migration(s).
```

Batch migrations

Batch migrations are run as batch jobs from reusable batch migration plan files. Migration plan files are created using the **create-plan command**.

A single batch migration plan can be either for devices or extents, but not both.

Note: Migrations consume cache resources. Running multiple migrations concurrently may impact host I/O.

Use batch migrations to:

- Retire storage arrays (off-lease arrays) and bring new ones online
- Migrate devices to a different class of storage array

The steps to perform a batch migration are generally the same as those described in the "General procedure to perform data migration" on page 110.

There are two additional steps to prepare for a batch migration:

- Create a batch migration plan file (using the **batch-migrate create-plan** command)
- Test the batch migration plan file (using the **batch-migrate check-plan** command)

Prerequisites

The following prerequisites are required for batch migrations:

- The source and targets are both devices *or* extents. Migrations between devices and extents are not supported.
- Extents must be claimed (extent migrations) or local devices must be configured (device migrations) on the target array.
- The structure of the target is the same as the structure of the source.
- For extent migrations, both source and target extents must be in the same cluster.

Create a batch migration plan

The **batch-migrate create-plan** command creates a migration plan using the specified sources and targets. The syntax for the command is:

batch-migrate create-plan --file ***migration-filename*** **--sources** ***sources*** **--targets** ***targets*** **--force**

- **--file -** Specify the ***migration-filename*** filename only if that name is unique in the global namespace. Otherwise, specify a full pathname.
- **--sources -** Pathname(s) of the source extents or devices as a comma-separated list.

  Each element may contain wildcards.

  For extent migrations, source extents must be in the same cluster.
- **--targets -** Pathname(s) of the target extents or devices as a comma-separated list.

  For extent migrations, target extents must be in the same cluster as the source extents.
- **--force** - If a plan file with the same name already exists, forces the old plan to be overwritten.

In the following example, the **batch--migrate create-plan** command creates a batch migration named 'MigDev-test.txt' to:

- Migrate two devices at cluster-1 to two devices at cluster-2
- Overwrite an existing plan with the same name:

```
VPlexcli:/> batch-migrate create-plan --file MigDev-test.txt --sources
/clusters/cluster-1/devices/base0,/clusters/cluster-1/devices/base1 --targets
/clusters/cluster-2/devices/dev1723_618, /clusters/cluster-2/devices/dev1723_61C --force
Extents matching source pattern: base0, base1
Extents matching target pattern: dev1723_61C, dev1723_618

Creating file /var/log/VPlex/cli/MigDev-test.txt as migration plan file.

Wrote file /var/log/VPlex/cli/MigDev-test.txt. Please review and edit this file, and run this
command in the check-plan phase afterward.
```

In the following example, the **batch-migrate create-plan** command creates a batch migration to migrate all devices at cluster-1 to cluster-2:

```
VPlexcli:/> batch-migrate create-plan migrate.txt --sources /clusters/cluster-1/devices/*
--targets /clusters/cluster-2/devices/*
```

Check a batch migration plan

The **batch-migrate check-plan --file** ***file-name*** command checks the specified batch migration plan for the following:

- Block-size of source and target extents is equal (4 K bytes)
- Capacity of target extent is equal or bigger than the source extent's capacity
- Device migrations:
  - Target device has no volumes on it
  - Source device has volumes on it
- Extent migrations:
  - Target extent is claimed and ready for use
  - Source extent is in use

If the migration plan contains errors, a description of the errors is displayed, and the plan check fails. For example:

```
VPlexcli:/> batch-migrate check-plan --file MigDev-test.txt
Checking migration plan file /var/log/VPlex/cli/MigDev-test.txt.

Target device '/clusters/cluster-2/devices/dev1723_61C' has a volume.
Target device '/clusters/cluster-2/devices/dev1723_618' has a volume.
Plan-check failed, 2 problems.
```

Use the steps described in "Modify a batch migration file" on page 121 to correct the plan.

Repeat the process of check and modify until the batch migration plan passes the plan check. For example:

```
VPlexcli:/> batch-migrate check-plan --file migrate.txt

Checking migration plan file /temp/migration_plans/migrate.txt.
Plan-check passed.
```

Modify a batch migration file

To modify a batch migration file, do one of the following:

- Use the **batch-migrate create-plan** command, specify the same filename, and use the **--force** option to overwrite the old plan with the new one.
- Exit to the management server, and navigate to /var/log/VPlex/cli/.

  Use a text editor (vi) to edit and save the file.

```
VPlexcli:/> exit
Connection closed by foreign host.
service@ManagementServer:~> cd /var/log/VPlex/cli/
service@ManagementServer:/var/log/VPlex/cli>
```

Note: To add comments to the migration plan file, add lines beginning with "/ ".

Start a batch migration

About transfer-size

Transfer-size is the size of the region in cache used to service the migration. The area is globally locked, read at the source, and written at the target.

Transfer-size can be as small 40K, as large as 128 MB, and must be a multiple of 4K. The default value is 128 K.

A larger transfer-size results in higher performance for the migration, but may negatively impact front-end I/O. This is especially true for VPLEX Metro migrations.

If transfer-size is set too large, the rebuild read I/O may overwhelm the source device affecting the host I/O.

A smaller transfer-size results in lower performance for the migration, but creates less impact on front-end I/O and response times for hosts.

Set a large transfer-size for migrations when the priority is data protection or migration performance.

Set a smaller transfer-size for migrations when the priority is front-end storage response time.

Factors to consider when specifying the transfer-size:

- For VPLEX Metro configurations with narrow inter-cluster bandwidth, set the transfer size lower so the migration does not impact inter-cluster I/O.
- The region specified by transfer-size is locked during migration. Host I/O to or from that region is held. Set a smaller transfer-size during periods of high host I/O.
- When a region of data is transferred, a broadcast is sent to the system. Smaller transfer-size mean more broadcasts, slowing the migration.

Use the **batch-migrate start --transfer-size [40K-128M] --file** ***filename*** command to start the specified batch migration:

For example:

```
VPlexcli:/> batch-migrate start --file migrate.txt --transfer-size 2M
   Started 4 of 4 migrations.
```

Pause/resume a batch migration (optional)

Active batch migrations (a migration that has been started) can be paused and resumed.

Pause an active batch migration to release bandwidth for host I/O during periods of peak traffic.

Resume the batch migration during periods of low I/O.

Use the **batch-migrate pause --file** ***filename*** command to pause the specified active migration. For example:

```
VPlexcli:/data-migrations/device-migrations> batch-migrate pause
   --file migrate.txt
```

Use the **batch-migrate resume --file** ***filename*** command to resume the specified paused migration. For example:

```
VPlexcli:/data-migrations/device-migrations> batch-migrate resume
   --file migrate.txt
```

Cancel a batch migration (optional)

Cancel an active batch migration to return the source volumes to their state prior to the start of the migration.

Use the **batch-migrate cancel --file** ***filename*** command to cancel the specified migration. For example:

```
VPlexcli:/data-migrations/device-migrations> batch-migrate cancel
   --file migrate.txt
```

Note: In order to re-run a canceled migration plan, the **batch-migrate remove** ***filename*** command must be used to remove the records of the migration. See "Remove batch migration records" on page 125.

Monitor a batch migration's progress

Use the **batch-migrate summary** ***filename*** **--verbose** command to monitor the progress of the specified batch migration:

```
VPlexcli:/data-migrations/device-migrations> batch-migrate summary
   filename --verbose
```

For example:

```
VPlexcli:/data-migrations/device-migrations> batch-migrate summary --file migrate.txt
--verbose
source-                  source-site
target                   target-cluster  migration-name status                          percentage-complete  eta.
-----------------        -----------     ----------------------   --------  ------------ -------------------  ---
R20061115_Symm2264_010   1               R20070107_Symm2A10_1B0   1          migrate.txt  100                  -
R20061115_Symm2264_011   1               R20070107_Symm2A10_1B1   1          migrate.txt  100                  -
R20061115_Symm2264_012   1               R20070107_Symm2A10_1B2   1          migrate.txt  100                  -
R20061115_Symm2264_0113  1               R20070107_Symm2A10_1B3   1          migrate.txt  27                   4.08min

Processed 4 migrations:
    committed:   0
    complete:    3
    in-progress: 1
    paused:      0
    error:       0
    cancelled:   0
```

```
no-record:    0
```

View a batch migration's status

Use the **batch-migrate summary** *filename* command to display the status of the specified batch migration.

For example:

```
VPlexcli:/> batch-migrate summary migrate.txt

Processed 10 migrations from batch migration BR0:
committed:    0
complete:     10
in-progress:  0
paused:       0
error:        0
cancelled:    0
no-record:    0
```

Table 7 Batch migration summary

Field	Description
Processed....	Of the number of source-target pairs specified in the batch migration plan, the number that have been processed.
committed	Of the number of source-target pairs that have been processed, the number that have been committed.
completed	Of the number of source-target pairs that have been processed, the number that are complete.
in-progress	Of the number of source-target pairs that have been processed, the number that are in progress.
paused	Of the number of source-target pairs that have been processed, the number that are paused.
error	Jobs that encountered errors during processing.
cancelled	Of the number of source-target pairs that have been processed, the number that have been cancelled.
no-record	Of the number of source-target pairs that have been processed, the number that have no record in the context tree.

Note: If more than 25 migrations are active at the same time, they are queued, their status is displayed as *in-progress*, and percentage-complete is displayed as "**?**".

Commit a batch migration

The migration process inserts a temporary RAID 1 structure above the source devices/extents with the target devices/extents as an out-of-date leg of the RAID 1. Migration can be understood as the synchronization of the out-of-date leg (the target).

After the migration is complete, the commit step detaches the source leg of the RAID 1 and then removes the RAID.

The virtual volume, device or extent is identical to the one before the migration except that the source device/extent is replaced with the target device/extent.

A migration must be committed in order to be cleaned.

When the batch migration is 100% complete, use the **batch-migrate commit** *filename* command to replicate the volumes on the target devices and remove the volumes from the source devices.

To commit a batch migration, do the following:

1. Use the **batch-migrate summary** command to verify that the migration has completed with no errors.
2. Use the **batch-migrate commit --file** *filename* command to commit the migration.

WARNING

Commit permanently removes the volumes from the source devices.

For example:

```
VPlexcli:/> batch-migrate commit --file migrate.txt
```

Clean a batch migration

Device batch migration

For device migrations, cleaning dismantles the source device down to its storage volumes. The storage volumes no longer in use are unclaimed.

For device migrations only, use the optional **--rename-target** argument to rename the target device after the source device. If the target device is renamed, the virtual volume on top of it is also renamed if the virtual volume has a system-assigned default name.

Without renaming, the target devices retain their target names, which can make the relationship between volume and device less evident.

Extent batch migration

For extent migrations, cleaning destroys the source extent and unclaims the underlying storage-volume if there are no extents on it.

Use the **batch-migrate clean --file** *filename* command to clean the specified batch migration.

CAUTION

This command must be run before the batch-migration has been removed. The command will not clean migrations that have no record in the VPlexcli context tree.

In the following example, source devices are torn down to their storage volumes and the target devices and volumes are renamed after the source device names

```
VPlexcli:/> batch-migrate clean --rename-targets --file migrate.txt

Using migration plan file /temp/migration_plans/migrate.txt for
   cleanup phase.

0: Deleted source extent
   /clusters/cluster-1/devices/R20061115_Symm2264_010, unclaimed its
   disks Symm2264_010

1: Deleted source extent
   /clusters/cluster-1/extents/R20061115_Symm2264_011, unclaimed its
   disks Symm2264_011
```

```
2: Deleted source extent
   /clusters/cluster-1/extents/R20061115_Symm2264_012, unclaimed its
   disks Symm2264_012

3: Deleted source extent
   /clusters/cluster-1/extents/R20061115_Symm2264_013, unclaimed its
   disks Symm2264_013
```

Remove batch migration records

Remove the migration record only if the migration has been committed or canceled.

Migration records are in the /data-migrations/device-migrations context.

Use the **batch-migrate remove --file** ***filename*** command to remove records for the specified migration.

For example:

```
VPlexcli:/data-migrations/device-migrations> batch-migrate remove
   --file migrate.txt
```

or:

```
VPlexcli:>batch-migrate remove /data-migrations/device-migrations
   --file migrate.txt.
```

8

Configure the Network

The two WAN ports on each VPLEX director support dual Gigabit Ethernet inter-cluster links. The WAN ports are configured as part of the installation of a second cluster. This chapter describes the CLI contexts and procedures to change the configuration created during installation.

VPLEX hardware and WAN ports

VPLEX Geo clusters can include either of two types of director hardware. The number and speed of the WAN ports on each type of director is as follows:

- **VS1** - The Wide Area Network (WAN) communication interface card (SLiC) has four 1 Gigabit Ethernet (GbE) ports. Only two ports are used for VPLEX WAN communications.

  The ports are named GE00 and GE01.

- **VS2** - WAN SLiC has two 10 Gigabit Ethernet (10 GbE) ports.

  The ports are named XG00 and XG01.

WARNING

Data carried on WAN ports on both VS1 and VS2 directors and between clusters in VPLEX Metro and Geo configurations is not encrypted.

To prevent DNS attacks, the WAN ports should be routed only on secure and trusted networks.

Refer to the EMC Simple Support Matrix (ESSM) for information about encryption devices supported in VPLEX configurations.

Port groups

All ports named GE00 or XG00 (in a cluster) are collectively referred to as port-group 0.

All ports named GE01 or XG01 (in a cluster) are collectively referred to as port-group 1.

Note: Port group names (port-group-0 and port-group-1) can not be modified.

Geo WAN port configuration rules

Geo WAN ports must conform to the following rules:

- The two WAN ports on a director should be on different physical networks, and must be on different subnets so that port GE00/XG00 (port group 0) can not see port GE01/XG01 (port group 1) on any director.
- All port GE00/XG00s in the cluster (one from each director) must be in the same subnet and connected to the same LAN. Ports in the same subnet are usually connected to the same Ethernet switch.
- All port GE01/XG01s must be in one subnet, which cannot be the same subnet used for ports GE00/XG00.
- Each director must have 2 statically assigned IP addresses; one in each subnet.
- Each cluster must have an additional statically assigned IP address in each subnet (cluster IP address). This address is used during discovery. The cluster IP address is not tied to a specific physical port. Any director may host the cluster IP address.
- The management port subnet can not be the same as either subnet used for the WAN ports.

Sub-contexts

The /clusters/*cluster*/cluster-connectivity has three sub-contexts:

- "subnets context"
- "port-groups context"
- "option-sets context"

CLI contexts

The parent context for configuring Ethernet and WAN connections is:

/clusters/*cluster*/cluster-connectivity

The /clusters/*cluster*/cluster-connectivity context includes the following addresses:

- **discovery-address** - The multicast address local directors use to discover the cluster.

Note: Multicast must be enabled on the local switch connecting the directors' Ethernet ports.

- **discovery-port** - The port local directors use (along with the discovery-address) to find the other directors in same cluster.
- **listening-port** - The port local directors use to communicate with the other cluster. The listening port is used when connecting to the directors in the other cluster.

  The default values for these three addresses should not be changed. They are used by the local directors to discover each other within the cluster.

IMPORTANT

The listening port must be open through any fire walls between clusters.

Use the **set** command to change the three addresses.

Use the **set** command with no arguments to display the allowed values for the three addresses:

```
VPlexcli:/clusters/cluster-1/cluster-connectivity> set
attribute                 input-description
------------------------  ------------------------------------------------------
discovery-address         Takes w.x.y.z where w,x,y,z are in [0..255]
                          e.g. 10.0.1.125.
discovery-port            Takes an integer between 1024 and 32767.
listening-port            Takes an integer between 1024 and 32767.
name                      Read-only.
remote-cluster-addresses  Read-only.
```

A second set of addresses directly under the /clusters/*cluster*/ cluster-connectivity context are the

- **remote-cluster-addresses -** A list of reachable remote cluster addresses.

These are the cluster addresses assigned to the other cluster, both the one in the port-group-0 and the one in port-group-1. There are exactly 2.

To change a remote address, you must first clear the remote address.

Use the **remote-clusters clear-addresses** and **remote-clusters add-addresses** commands to add or clear entries in this list.

For example, to change address 42.29.20.214 to 42.29.20.254:

```
VPlexcli:/clusters/cluster-1/cluster-connectivity> remote-clusters clear-addresses
--remote-cluster cluster-2 --addresses 42.29.20.214:11000

VPlexcli:/clusters/cluster-1/cluster-connectivity> remote-clusters add-addresses
--remote-cluster cluster-2 --addresses 42.29.20.254:11000
```

Alternatively, use the **--default** argument to create a default list of reachable IP addresses (using the cluster-address attribute of the active subnets of remote clusters) for all remote clusters).

```
VPlexcli:/clusters/cluster-1/cluster-connectivity> remote-clusters add-addresses --default
```

Default values are determined by the cluster-address attribute of the active subnets from all remote clusters. For example:

```
remote-cluster-addresses cluster-2
   [192.168.91.252:11000,192.168.101.252:11000]
```

subnets context

A subnet is a logical subdivision of an IP network. VPLEX IP addresses are logically divided into two fields:

- A network or routing prefix.

  On a VPLEX, the prefix attribute is really a prefix and subnet mask. specified as an IP address and subnet mask in integer dot notation, separated by a colon.

  For example: 192.168.20.0:255.255.255.0

- A specific identifier for the configuration or the network interface.

IMPORTANT

VPLEX subnet addresses must be consistent, that is the cluster address and the gateway address must be in the subnet specified by the prefix.

If a change is made to the subnet, the change is validated and applied to all ports using this subnet.

When re-configuring a port-group, there are multiple values that must be consistent with each other. It may be necessary to clear or erase some attribute values before others can be changed.

```
VPlexcli:/clusters/cluster-1/cluster-connectivity> cd subnets/

VPlexcli:/clusters/cluster-1/cluster-connectivity/subnets> ll
Name
--------------
cluster-1-SN00
cluster-1-SN01
default-subnet
```

Use the following 4 CLI commands to create, modify, and delete subnets:

- **subnet clear**
- **subnet create**
- **subnet destroy**
- **subnet modify**

port-groups context

Ports named GE00/XG00 on each cluster are collectively referred to as port-group-0. There are two port-group-0s, one in each cluster. The port-group-0s on each cluster form one communication channel between the clusters.

Ports named GE01/XG01 on each cluster are collectively referred to as port-group-1. There are two port-group-1s, one in each cluster. The port-group-1s on each cluster form a second communication channel between the clusters.

The number of ports in each port-group varies depending on the number of engines in each cluster.

In the following example, a VPLEX Geo configuration has 1 engine in each cluster:

```
VPlexcli:/clusters/cluster-1/cluster-connectivity> cd port-groups/

VPlexcli:/clusters/cluster-1/cluster-connectivity/port-groups> ll
Name          Subnet          Option Set       Enabled      Member Ports
------------  --------------  ---------------  -----------  ----------------------------------------
port-group-0  cluster-1-SN00  optionset-com-0  all-enabled  engine-1-1|A2-XG00|192.168.11.140|enabled,
                                                            engine-1-1|B2-XG00|192.168.11.142|enabled
port-group-1  cluster-1-SN01  optionset-com-1  all-enabled  engine-1-1|A2-XG01|192.168.12.140|enabled,
                                                            engine-1-1|B2-XG01|192.168.12.142|enabled
```

Port-groups include the following properties:

- **member-ports** - A read-only list of ports that are part of this port-group, including their subnet, option-set, address and owner engine and director.
- **option-set** - The option-set associated with this port-group, or “inconsistent” if there is a problem in the configuration. This property is read-only.
- **subnet** - The subnet associated with this port-group, or “inconsistent” if there is a problem in the configuration.
- **enabled** - Summarizes the 'enabled' status of the individual member ports.
  - **all-enabled** - All member-ports in the port-group are enabled.
  - all-disabled - All member-ports in the port-group are disabled.
  - **inconsistent** - All member-ports don't have the same enabled status.

Use the **set** command to modify the subnet applied to a port-group:

```
VPlexcli:/clusters/cluster-1/cluster-connectivity/port-groups/port-group-1> set subnet
cluster-1-SN01

VPlexcli:/clusters/cluster-1/cluster-connectivity/port-groups/port-group-1> ll
Name          Value
------------  ------------------------------------------------------------------
enabled       all-enabled
member-ports  A2-XG01 [cluster-1-SN01 optionset-com-1 engine-1-1 director-1-1-A
              192.168.12.140 enabled], B2-XG01 [cluster-1-SN01 optionset-com-1
              engine-1-1 director-1-1-B 192.168.12.142 enabled]
option-set    optionset-com-1
subnet        cluster-1-SN01
```

option-sets context

Option-sets group connection properties so that they can be collectively applied to the ports contained in a port-group.

Option sets include the following properties:

- **connection-open-timeout** - The time in seconds to wait for a response when attempting to establish a connection.

  Default: 3 seconds.

  Range: 2 - 10 seconds.

- **keepalive-timeout** - The time in seconds to keep a connection open while it's idle.

  Default: 10 seconds.

Range: 5 - 20 seconds.

- **socket-buf-size** - The receiving/sending buffer sizes in bytes.

  Default: 1 MB.

  Range: 16 KB - 256 MB.

CAUTION

Consult EMC Customer Support before changing the connection-open-timeout and/or keepalive-timeout.

Use the **set** command to modify one or more properties of an option set:

```
VPlexcli:/clusters/cluster-1/cluster-connectivity/option-sets/optionset-com-1> set
connection-open-timeout 3s
VPlexcli:/clusters/cluster-1/cluster-connectivity/option-sets/optionset-com-1> set
keepalive-timeout 10s
VPlexcli:/clusters/cluster-1/cluster-connectivity/option-sets/optionset-com-1> set
socket-buf-size 10M
VPlexcli:/clusters/cluster-1/cluster-connectivity/option-sets/optionset-com-1> ll
Name                     Value
-----------------------  -----
connection-open-timeout  3s
keepalive-timeout        10s
socket-buf-size          10M
```

Modify the network configuration

About disabling port-groups

Before disabling a port-group, always ensure complete connectivity through the other port-group.

WARNING

An inter-cluster link outage will occur if a port-group is disabled when there are missing connections through the other port-group.

Change a port group's network configuration

When network changes are made in the data center it may be necessary to change the configuration of a port group.

The general procedure to modify a port group includes the following steps:

1. Verify connectivity.
2. Disable the target port-group
3. Unassign the subnet
4. Change one or more subnet properties, as desired
5. Change member-port IP addresses, as desired
6. Change the option-set values, as desired
7. Re-apply the modified subnet to the port-group
8. Enable the target port-group
9. Update the remote-cluster address on the second cluster
10. Verify connectivity

The following section describes the procedure in greater detail:

1. Verify connectivity.

Use the **connectivity validate-wan-com** command to verify that all directors have complete connectivity through the other port group.

```
VPlexcli:/> connectivity validate-wan-com
connectivity: FULL

port-group-1 - OK - All expected connectivity is present.
port-group-0 - OK - All expected connectivity is present.
```

2. Disable the target port-group

See About disabling port-groups on page 134.

Use the **set** command to disable the member ports in the target port-group:

```
VPlexcli:/clusters/cluster-1/cluster-connectivity/port-groups/port-group-1> set enabled
all-disabled
```

3. Unassign the subnet

Use the **set subnet " "** command to unassign the subnet from the port-group (set the port-group's subnet value to null):

```
VPlexcli:/clusters/cluster-1/cluster-connectivity/port-groups/port-group-1> set subnet ""
```

4. Change one or more subnet properties, as desired

Use the **subnet modify** command to modify one or more subnet properties. The syntax for the command is:

```
subnet modify
   [-s|--subnet] subnet
   [-a|--cluster-address] address
   [-g|--gateway] IP address
   [-p|--prefix] prefix
```

Required argument:

[-s | --subnet] ***subnet*** - Context path of the subnet configuration to modify.

Optional arguments:

[-a | --cluster-address] ***address*** - The public address of the cluster to which this subnet belongs.

[-g | --gateway] ***IP address*** - The gateway address for this subnet.

[-p | --prefix] ***prefix*** - The prefix/subnet mask for this subnet. Specified as an IP address and subnet mask in integer dot notation, separated by a colon. For example, 192.168.20.0:255.255.255.0

To modify the subnet's public IP address:

```
VPlexcli:/clusters/cluster-1/cluster-connectivity/subnets/cluster-1-SN01> subnet modify
--subnet cluster-1-SN01 --cluster-address 192.168.12.200
```

CAUTION

If the prefix is changed, ensure that the cluster IP address, gateway address, and port IP addresses are all consistent with the subnet prefix.

Use the **ll** command to display the addresses of the ports in the target port-group:

```
VPlexcli:/> ll /clusters/cluster-1/cluster-connectivity/port-groups/port-group-1

/clusters/cluster-1/cluster-connectivity/port-groups/port-group-1:
Name          Value
------------  ------------------------------------------------------------------
enabled       all-enabled
member-ports  engine-1-1|A2-XG01|192.168.10.140|enabled,
              engine-1-1|B2-XG01|192.168.10.142|enabled
option-set    optionset-com-1
subnet        -
```

5. Change member-port IP addresses, as desired

Use the **set** command to change the address of ports in the port-group to the new prefix:

```
VPlexcli:/> cd engines/engine-1-1/directors/director-1-1-A/hardware/ports/A2-XG01
VPlexcli:/engines/engine-1-1/directors/director-1-1-A/hardware/ports/A2-XG01> set address
192.168.12.140
VPlexcli:/engines/engine-1-1/directors/director-1-1-A/hardware/ports/A2-XG01> cd
engines/engine-1-1/directors/director-1-1-B/hardware/ports/B2-XG01
VPlexcli:/engines/engine-1-1/directors/director-1-1-B/hardware/ports/B2-XG01> set address
192.168.12.142
```

6. Change the option-set values, as desired

Use the **set** command to change option-set attribute values as desired:

```
VPlexcli:/clusters/cluster-1/cluster-connectivity/option-sets/optionset-com-1> set
connection-open-timeout 3s
```

7. Re-apply the modified subnet to the port-group

Use the **set** command to re-apply the modified subnet to the port-group:

```
VPlexcli:/clusters/cluster-1/cluster-connectivity/port-groups/port-group-1> set subnet
/clusters/cluster-1/cluster-connectivity/subnets/cluster-1-SN01
```

Use the **ll** command to confirm the changes:

```
VVPlexcli:/clusters/cluster-1/cluster-connectivity/port-groups/port-group-1> ll
Name          Value
------------  ------------------------------------------------------------------
enabled       all-disabled
member-ports  A2-XG01 [cluster-1-SN01 optionset-com-1 engine-1-1 director-1-1-A
              192.168.12.140 enabled], B2-XG01 [cluster-1-SN01 optionset-com-1
              engine-1-1 director-1-1-B 192.168.12.142 enabled]
option-set    optionset-com-1
subnet        cluster-1-SN01
```

8. Enable the target port-group

Use the **set** command to enable the target port-group:

```
VPlexcli:/clusters/cluster-1/cluster-connectivity/port-groups/port-group-1> set enabled
all-enabled

VPlexcli:/clusters/cluster-1/cluster-connectivity/port-groups/port-group-1> ll
Name          Value
------------  ------------------------------------------------------------------
enabled       all-enabled
member-ports  A2-XG01 [cluster-1-SN01 optionset-com-1 engine-1-1 director-1-1-A
              192.168.12.140 enabled], B2-XG01 [cluster-1-SN01 optionset-com-1
              engine-1-1 director-1-1-B 192.168.12.142 enabled]
option-set    optionset-com-1
subnet        cluster-1-SN01
```

9. Update the remote-cluster address on the second cluster

Use the **remote-clusters clear-addresses** command to clear the remote addresses.

```
VPlexcli:/clusters/cluster-1/cluster-connectivity> remote-clusters clear-addresses
--remote-cluster cluster-2 --addresses 42.29.20.214:11000
```

Alternatively, use the **remote-clusters add-addresses** command with the **--default** argument to create a default list of reachable IP addresses (using the cluster-address attribute of the active subnets of remote clusters) for all remote clusters).

```
VPlexcli:/clusters/cluster-1/cluster-connectivity> remote-clusters add-addresses --default
```

Default values are determined by the cluster-address attribute of the active subnets from all remote clusters. For example:

```
remote-cluster-addresses cluster-2
   [192.168.91.252:11000,192.168.101.252:11000]
```

10. Verify connectivity

Use the **connectivity validate-wan-com** command to verify connectivity:

```
VPlexcli:/> connectivity validate-wan-com
connectivity: FULL
```

```
port-group-1 - OK - All expected connectivity is present.
port-group-0 - OK - All expected connectivity is present.
```

Change one port's IP address

Use the **set** command to change a single WAN COM port's IP address to a different address.

It is not necessary to first disable the port-group.

Use the **ll** command in port-group context to display the port name and address:

```
VPlexcli:/clusters/cluster-1/cluster-connectivity/port-groups> ll
Name          Subnet          Option Set       Enabled      Member Ports
------------  --------------  ---------------  -----------  ----------------------------------------
port-group-0  cluster-1-SN00  optionset-com-0  all-enabled  engine-1-1|A2-XG00|192.168.11.140|enabled,
                                                            engine-1-1|B2-XG00|192.168.11.142|enabled
port-group-1  cluster-1-SN01  optionset-com-1  all-enabled  engine-1-1|A2-XG01|192.168.12.140|enabled,
                                                            engine-1-1|B2-XG01|192.168.12.142|enabled
```

Use the **set** command in the target port's context to change the port's IP address:

```
VPlexcli:/> cd engines/engine-1-1/directors/director-1-1-A/hardware/ports /A2-XG00

VPlexcli:/engines/engine-1-1/directors/director-1-1-A/hardware/ports/A2-XG00> set address
192.168.10.140
```

Change a port-group's MTU size

To change the MTU size for both port groups, modify one port-group at a time.

Before disabling a port group, always ensure complete connectivity through the other port group.

To change a port group's MTU size:

1. Verify connectivity.

Use the **connectivity validate-wan-com** command to verify that all directors have complete connectivity through the other port group.

```
VPlexcli:/> connectivity validate-wan-com
connectivity: FULL

port-group-1 - OK - All expected connectivity is present.
port-group-0 - OK - All expected connectivity is present.
```

2. Disable the target port-group

See About disabling port-groups on page 134.

Navigate to the port-group context of the target port-group:

```
VPlexcli:/> cd /clusters/cluster-1/cluster-connectivity/port-groups/port-group-1
VPlexcli:/clusters/cluster-1/cluster-connectivity/port-groups/port-group-1
```

Use the **set** command to disable the member ports in the target port-group:

```
VPlexcli:/clusters/cluster-1/cluster-connectivity/port-groups/port-group-1> set enabled
all-disabled
```

3. Modify the MTU size in the subnet

Navigate to the subnet context for the subnet associated with the target port-group:

```
VPlexcli:/> cd /clusters/cluster-1/cluster-connectivity/subnets/cluster-1-SN01
VPlexcli:/clusters/cluster-1/cluster-connectivity/subnets/cluster-1-SN01>
```

Use the **set** command to change the MTU (valid values are 96 - 9000):

CAUTION

The VPLEX CLI accepts MTU values lower than 96, but they are not supported. Entering a value less than 96 prevents the port-group from operating.

```
VPlexcli:/clusters/cluster-1/cluster-connectivity/subnets/cluster-1-SN01> set mtu 1480
WARNING: Incompatible MTU settings on clusters. You must also set the MTU in subnet
'cluster-2-SN01' (cluster-2) to 1480. Performance will be negatively impacted by incorrect
settings.
```

CAUTION

MTU size on the VPLEX and the attached switch must be the same.

Type the **ll** command to verify the new MTU on VPLEX:

```
VPlexcli:/clusters/cluster-1/cluster-connectivity/subnets/
   cluster-2-SN01> ll
Name                   Value
---------------------  -------------------------
cluster-address        192.168.2.252
gateway                192.168.2.1
mtu                    1480
.
.
.
```

Verify that the MTU size on VPLEX matches the MTU on attached switch.

Repeat for the second subnet.

4.Enable the port group

Navigate to the port-group context of the target port-group:

```
VPlexcli:/> cd /clusters/cluster-1/cluster-connectivity/port-groups/port-group-1
VPlexcli:/clusters/cluster-1/cluster-connectivity/port-groups/port-group-1
```

Use the **set** command to enable the member ports in the target port-group:

```
VPlexcli:/clusters/cluster-1/cluster-connectivity/port-groups/port-group-1> set enabled
all-enabled
```

5.Validate connectivity

Use the **connectivity validate-wan-com** command to verify that the directors have complete connectivity through all port groups.

```
VPlexcli:/> connectivity validate-wan-com
connectivity: FULL

port-group-1 - OK - All expected connectivity is present.
port-group-0 - OK - All expected connectivity is present.
```

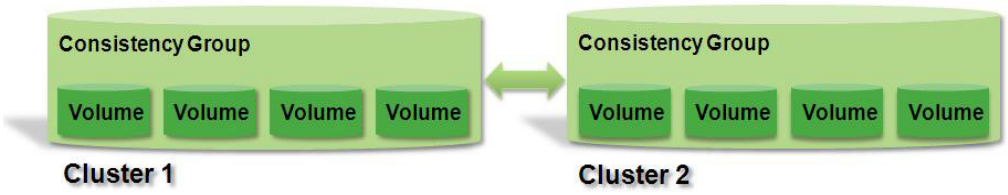
9

Consistency groups

This chapter describes the following topics and procedures:

About VPLEX consistency groups

VPLEX consistency groups aggregate volumes to enable the application of a common set of properties to the entire group.



Consistency groups aggregate up to 1000 virtual volumes into a single entity that can be managed as easily as individual volumes.

Consistency group detach rules define on which cluster I/O continues during cluster or inter-cluster link failures.

If all storage for an application with rollback capabilities is in a single consistency group, the application can recover from a complete cluster failure or inter-cluster link failure with little or no data loss. Data loss, if any is determined by the application data access pattern and the consistency group's cache-mode.

All consistency groups guarantee a crash consistent image of their member virtual-volumes. In the event of a director, cluster, or inter-cluster link failure, consistency groups prevent possible data corruption.

Create consistency groups for sets of volumes that require the same I/O behavior during failures.

There are two types of consistency groups:

- "Synchronous consistency groups"

  Synchronous consistency groups aggregate local and distributed volumes on VPLEX Local and VPLEX Metro systems separated by 5ms or less of latency.

- "Asynchronous consistency groups"

  Asynchronous consistency groups aggregate distributed volumes on VPLEX Geo systems separated by 50ms or less of latency.

  Note: In the current release, asynchronous consistency groups cannot be replicated by RecoverPoint.

A consistency group has either synchronous cache mode *or* asynchronous cache mode, but not both.

Synchronous consistency groups

Synchronous consistency groups provide a convenient way to apply rule sets and other properties to a group of volumes in a VPLEX Local or VPLEX Metro.

Synchronous consistency groups simplify configuration and administration on large systems.

VPLEX supports up to 1024 synchronous consistency groups.

A synchronous consistency group:

- Contains up to 1000 virtual volumes.
- Contains either local or distributed volumes, (but not a mixture of both).
- Contains volumes with either global or local visibility.
- Uses write-through caching (known as synchronous cache mode in the VPLEX user interface).

  Write order fidelity is maintained by completing all writes to disk before acknowledging the write to the host.

Figure 6 shows a synchronous consistency group that spans two clusters in a VPLEX Metro configuration.

- The hosts at both clusters write to the VPLEX distributed volumes in the consistency group.
- VPLEX writes data to the back-end storage on both clusters
- An acknowledgment is returned to the host issuing the write.

This guarantees that the image on the back end storage is an exact copy on both sides.

Using Consistency Groups with VPLEX Witness

VPLEX Witness failure recovery semantics apply *only* to volumes in synchronous consistency groups and only if a rule identifying specific preference is configured.

Using Consistency Groups with RecoverPoint

Recover Point protection can only be enabled through volumes in synchronous consistency groups with the recover-point flag set.

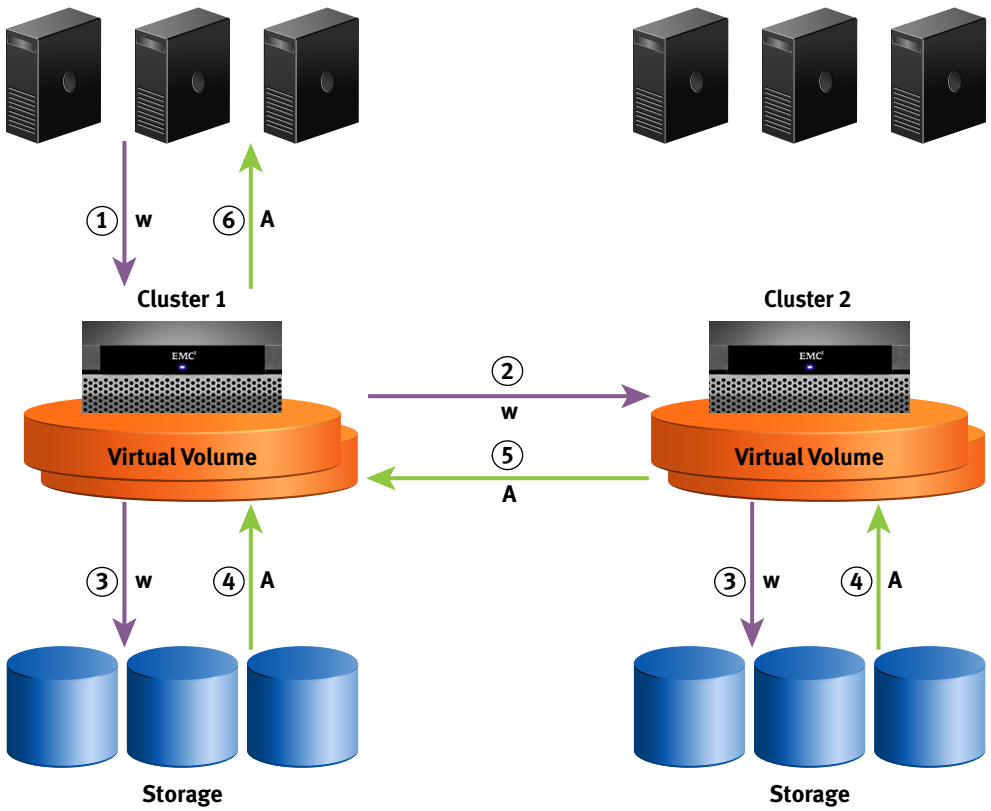


Figure 6 Synchronous consistency group

Synchronous consistency groups: visibility

Synchronous consistency groups support either distributed or local volumes (but not both in the same consistency group).

Local synchronous consistency groups have only local volumes as members. Local synchronous consistency groups can have the "visibility" property set to either:

- "Local visibility"- The local volumes in the consistency group are visible only to local cluster.
- "Global visibility"- The local volumes in the consistency group have storage at one cluster, but are visible to both clusters.

Local visibility

Local consistency groups with the "visibility"property set to only the local cluster read and write only to their local cluster.

Figure 7 shows a local consistency group with local visibility.

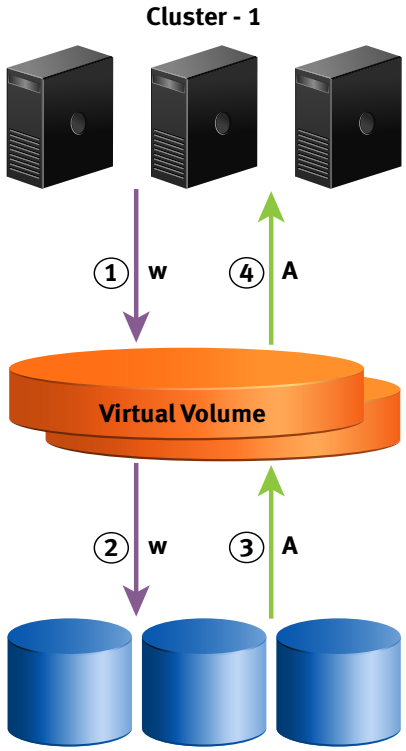


Figure 7 Local consistency groups with local visibility

Global visibility

If the local consistency group's "visibility" property is set to both clusters (global visibility), both clusters can receive I/O from the cluster that does not have a local copy.

All writes from that remote cluster pass over the inter-cluster WAN link before they are acknowledged.

Any reads that cannot be serviced from local cache are also transferred across the link. This allows the remote cluster to have instant on-demand access to the consistency group, but also adds additional latency for the remote cluster.

Local consistency groups with global visibility are supported in VPLEX Metro environments. Only local volumes can be placed into the local consistency group with global visibility. Local consistency groups with global visibility always use

write-through cache mode (synchronous cache mode). I/O that goes to local consistency groups with global visibility will always be synchronous.

Figure 8 shows a local consistency group with global visibility.

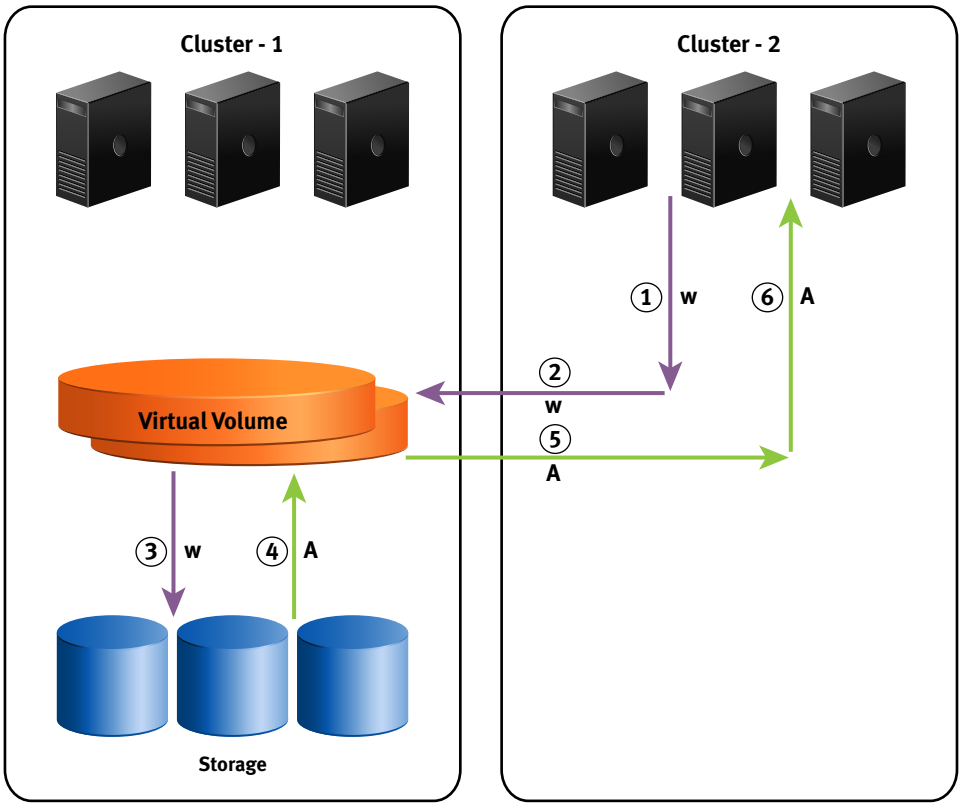


Figure 8 Local consistency group with global visibility

Asynchronous consistency groups

Asynchronous consistency groups apply rule-sets and other properties to distributed volumes in a VPLEX Geo.

Asynchronous consistency groups are required to support distributed volumes in a VPLEX Geo.

VPLEX supports up to 16 asynchronous consistency groups.

An asynchronous consistency group:

- Contains up to 1000 virtual volumes.
- Contains only distributed volumes.
- Contains volumes with either global or local visibility.
- Uses write-back caching (known as asynchronous cache mode in the VPLEX user interface).

Note: In the current release, asynchronous consistency groups cannot be replicated by RecoverPoint.

Write-back caching

In a synchronous cache mode, write order fidelity is maintained by batching I/O between clusters into packages called deltas that are exchanged between clusters.

Each delta contains a group of writes that were initiated in the same window of time. All writes in one delta are guaranteed to be newer than all writes in the next delta.

Write order consistency is maintained on delta set boundaries, not on individual writes.

Entire deltas are exchanged between clusters and committed to disks as a logical unit.

Each asynchronous consistency group maintains its own queue of deltas in various states:

- **open** - The delta is accepting new writes.
- **closed** - The delta is not accepting writes. Deltas are closed when their timer expires or they are full
- **exchanging** - The delta's contents are being synchronized between the clusters.
- **exchanged** - The delta is exchanged with the remote cluster. The delta's contents are the same at all clusters.
- **committing**- The delta's contents are being written out to the back-end storage.
- **committed** - The write of the delta's contents to the back-end storage is complete.

There can be only one delta in each state except closed.

There can be multiple deltas in the closed delta queue.

Before a delta is exchanged between clusters, data within the delta can vary by cluster. After a delta is exchanged and committed, data is exactly the same on both clusters.

If access to the back end array is lost while the system is writing a delta, the data on disk is no longer consistent and requires automatic recovery when access is restored.

Asynchronous cache mode can deliver better performance, but there is a higher risk that data will be lost if:

- Multiple directors fail at the same time
- There is an inter-cluster link partition and both clusters are actively writing and instead of waiting for the link to be restored, the user chooses to accept a data rollback in order to reduce the RTO
- The cluster that is actively writing fails

Asynchronous consistency group: active/passive

In an asynchronous consistency group, the host receives acknowledgment after the write reaches the VPLEX cache and has been protected to another director in the local cluster.

VPLEX collects writes at the cluster in the form of a *delta*. At a later point in time, the clusters exchange deltas creating a globally merged delta. The clusters send communication messages back and forth between each other to facilitate the exchange.

Once the exchange is complete, the clusters write a global merged delta to the back-end arrays. Asynchronous consistency group active/passive

In one cluster is actively reading and writing. This simplifies the view of asynchronous I/O. Application data is written to the director in Cluster 1 and protected in another director of Cluster 1.

VPLEX collects writes into a delta of a fixed size. Once that delta is filled or when a set time period ("default closeout-time") has elapsed, the two clusters of the VPLEX Geo begin a communication process to exchange deltas.

The combination of the deltas is referred to as a global delta. In , the global delta only includes the writes that occurred on Cluster 1 because Cluster 2 was inactive. This data is then written to the back-end storage at Cluster 1 and Cluster 2.

Asynchronous consistency group: active/active

Figure 9 shows asynchronous I/O when both clusters are actively reading and writing. The applications at Cluster 1 and Cluster 2 are both writing to their local VPLEX cluster.

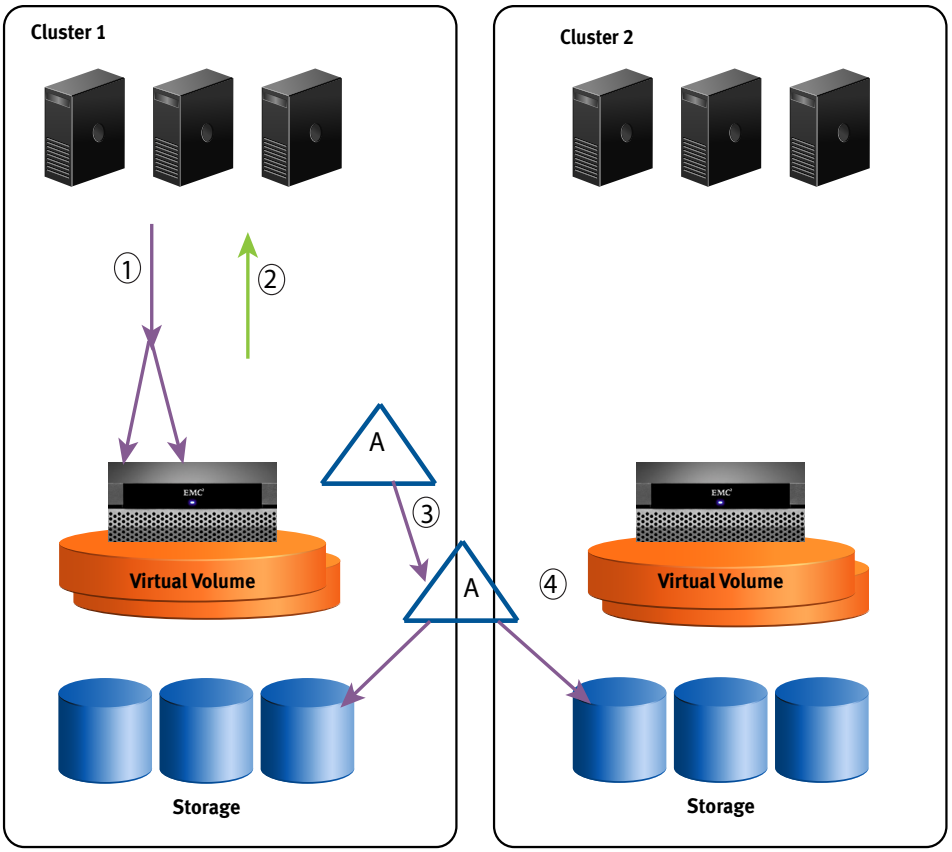


Figure 9 **Asynchronous consistency group active/active**

Figure 9 shows I/O in an a synch or no us consistency group when both clusters are actively reading and writing. The applications at Cluster 1 and Cluster 2 are both writing to their local VPLEX cluster.

Application writes are acknowledged once each cluster caches the data in two directors. The VPLEX collects the data in deltas at each cluster. After the "default closeout-time", or after a deltas becomes full, the clusters exchange deltas. VPLEX then writes the resulting delta to back end storage at each cluster.

This process coordinates data written to the storage at each cluster.

Asynchronous consistency group: RPO

The "default closeout-time" and "maximum-queue-depth" properties control the recovery point objective in the event of a failure.

The *recovery point objective (RPO)* is the maximum acceptable level of data loss resulting from a failure. RPO represents the point in time (prior to the failure) to which lost data can be recovered.

Use the "maximum-queue-depth" and "default closeout-time" properties to control the RPO. Refer to "Calculating RPO in VPLEX Geo systems" on page 240.

Note: Both properties must be set sufficiently high to accommodate the latency and usable bandwidth between clusters.

Properties of consistency groups

Properties of a consistency group are applied to all the virtual volumes in the consistency group.

All consistency groups have configurable properties that determine I/O behavior, including:

- "cache mode"
- "visibility"
- "storage-at-clusters"
- "local-read-override"
- "detach-rule"
- "auto-resume-at-loser"
- "virtual-volumes"
- "recoverpoint-enabled"

Additional configurable properties are applicable only to consistency groups with asynchronous cache mode, including:

- "active cluster"
- "default closeout-time"
- "maximum-queue-depth"

IMPORTANT

When RecoverPoint is deployed, it may take up to 2 minutes for the RecoverPoint cluster to take note of changes to a VPLEX consistency group. Wait for 2 minutes after making the following changes to a VPLEX consistency group before creating or changing a RecoverPoint consistency group:
- Add/remove virtual volumes to/from a VPLEX consistency group
- Enable/disable the recoverpoint-enabled property of a VPLEX consistency group
- Change the detach rule of a VPLEX consistency group

cache mode

Cache mode describes how data is written to storage. Cache mode can be either synchronous or asynchronous:

- **Synchronous cache mode** - Supported on VPLEX Local and VPLEX Metro configurations where clusters are separated by up to 5 ms of latency. In synchronous cache mode, writes to the back-end storage volumes are not acknowledged to the host until the back-end storage volumes acknowledge the write.

  Writes to the virtual volumes in a synchronous consistency group are written to disk only if all prior acknowledged writes to all volumes in the consistency group are also present on the disk.

- **Asynchronous cache mode** - Supported on VPLEX Geo configurations where clusters separated by up to 50 ms of latency. In asynchronous cache mode, host writes to a distributed volume are acknowledged back to the host after the data is protected in the cache of another director in the local cluster. These writes are grouped into deltas.

Writes to the virtual volumes in an asynchronous consistency group are ordered such that all the writes in a given delta are written before writes from the next delta.

A maximum of 16 consistency groups with asynchronous cache mode can be configured.

Each asynchronous consistency group can contain up to 1000 virtual volumes.

Use the **set** command in /clusters/*cluster*/consistency-groups/*consistency-group* context to modify the cache mode property.

Note: Storage-at-clusters must be set to cluster-1 and cluster-2 before cache-mode can be changed to asynchronous (asynchronous mode is supported only for DR volumes).

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> set cache-mode asynchronous
```

CAUTION

Changing cache mode for a non-empty consistency group that is receiving host I/O requests may temporarily worsen I/O performance.

visibility

Visibility controls which clusters know about a consistency group.

Note: Visibility for consistency groups differs from the visibility property for devices. Devices can have visibility set to local (visible only to the local cluster) or global (visible to both clusters). All distributed devices have global visibility.

By default, a consistency groups's visibility property is set only to the cluster where the consistency group was created. If a consistency group is created on cluster-2, it is initially visible only on cluster-2.

The visibility of the volumes within the consistency group must match the visibility of the consistency group.

If the visibility of a volume in a consistency group is set to local, the visibility of the consistency group cannot be set to include other clusters. For example, if volume "LocalVolume" with visibility property set to "local" is added to consistency group 'TestCG" the visibility of TestCG cannot be modified to include other clusters.

In general, visibility is set to one of three options:

- Configure the consistency group to contain only volumes local to the local cluster.
- Configure the consistency group to contain only volumes that have storage at one cluster, but have global visibility.
- Configure the consistency group to contain only volumes that are distributed with legs at both clusters.

When a consistency group's visibility is set to a cluster, the consistency group appears below /clusters/cluster-n/consistency-groups context for the cluster.

Note: The context for a specified consistency group appears in a cluster's consistency group CLI context only if the ""visibility"" property of the consistency group includes that cluster.

Under normal operations, the visibility property can be modified to expand from one cluster to both clusters.

Use the **set** command in /clusters/*cluster*/consistency-groups/*consistency-group* context to modify the visibility property. If consistency group 'TestCG" is visible only at cluster-1, use the **set** command to make it visible to cluster-1 and cluster-2:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> set visibility cluster-1,cluster-2
```

If a consistency group contains virtual volumes with a given visibility (for example, a member volume's visibility is "local"), the visibility property for the consistency group cannot be changed to conflict with the visibility property of the member virtual volume.

For example, consistency group "TestCG" is visible only at cluster-1, and contains a volume "V" whose device is at cluster-1 and has local visibility. Both the following commands will fail, since the volume V is not visible at cluster-2.

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> set visibility cluster-1,cluster-2
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> set visibility cluster-2
```

storage-at-clusters

Storage-at-clusters tells VPLEX at which cluster the physical storage associated with a consistency group is located.

The storage-at-clusters property of a consistency group must be a non-empty subset of the consistency group's "visibility" property.

- If visibility is set to one cluster, then storage-at-clusters must be exactly the same as visibility.
- If visibility is set to two clusters (1 and 2), then storage-at-clusters can be one of:
  - cluster-1
  - cluster-2
  - cluster-1 and cluster-2

A volume that does not have local storage at every cluster specified by the storage-at-clusters property of a consistency group, cannot be added to the consistency group.

For example, if a volume has storage only at cluster-1, it cannot be added to a consistency group that has its storage-at-cluster property set to cluster-1 and cluster-2.

A volume that has local storage at more clusters than those specified by the storage-at-clusters property of a consistency group, cannot be added to the consistency group.

For example, if a volume has storage at cluster-1 and cluster-2, it cannot be added to a consistency group that has its storage-at-cluster property set to cluster-1.

The storage-at-clusters property cannot be modified if doing so conflicts with the topology of any of the volumes currently in the consistency group.

Use the **set** command in /clusters/*cluster*/consistency-groups/*consistency-group* context to modify the storage-at-clusters property. For example, to set the storage-at-clusters property to both clusters:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> set storage-at-clusters
cluster-1,cluster-2
```

Note: Best practice is to set the storage-at-clusters property when the consistency group is empty.

local-read-override

The local-read-override property determines whether the volumes in this consistency group use the local read override optimization.

When a director receives a read request, it first checks the distributed cache to see if that page is dirty (written to some director's cache, but not yet written to disk). If the page is dirty in any director's cache, the page is sent from that director to the reading director. The two directors can at be at the same cluster or in different clusters.

When a read request for a page is received by a director, and none of the directors in the same cluster have that page in cache, it has two ways to get the page:

- Query the other directors in other clusters whether they have the page in their caches.
- Read the page from the underlying back-end storage.

If no peer director has the page in its cache, the page is read from the underlying back-end storage. The local-read-override property can be set to:

true (default)- A director reading from a volume in this consistency group prefers to read from back-end storage over getting the data from a peer director's cache.

false - A director reading from a volume in the consistency group prefers to read from a peer director's cache.

Local-read-override should be set to true if the inter-cluster latency is large or back-end storage is fast and has a large cache of its own that enables it to respond faster than the VPLEX director.

Local-read-override should be set to false only if it is faster to retrieve pages from the remote cluster's cache than from the local cluster's storage. For example, if the clusters are located close to one another and the storage on the local cluster is very slow.

Use the **set** command in /clusters/*cluster*/consistency-groups/*consistency-group*/advanced context to modify the local-read-override property. For example, to disable local-read-override:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG/advanced> set local-read-override
false
```

detach-rule

Detach rules are a consistency group's policy for automatically picking a "winning" cluster when there is an inter-cluster link outage.

There are three consistency group detach rules:

- **no-automatic-winner** - The consistency group does not select a winning cluster.
- **active-cluster-wins** - Applicable only to asynchronous consistency groups. If one cluster was active and one was passive at the time of the link outage, the consistency group selects the active cluster as the winner.

  See "When a cluster is active vs. passive" on page 153.
- **winner *cluster-name* delay *seconds*** - Applicable only to synchronous consistency groups. The cluster specified by *cluster-name* is declared the winner if an inter-cluster link outage lasts more than the number of seconds specified by **delay**.

If a consistency group has a detach-rule configured, the rule applies to all volumes in the consistency group, and overrides any rule-sets applied to individual volumes.

This property is not relevant for local consistency groups.

By default, no specific detach rule is configured for a consistency group. Instead, the **no-automatic-winner** detach rule is set as default for a consistency group with visibility to both clusters.

Best practice is to apply detach rules to a consistency group that meet the needs of your application in terms of I/O continuance and data loss tolerance.

Use the **consistency-group set-detach-rule** commands to configure the detach-rule- for a consistency group:

- Use the "consistency-group set-detach-rule no-automatic-winner" command to set the detach-rule as no-automatic-winner:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG>
   set-detach-rule no-automatic-winner
```

- Use the "consistency-group set-detach-rule active-cluster-wins" command to set the detach-rule as active-cluster-wins:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG>
   set-detach-rule active-cluster-wins
```

- Use the "consistency-group set-detach-rule winner" command to specify which cluster is the winner, and the number of seconds VPLEX waits after a link outage before detaching the winning cluster:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG>
   set-detach-rule winner --cluster cluster-1 --delay 5s
```

auto-resume-at-loser

Determines whether the loser automatically resumes I/O when the inter-cluster link is repaired after a failure.

When the link is restored, the losing cluster finds out that it's the loser, and that the data on the winning cluster is now different. The loser must determine whether to suddenly change to the winner's data, or to keep suspending I/O.

By default, this property is set to false (auto-resume is disabled).

Usually, this property is set to false to give the administrator time to halt and restart the application. Otherwise, dirty data in the host's cache may be inconsistent with the image on disk to which the winning cluster has been writing. If the host flushes dirty pages out of sequence, the data image may be corrupted.

Set this property to true for consistency groups used in a cluster cross-connect. In this case, there is no risk of data loss since the winner is always connected to the host, avoiding out of sequence delivery.

true - I/O automatically resumes on the losing cluster after the inter-cluster link has been restored.

Set auto-resume-at-loser to true only when the losing cluster is servicing a read-only application such as servicing web pages.

false (default) - I/O remains suspended on the losing cluster after the inter-cluster link has been restored. I/O must be manually resumed.

Set auto-resume-at-loser to false for all applications that cannot tolerate a sudden change in data.

CAUTION

Setting this property to true may cause a spontaneous change of the data view presented to applications at the losing cluster when the inter-cluster link is restored. If the application has not failed, it may not be able to tolerate the sudden change in the data view and this can cause data corruption. Set the property to false except for applications that can tolerate this issue or cross connected hosts.

Use the **set** command in the `advanced` context to configure the **auto-resume** property for a consistency group:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG/advanced> set auto-resume-at-loser
true
```

virtual-volumes

Administrators can add and remove virtual volumes to a consistency group. In order to be added to a consistency group, a virtual volume:

- Must not be a logging volume
- Have storage at every cluster in the 'storage-at-clusters' property of the target consistency group
- Must not be a member of any other consistency group
- Any properties (such as detach rules or auto-resume) that conflict with those of the consistency group or are automatically changed to match those of the consistency group

Note: Virtual volumes with different properties are allowed to join a consistency group, but inherit the properties of the consistency group.

Use the "consistency-group list-eligible-virtual-volumes" command to display virtual volumes that are eligible to be added to a consistency group.

Use the "consistency-group add-virtual-volumes" command to add one or more virtual volumes to a consistency group.

Use the **ll /clusters/*cluster*/consistency-groups/ *consistency-group*** command to display the virtual volumes in the specified consistency group.

Use the "consistency-group remove-virtual-volumes" command to remove one or more virtual volumes from a consistency group.

recoverpoint-enabled

Starting in Release 5.1, VPLEX includes a RecoverPoint splitter. The splitter "splits" application writes so that the writes are sent to their normally designated storage volumes and a RecoverPoint Appliance (RPA) simultaneously.

To configure a consistency group for use with RecoverPoint:

- Cache mode must be synchronous
- Consistency groups with the "visibility" property set to both clusters must also have their "storage-at-clusters" set to both clusters in order to set the recoverpoint-enable property to true.

- All production and replica volumes associated with RecoverPoint must be in a VPLEX consistency group with the recoverpoint-enabled property set to true.

  Configure two consistency groups for each set of virtual volumes to be protected by RecoverPoint: one for the production volumes and one for the replica volumes.

  RecoverPoint journal volumes are not required to be in a consistency group with this attribute enabled.

In addition to setting this property to true, the VPLEX consistency group must be aligned with the RecoverPoint consistency group.

Use the **set** command in /clusters/*cluster*/consistency-groups/*consistency-group*/ context to configure the recoverpoint-enabled property:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> set recoverpoint-enabled true
```

Properties of asynchronous consistency groups

The following configurable properties are applicable only to consistency groups with asynchronous cache mode.

active cluster

The active cluster property is not configurable by users. Instead, a configuration changes dynamically between active/passive and active/active depending on the write activity at each cluster.

When a cluster is active vs. passive

Failure behaviors dictated by the **active-cluster-wins** detach rule vary depending on whether the configuration is active/passive or active/active at the time of the failure.

In asynchronous consistency groups, the process of synchronizing data between the two clusters includes the following phases:

- **Open** - Host writes to distributed volumes are acknowledged back to the host after the data is protected in the cache of another director in the local cluster. Writes from each application are collected into “deltas”.
- **Closed** - Deltas are closed when their timer expires or they are full.
- **Exchanged** - The delta is exchanged with the remote cluster.
- **Commit** - The delta's contents are written to back-end storage.

Data that has been protected to another director in the local cluster (but not exchanged with the remote cluster) is known as dirty data.

A configuration is **active/passive** if hosts at only one cluster were writing at the time of the failure. A configuration is **active/active** if hosts at both clusters were writing at the time of the failure.

A cluster is marked **passive** when all dirty data contributed by the cluster has been committed (written to the back end at both clusters). The cluster remains passive as long as no further data is contributed by the time the next delta closes. Specifically:

- Hosts at a cluster stop writing to a consistency group.
- The currently open delta closes, either due to a timer (default-closeout-time) or due to the delta becoming full at either cluster.
- The closed delta is exchanged and committed the back-end.
- The next open (empty at one cluster because the host is not writing) delta is closed.

- When closure of the next delta completes without any new data, that cluster is marked passive.
- When a host writes to a passive cluster (creates dirty data), the cluster is marked as active.

default closeout-time

Sets the default for the maximum time a delta remains open to accept new writes. Closeout-time can be set as either a positive integer or zero.

Default: 30 seconds.

zero (0) - There is no time limit on closing the open delta. The delta is closed when it is full and cannot accept more data.

A larger value for the default close-out -time unnecessarily exposes dirty data in the open delta if the exchange delta is idle.

The ideal default close-out -time is equal to the time it takes to exchange a full delta set.

The default closeout-time and maximum-queue-depth properties work together to allow administrators to fine-tune the maximum possible data loss in the event of an inter-cluster link outage.

Maximum RPO for the consistency group can be calculated as follows:

maximum-queue-depth x default closeout-time = RPO

Increasing either the default closeout-time or maximum-queue-depth property increases the maximum RPO in the event of a inter-cluster link outage or cluster failure.

Use the **set** command in /clusters/*cluster*/consistency-groups/*consistency-group*/advanced context to configure the closeout-time property for an asynchronous consistency group:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG/advanced> set closeout-time 12
```

maximum-queue-depth

For an asynchronous consistency group, this property configures the maximum possible depth of the delta processing queues.

Each consistency group maintains its own queue of deltas in various states:

- **open** - The delta is accepting new writes
- **closed** - The delta is not accepting writes.
- **exchanging** - The delta's contents are being synchronized between the clusters.
- **exchanged** - The delta's contents are the same at all clusters.
- **committing**- The delta's contents are being written out to the back-end storage.
- **committed** - The write of the delta's contents to the back-end storage is complete.

All deltas pass through these states, forming a "pipeline" in the order in which they were received. For applications that generate occasional brief bursts of writes, the rate

of incoming writes may (for a short time) be faster than the rate at which deltas are exchanged and committed.

Setting the maximum queue depth allows multiple closed deltas to wait to be exchanged and committed.

The default closeout-time and maximum-queue-depth properties work together to allow administrators to fine-tune the maximum possible data loss in the event of an inter-cluster link outage.

Decrease the maximum-queue-depth to decrease the maximum RPO in the event of a inter-cluster link outage or cluster failure.

Increase the maximum-queue-depth to increase the ability of the system to handle bursts of traffic.

CAUTION

Increasing the maximum-queue-depth also increases the data that must rollback in the case of a cluster failure or a link outage (I/O is resumed when the active cluster became the loser).

Default: 6.

Range: 6 - 64. The maximum value is platform-specific.

Use the **set** command in /clusters/*cluster*/consistency-groups/*consistency-group*/advanced context to configure the maximum-queue-depth property for an asynchronous consistency group:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG/advanced>set maximum-queue-depth 8
```

Manage consistency groups

IMPORTANT

A key best practice for creating and managing consistency groups is to create a 1:1 relationship between consistency groups and applications. All volumes (and only those volumes) required for an application should be in a single consistency group.

Create a consistency group

Before you begin

Before you create a consistency group, consider how it will be used:

- Are the clusters more than 100km from one another or is latency greater than 5 ms? Will the consistency group contain volumes on distributed devices? If yes, the cache mode must be set to asynchronous.
- At which cluster(s) is the underlying storage of the virtual volumes located? If volumes are located at both clusters, set the storage-at-cluster property as "cluster-1,cluster-2".
- Will I/O to the volumes in the consistency group be split by a RecoverPoint Appliance? If yes, set the recoverpoint-enable property to "true".
- What is the visibility of the virtual volumes to be added?

Some properties of virtual volumes and consistency groups limit which volumes can be added to a consistency group, or prevent a property of the consistency group from being modified.

For example, a consistency group's visibility property is set to "cluster-1". Virtual volumes local to cluster-1 are added. The visibility property of the consistency group can not be changed to either "cluster-2" or "cluster-1,cluster-2" since the volumes are not visible at cluster-2.

Create a consistency group

To create a consistency group and configure those properties that should be set before virtual volumes are added, do the following:

1. Use the **ls /clusters/*/consistency-groups/** command to display the names of all the consistency-groups:

```
VPlexcli:/> ls /clusters/*/consistency-groups/

/clusters/cluster-1/consistency-groups:
TestCG          local_test      test10      test11      test12  test13  test14
test15          test16          test5       test6       test7   test8   test9
vs_RAM_c1wins   vs_RAM_c2wins   vs_oban005  vs_sun190


/clusters/cluster-2/consistency-groups:
TestCG          local_test      test10      test11      test12  test13  test14
test15          test16          test5       test6       test7   test8   test9
vs_RAM_c1wins   vs_RAM_c2wins   vs_oban005  vs_sun190
```

2. Use the **consistency-group create** command to create a new consistency-group on a cluster. The syntax for the command is:

```
consistency-group create
   [-n|--name]consistency-group name
   [-c|--cluster] cluster
```

[-n | --name] ***consistency-group name*** - Name of the new consistency-group. Must be unique on the entire VPLEX.

[-c | --cluster] ***cluster*** - Context path of the cluster at which to create the consistency group. If the current context is a cluster or below, that cluster is the default. Otherwise, this argument is required.

Specify a name for the new consistency group that did not appear in the output in Step 1.

```
VPlexcli:/> consistency-group create --name TestCG --cluster cluster-1
```

3. Use the **ls /clusters/*cluster-id*/consistency-groups/ *consistency-group*/** command to display the new consistency-group:

```
VPlexcli:/> ls /clusters/cluster-1/consistency-groups/TestCG/

/clusters/cluster-1/consistency-groups/TestCG:

Attributes:
Name                  Value
--------------------  ------------------------------------------
active-clusters       []
cache-mode            synchronous
detach-rule           -
operational-status    [(cluster-1,{ summary:: ok, details:: [] })]
passive-clusters      []
recoverpoint-enabled  false
storage-at-clusters   []
virtual-volumes       []
visibility            [cluster-1]


Contexts:
Name           Description
-------------  -----------
advanced       -
recoverpoint   -
```

Set the visibility property

By default, the consistency group's "visibility" property is set to the cluster where the consistency group was created. If a consistency group is created on cluster-2, it is initially visible only on cluster-2.

Visibility can be configured as follows:

- **cluster-1** - volumes local to cluster-1.
- **cluster-2** - volumes local to cluster-2.
- **cluster-1,cluster-2** - volumes that are distributed with legs at both clusters.

4. Use the **set** command to configure the consistency group's visibility property.

CAUTION

The CLI context of the consistency group appears only at the cluster where the consistency group has visibility. If visibility is set from cluster-1 to include only cluster-2, the CLI context for the consistency group disappears at cluster-1 and is visible only from cluster-2.

To set the consistency group's visibility property to both clusters:

```
VPlexcli:/clusters/cluster-1/consistency-groups> set TestCG::visibility cluster-1,cluster-2
```

To set the consistency group's visibility property to cluster-1:

```
VPlexcli:/clusters/cluster-1/consistency-groups> set TestCG::visibility cluster-1
```

To set the consistency group's visibility property to cluster-2:

```
VPlexcli:/clusters/cluster-1/consistency-groups> set TestCG::visibility cluster-2
```

Set the storage-at-clusters property

By default, the consistency group's "storage-at-clusters" property is set to empty.

Storage-at-clusters tells VPLEX at which cluster the physical storage associated with a consistency group is located. If "visibility" is set to one cluster, then storage-at-clusters must be exactly the same as visibility. If visibility is set to two clusters (1 and 2), then storage-at-clusters can be one of:

- cluster-1
- cluster-2
- cluster-1 and cluster-2

A volume that does not have local storage at every cluster specified by the storage-at-clusters property of a consistency group, cannot be added to the consistency group.

5. Use the **set** command to configure the consistency group's storage-at-clusters property

```
VPlexcli:/clusters/cluster-1/consistency-groups> set TestCG::storage-at-clusters
cluster-1,cluster-2
```

Set the cache-mode property

By default, the cache mode for the new consistency group is synchronous.

If this consistency group is intended to have synchronous cache mode, skip to step 8.

6. Otherwise, use the **set** command to change the cache-mode to asynchronous:

```
VPlexcli:/clusters/cluster-1/consistency-groups> set TestCG::cache-mode asynchronous
```

Note: When you set the cache-mode to asynchronous, the CLI automatically applies the active-cluster-wins rule.

When you set the cache mode to synchronous, the CLI automatically applies a winner rule where the winner is the cluster with the lowest cluster ID (typically cluster-1) and the time-out is five seconds.

(Optional) set additional properties

7. Optionally, use one of the **consistency-group set-detach-rule** commands to apply a detach rule.

   For example; configure the detach-rule as active-cluster-wins:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> set-detach-rule active-cluster-wins
```

8. Use the **ll** command to display the new consistency group:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> ll

Attributes:
Name                  Value
--------------------  ------------------------------------------------------------
active-clusters       []
cache-mode            asynchronous
detach-rule           active-cluster-wins
operational-status    [(cluster-1,{ summary:: ok, details:: [] }), (cluster-2,{
                      summary:: ok, details:: [] })]
passive-clusters      [cluster-1, cluster-2]
recoverpoint-enabled  false
storage-at-clusters   [cluster-1, cluster-2]
```

```
virtual-volumes       []
visibility            [cluster-1, cluster-2]


                      Contexts:
                      Name          Description
                      ------------  -----------
                      advanced      -
                      recoverpoint  -
```

Refer to Table 9, "Display consistency group field descriptions," for descriptions of the fields in the display.

Add volumes to a consistency group

A maximum number of 1000 volumes can be added to a consistency group.

All volumes used by the same application should be grouped together in a consistency group.

Only volumes with storage at both clusters (distributed volumes) can be added to asynchronous consistency groups.

Only local volumes can be added to synchronous consistency groups with visibility and storage-at-cluster set to the local cluster.

Remote volumes can be added to synchronous consistency groups with visibility set to both clusters and storage-at-cluster set to one cluster.

To add virtual volumes to an existing consistency group, do the following:

1. Navigate to the target consistency group's context:

   ```
   VPlexcli:/> cd clusters/cluster-1/consistency-groups/ TestCG
   ```

2. Use the **consistency-group list-eligible-virtual-volumes** command to display virtual volumes are eligible to be added to the consistency group:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> consistency-group
list-eligible-virtual-volumes
[TestDDevice-1_vol, TestDDevice-2_vol, TestDDevice-3_vol, TestDDevice-4_vol,
TestDDevice-5_vol]
```

3. Use the **add-virtual-volumes** command to add virtual volumes to the consistency group.

   The syntax for the command is:

   ```
   consistency-group add-virtual-volumes
   [-v|--virtual-volumes] virtual-volume,virtual-volume,...
   [-g|--consistency-group] consistency-group
   ```

   [-v | --virtual-volumes] ***virtual-volume,virtual-volume,...*** - List of one or more comma-separated glob patterns or context paths of the virtual volume(s) to add.

   [-g | --consistency-group] ***consistency-group*** - Context path of the consistency group to which to add the specified virtual volume(s).

   To add a single virtual volume:

```
VPlexcli:/clusters/cluster-2/consistency-groups/TestCG> add-virtual-volumes --virtual-volumes
TestDDevice-2_vol
```

Note: The full path is not required if the volume name is unique in the VPLEX.

To add multiple volumes using a single command, separate virtual volumes by commas:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> add-virtual-volumes
TestDDevice-1_vol,TestDDevice-2_vol
```

4. Use the **ll** command to display the change:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> ll

Attributes:
Name                  Value
--------------------  ------------------------------------------------------------
active-clusters       []
cache-mode            asynchronous
detach-rule           active-cluster-wins
operational-status    [(cluster-1,{ summary:: ok, details:: [] }), (cluster-2,{
                      summary:: ok, details:: [] })]
passive-clusters      [cluster-1, cluster-2]
recoverpoint-enabled  false
storage-at-clusters   [cluster-1, cluster-2]
virtual-volumes       [TestDDevice-1_vol, TestDDevice-2_vol]
visibility            [cluster-1, cluster-2]


Contexts:
Name           Description
-------------  -----------
advanced       -
recoverpoint   -
```

Remove volumes from a consistency group

To remove one or more virtual volumes from a consistency group:

1. Use the **ll** command to display the virtual volumes in the target consistency group:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> ll

Attributes:
Name                  Value
--------------------  ------------------------------------------------------------
active-clusters       []
cache-mode            asynchronous
detach-rule           active-cluster-wins
operational-status    [(cluster-1,{ summary:: ok, details:: [] }), (cluster-2,{
                      summary:: ok, details:: [] })]
passive-clusters      [cluster-1, cluster-2]
recoverpoint-enabled  false
storage-at-clusters   [cluster-1, cluster-2]
virtual-volumes       [TestDDevice-1_vol, TestDDevice-2_vol,
                      TestDDevice-3_vol,TestDDevice-4_vol,
                      TestDDevice-5_vol]
visibility            [cluster-1, cluster-2]


Contexts:
Name           Description
-------------  -----------
advanced       -
recoverpoint   -
```

2. Use the **consistency-group remove-virtual-volumes** command to remove one or more virtual volumes.

WARNING

Before using consistency-group remove-virtual-volumes command on an asynchronous consistency group, ensure that no host applications are using the volumes you are removing.

The syntax for the command is:

```
consistency-group remove-virtual-volumes
[-v|--virtual-volumes] virtual-volume,virtual-volume,...
[-g|--consistency-group context path
```

[-v | --virtual-volumes] ***virtual-volume,virtual-volume,...*** - *Glob pattern or a list of one or more comma-separated context paths of the virtual volume(s) to remove from the consistency group.

[-g | --consistency-group] ***context path*** - * Context path of the consistency group from which to remove the specified virtual volume.

Remove one virtual volume from the root context:

```
VPlexcli:/> consistency-group remove-virtual-volumes
/clusters/cluster-1/virtual-volumes/TestDDevice-2_vol, --consistency-group
/clusters/cluster-1/consistency-groups/TestCG
```

To remove multiple virtual volumes with a single command, separate the volumes using commas:

```
VPlexcli:/> consistency-group remove-virtual-volumes
/clusters/cluster-1/virtual-volumes/TestDDevice-2_vol,
/clusters/cluster-1/virtual-volumes/TestDDevice-3_vol --consistency-group
/clusters/cluster-1/consistency-groups/TestCG
```

Remove two virtual volumes from the target consistency group context:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> remove-virtual-volumes
TestDDevice-2_vol, TestDDevice-3_vol
```

3. Use the **ls** command to display the change:

```
VPlexcli:/> ls clusters/cluster-1/consistency-groups/TestCG

/clusters/cluster-1/consistency-groups/TestCG:

Attributes:
Name                  Value
--------------------  ------------------------------------------------------------
active-clusters       []
cache-mode            asynchronous
detach-rule           active-cluster-wins
operational-status    [(cluster-1,{ summary:: ok, details:: [] }), (cluster-2,{
                      summary:: ok, details:: [] })]
passive-clusters      [cluster-1, cluster-2]
recoverpoint-enabled false
storage-at-clusters   [cluster-1, cluster-2]
virtual-volumes       [TestDDevice-1_vol, TestDDevice-4_vol, TestDDevice-5_vol]
visibility            [cluster-1, cluster-2]


Contexts:
Name          Description
------------  -----------
advanced      -
recoverpoint  -
```

Modify consistency group properties

WARNING

Modifying cache-mode from synchronous to asynchronous when I/O is active is supported but may result in a disruption of access to data during the operation.

Schedule this activity during maintenance windows of low workload to reduce impact on applications and possibility of a disruption.

Modifying cache-mode from asynchronous to synchronous is not supported.

Use the 3 consistency group set-detach rules to modify the "detach-rule" applied to a consistency group:

- **consistency-group set-detach-rule active-cluster-wins**
- **consistency-group set-detach-rule no-automatic-winner**
- **consistency-group set-detach-rule winner**

Use the **set** command to modify the following properties of a consistency group:

All consistency groups:

- "cache mode"
- "visibility"
- "storage-at-clusters"
- "local-read-override"
- "auto-resume-at-loser"

Asynchronous consistency groups only:

- "default closeout-time"
- "maximum-queue-depth"

Synchronous consistency groups only:

- "recoverpoint-enabled"

The syntax for the **set** command is:

```
set
   [-d|--default]
   [-f|--force]
   [-a|--attributes] selector pattern
   [-v|--value] new value
```

[-d|--default] - Sets the specified attribute(s) to the default value(s), if any exist. If no attributes are specified, displays the default values for attributes in the current/specified given context.

[-f|--force] - Force the value to be set, bypassing any confirmations or guards.

[-a|--attributes] ***selector pattern*** - Attribute selector pattern.

[-v|--value] ***new value*** - The new value to assign to the specified attribute(s).

To display which attributes are modifiable (writable) using the **set** command and their valid inputs:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> set
attribute           input-description
```

```
------------------
-------------------------------------------------------------------------------------------------
active-clusters      Read-only.
cache-mode           Takes one of 'asynchronous', 'synchronous' (case sensitive).
detach-rule          Read-only.
name                 Takes a unique, non-empty and non-null name. A valid name starts with a letter or '_'
                     and contains only letters, numbers, '-' and '_'.
operational-status   Read-only.
passive-clusters     Read-only.
recoverpoint-enabled false
storage-at-clusters  Takes a list with each element being a 'cluster' context or a context pattern.
virtual-volumes      Read-only.
visibility           Takes a list with each element being a 'cluster' context or a context pattern.
```

To display the current setting of a property:

```
VPlexcli:/> set /clusters/cluster-1/consistency-groups/TestCG::cache-mode
attribute                                                  input-description
-----------------------------------------------------------
-------------------------------------------------------------
/clusters/cluster-1/consistency-groups/TestCG::cache-mode  Takes one of 'asynchronous',
'synchronous' (case sensitive).
```

To display the default values for the target consistency group:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> set --default
attribute             default-value
--------------------  -----------------
active-clusters       No default value.
cache-mode            No default value.
detach-rule           No default value.
name                  No default value.
operational-status    No default value.
passive-clusters      No default value.
recoverpoint-enabled  No default value.
storage-at-clusters   No default value.
virtual-volumes       No default value.
visibility            No default value.
```

Example of modify: set visibility

To change the visibility property from the target consistency group context:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> set visibility cluster-1,cluster-2
```

To change the visibility property from the consistency group context:

```
VPlexcli:/clusters/cluster-1/consistency-groups> set TestCG::visibility cluster-1,cluster-2
```

To change the visibility property from the root context:

```
VPlexcli:/> set /clusters/cluster-1/consistency-groups/TestCG::visibility cluster-1,cluster-2
```

Example of modify: apply a detach rule

Table 8 lists the applicable detach-rules for consistency groups with various settings for visibility, storage-at-clusters, and cache-mode.

Table 8 Consistency groups detach-rules, visibility, storage-at-volumes, and cache-mode

visibility	storage-at-clusters	cache-mode	Applicable detach-rule setting(s)
cluster-1	cluster-1	synchronous	N/A
cluster-1 and cluster-2	cluster-1 and cluster-2	synchronous	no-automatic-winner winner cluster-1 winner cluster-2
cluster-1 and cluster-2	cluster-1	synchronous	no-automatic-winner winner cluster-1
cluster-1 and cluster-2	cluster-1 and cluster-2	asynchronous	no-automatic-winner active-cluster-wins

To apply a detach rule that will determine the behavior of all volumes in a consistency group:

1. Use the **ll** command to display the current detach rule (if any) applied to the consistency group:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG2> ll
Attributes:
Name                  Value
--------------------  ----------------------
active-clusters       []
cache-mode            synchronous
detach-rule           -
.
.
.
```

2. Use one of the **consistency-group set-detach-rule** commands to apply a detach-rule to the consistency group:

Use the **consistency-group set-detach-rule no-automatic-winner** command to set the detach-rule as no-automatic-winner.

In the following example, the command is used in the target consistency group's context:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> set-detach-rule no-automatic-winner
```

Use the **consistency-group set-detach-rule active-cluster-wins** command to set the detach-rule as active-cluster-wins.

In the following example, the command is used in the consistency group context:

```
VPlexcli:/clusters/cluster-1/consistency-groups> consistency-group set-detach-rule
active-cluster-wins --consistency-groups TestCG
```

Use the **consistency-group set-detach-rule winner** command to specify which cluster is the winner, and the number of seconds VPLEX waits after a link outage before detaching the winning cluster.

In the following example, the command is used in the root context:

```
VPlexcli:/> consistency-group set-detach-rule winner --cluster cluster-1 --delay 5s
--consistency-groups TestCG
```

Example of modify: enable local-read-override

1. Use the **ll** command in the advanced context of the target consistency group to display the current setting:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG2/advanced> ll
Name                          Value
----------------------------  --------
auto-resume-at-loser          false
current-queue-depth           -
current-rollback-data         -
default-closeout-time         -
delta-size                    -
local-read-override           false
.
.
.
```

2. Use the **set** command modify the local-read-override property:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG2/advanced> set local-read -override
true
```

3. Use the **ll** command to confirm the change:

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG2/advanced> ll
Name                          Value
----------------------------  --------
auto-resume-at-loser          false
current-queue-depth           -
current-rollback-data         -
default-closeout-time         -
delta-size                    -
local-read-override           true
.
.
.
```

Example of modify: configure a consistency group for use with RecoverPoint

Consistency groups that will be used to contain RecoverPoint production source or replica volumes must have the following properties:

- "cache mode" is synchronous (default value)
- Consistency groups with the "visibility" property set to both clusters must also have their "storage-at-clusters" set to both clusters
- "recoverpoint-enabled" property set to true.

In the following example, the visibility, storage-at-clusters, and recoverpoint-enabled properties are modified to enable the consistency group to be used with recoverPoint.

```
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> set visibility cluster-1,cluster-2
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> set storage-at-clusters
cluster-1,cluster-2
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG> set recoverpoint-enabled true
VPlexcli:/clusters/cluster-1/consistency-groups/TestCG>ls
Attributes:
Name                  Value
--------------------  ------------------------------------------------------------
active-clusters       []
cache-mode            synchronous
detach-rule           winner cluster-1 after 10s
operational-status    [(cluster-1,{ summary:: ok, details:: [] }), (cluster-2,{
                      summary:: ok, details:: [] })]
```

```
passive-clusters      []
recoverpoint-enabled  true
storage-at-clusters   [cluster-1, cluster-2]
virtual-volumes       []
visibility            [cluster-1, cluster-2]


Contexts:
advanced  recoverpoint
```

Delete a consistency group

WARNING

Before using the consistency-group destroy command on an asynchronous consistency group, ensure that no host applications are using the consistency group.

To destroy an empty consistency group:

1. Use the **ls** command to verify that there are no virtual volumes in the consistency group (virtual volumes = [ ]).

```
VPlexcli:/> ls clusters/cluster-1/consistency-groups/ TestCG

Attributes:
Name                   Value
---------------------  ----------------------
active-clusters        []
cache-mode             synchronous
detach-rule            -
operational-status     [ok]
passive-clusters       []
recoverpoint-enabled   false
storage-at-clusters    [cluster-1, cluster-2]
virtual-volumes        []
visibility             [cluster-1, cluster-2]
.
.
.
```

2. Use the **consistency-group destroy** command to delete the consistency group. The syntax for the command is:

```
consistency-group destroy
[-g|--consistency-group] consistency-group, consistency-group,...
```

[-g | **--consistency-group]** ***consistency-group, consistency-group, ...*** - List of one or more comma-separated context paths of the consistency group(s) to destroy.

To delete a consistency group from the root context:

```
VPlexcli:/> consistency-group destroy clusters/cluster-1/ consistency-groups/TestCG
WARNING: The following items will be destroyed:

Context
------------------------------------------
/clusters/cluster-1/consistency-groups/TestCG

Do you wish to proceed? (Yes/No)Yes
```

To delete a consistency group from the consistency group context:

```
VPlexcli:/clusters/cluster-1/consistency-groups> destroy TestCG
WARNING: The following items will be destroyed:
```

```
Context
------------------------------------------
/clusters/cluster-1/consistency-groups/TestCG

Do you wish to proceed? (Yes/No)Yes
```

Display consistency groups

This section describes the following topics:

For descriptions of the fields in the displays, see ""Display consistency group field descriptions" on page 171

Display consistency group names

Use the **ls /clusters/*/consistency-groups** command to display only the names of consistency groups on all clusters:

```
VPlexcli:/> ls /clusters/*/consistency-groups/

/clusters/cluster-1/consistency-groups:
TestCG          local_test      test10      test11      test12  test13  test14
test15          test16          test5       test6       test7   test8   test9
vs_RAM_c1wins   vs_RAM_c2wins   vs_oban005  vs_sun190


/clusters/cluster-2/consistency-groups:
TestCG          local_test      test10      test11      test12  test13  test14
test15          test16          test5       test6       test7   test8   test9
vs_RAM_c1wins   vs_RAM_c2wins   vs_oban005  vs_sun190
```

Use the **ls /clusters/*cluster-name*/consistency-groups** command to display the names of consistency groups only on the specified cluster:

```
VPlexcli:/> ls /clusters/cluster-1/consistency-groups/

/clusters/cluster-1/consistency-groups:
TestCG      test10      test11  test12  test13  test14  test15  test16  test5  test6  test7
test8  test9  vs_RAM_c1wins  vs_RAM_c2wins
vs_oban005  vs_sun190
```

Display detailed listing of consistency groups

Use the **ll** command in /clusters/*cluster-name*/consistency-groups context to display an overview of the consistency groups.

Use this command to monitor the overall health of consistency groups and identify poorly configured rules:

```
VPlexcli:/clusters/cluster-1/consistency-groups>ll
Name                Operational Status                Active      Passive     Detach Rule          Cache Mode
------------------  --------------------------------  Clusters    Clusters    -------------------  ------------
------------------  --------------------------------  ----------  ----------  -------------------  ------------
D850-008_view1      (cluster-1,{ summary:: ok,    cluster-1   cluster-2   active-cluster-wins  asynchronous
                     details:: [] }),
```

```
                     (cluster-2,{ summary:: ok,
                     details:: [] })
D850-008_view2       (cluster-1,{ summary:: ok,              cluster-1,  active-cluster-wins  asynchronous
                     details:: [] }),                           cluster-2
                     (cluster-2,{ summary:: ok,
                     details:: [] })
RAM_LR_cluster-1     (cluster-1,{ summary:: ok,                        -                    synchronous
                     details:: [] }),
                     (cluster-2,{ summary::
                     unknown, details:: [] })
RAM_RR_cluster-2     (cluster-1,{ summary:: ok,                        no-automatic-winner  synchronous
                     details:: [] }),
                     (cluster-2,{ summary:: ok,
                     details:: [] })
.
.
.
```

Display operational status

Use the **ls** command in /clusters/*cluster-name*/consistency-groups/*consistency-group* context to display the groups' operational status.

In the following example, the command displays the operational status of a consistency group on a healthy VPLEX:

```
VPlexcli:/> ls /clusters/cluster-1/consistency-groups/cg1

/clusters/cluster-1/consistency-groups/cg1:

VPlexcli:/clusters/cluster-1/consistency-groups/cg1> ls

Attributes:
Name                  Value
--------------------  ------------------------------------------------------------
active-clusters       [cluster-1, cluster-2]
cache-mode            asynchronous
detach-rule           no-automatic-winner
operational-status    [(cluster-1,{ summary:: ok, details:: [] }),
                       (cluster-2,{ summary:: ok, details:: [] })]
passive-clusters      []
recoverpoint-enabled  false
storage-at-clusters   [cluster-1, cluster-2]
virtual-volumes       [dd1_vol, dd2_vol]
visibility            [cluster-1, cluster-2]


Contexts:
Name           Description
-------------  -----------
advanced       -
recoverpoint   -
```

Display advanced properties

Display the advanced properties of a specified consistency group:

```
VPlexcli:/clusters/cluster-1/consistency-groups/cg1> ll /advanced

/clusters/cluster-1/consistency-groups/cg1/advanced:
Name                          Value
----------------------------  --------
auto-resume-at-loser          false
current-queue-depth           1
current-rollback-data         0B
default-closeout-time         0.5min
delta-size                    0.25M
local-read-override           true
max-possible-rollback-data    1.25M
```

```
maximum-queue-depth          6
potential-winner             -
write-pacing                 inactive
```

Display recoverpoint properties

Display the RecoverPoint properties of a specified consistency group:

```
VPlexcli:/clusters/cluster-1/consistency-groups/cg1> ll /recoverpoint

/clusters/cluster-1/consistency-groups/cg1/recoverpoint:
Name                              Value
--------------------------------  ------------------------------------------------------------------------
recoverpoint-information          [(device_storage_vol_1_7_vol,,,,),
                                  (device_storage_vol_1_1_vol,CG_1,excedrin,10.6.210.79,Production
                                  Source),
                                  (device_storage_vol_1_2_vol,CG_1,excedrin,10.6.210.79,Production
                                  Source)]
vplex-rp-cg-alignment-indications ['device_storage_vol_1_7_vol' is not in a RecoverPoint consistency
                                  group and is not being protected.]
vplex-rp-cg-alignment-status      error
```

Display operational status - inter-cluster link outage

The following example displays output of the **ls** command in /clusters/*cluster-name*/consistency-groups/*consistency-group* context during an inter-cluster link outage.

- The detach-rule is 'no-automatic-winner', so I/O stops at both clusters. VPLEX remains in this state until either the inter-cluster link restarts, or the user intervenes using the **consistency-group choose-winner** command.
- Status summary is 'suspended', showing that I/O has stopped
- Status details contain 'cluster-departure', indicating that the clusters can no longer communicate with one another.

```
VPlexcli:/clusters/cluster-1/consistency-groups/cg1> ls

Attributes:
Name                  Value
--------------------  ------------------------------------------------------------
active-clusters       [cluster-1, cluster-2]
cache-mode            asynchronous
detach-rule           no-automatic-winner
operational-status    [(cluster-1,{ summary:: suspended, details:: [cluster-departure] }),
                       (cluster-2,{ summary:: suspended, details:: [cluster-departure] })]
passive-clusters      []
recoverpoint-enabled  false
storage-at-clusters   [cluster-1, cluster-2]
virtual-volumes       [dd1_vol, dd2_vol]
visibility            [cluster-1, cluster-2]


Contexts:
advanced  recoverpoint
```

Display operational status - requires resume at loser

- The **ls** command shows consistency group cg1 as 'suspended, requires-resume-at-loser' on cluster-2 after cluster-2 is declared the losing cluster during an inter-cluster link outage.
- The **resume-at-loser** command restarts I/O on cluster-2.
- The **ls** command displays the change in operational status:

```
VPlexcli:/clusters/cluster-1/consistency-groups/cg1> ls

Attributes:
Name                  Value
```

```
------------------  ------------------------------------------------------------
active-clusters     [cluster-1, cluster-2]
cache-mode          asynchronous
detach-rule         no-automatic-winner
operational-status  [(cluster-1,{ summary:: ok, details:: [] }),
                    (cluster-2,{ summary:: suspended, details:: [requires-resume-at-loser] })]
passive-clusters    []
recoverpoint-enabled false
storage-at-clusters [cluster-1, cluster-2]
virtual-volumes     [dd1_vol, dd2_vol]
visibility          [cluster-1, cluster-2]


Contexts:
advanced  recoverpoint

VPlexcli:/clusters/cluster-1/consistency-groups/cg1> resume-at-loser -c cluster-2
This may change the view of data presented to applications at cluster cluster-2. You should
first stop applications at that cluster. Continue? (Yes/No) Yes

VPlexcli:/clusters/cluster-1/consistency-groups/cg1> ls

Attributes:
Name                Value
------------------  ------------------------------------------------------------
active-clusters     [cluster-1, cluster-2]
cache-mode          asynchronous
detach-rule         no-automatic-winner
operational-status  [(cluster-1,{ summary:: ok, details:: [] }),
                     (cluster-2,{ summary:: ok, details:: [] })]
passive-clusters    []
recoverpoint-enabled false
storage-at-clusters [cluster-1, cluster-2]
virtual-volumes     [dd1_vol, dd2_vol]
visibility          [cluster-1, cluster-2]


Contexts:
advanced  recoverpoint
```

Table 9 Display consistency group field descriptions (1 of 5)

Property	Description
Standard properties	
"cache mode"	Whether cache-mode is synchronous or asynchronous. Modifiable using the **set** command. **synchronous** (default) - Writes are done synchronously. Writes are not acknowledged to a host unless they have been sent all the way to back-end storage at all clusters. **asynchronous** - Writes are done asynchronously. Writes are acknowledged when they are received by the cluster that receives them, and the data need not reach back-end storage, or indeed other clusters, until after a delay.
"detach-rule"	Policy for automatically picking a "winning" cluster when there is an inter-cluster link outage. A winning cluster is intended to resume I/O operations when the link fails. For a synchronous consistency group the winning cluster will resume I/O operations shortly after the link fails. For an asynchronous consistency group, the winning cluster prepares to resume I/O operations, but may need to roll back the view of data in the process. If such a roll-back is required, I/O will remain suspended until the administrator manually intervenes with the resume-after-rollback command. Modifiable using the following commands: **consistency-group set-detach-rule active-cluster-wins** - If one cluster was active and one was passive at the time of the link outage, the group selects the active cluster as the winner. **consistency-group set-detach-rule no-automatic-winner** - The consistency group will not select a winning cluster. **consistency-group set-detach-rule winner** - The cluster specified by *cluster-name* will be declared the winner if an inter-cluster link outage lasts more than the number of seconds specified by *delay*.
"recoverpoint-enabled"	Whether this consistency group can be used with RecoverPoint. Modifiable using the **set** command. **true** - The volumes in the consistency group can be used with RecoverPoint. **false** (default) - Integration with RecoverPoint is not enabled for volumes in the consistency group.
"storage-at-clusters"	At which cluster the physical storage associated with a consistency group is located. Modifiable using the **set** command. If cluster names are "cluster-1" and "cluster-2" valid values are: **cluster-1** - Storage associated with this consistency group is located only at cluster-1. **cluster-2** - Storage associated with this consistency group is located only at cluster-2. **cluster-1,cluster-2** - Storage associated with this consistency group is located at both cluster-1 and cluster-2. When modified, the new value cannot be incompatible with the volumes that are already in the consistency group. Change storage-at-clusters only when the consistency group has no member volumes.
"visibility"	Lists the clusters at which this consistency group is visible. Modifiable using the **set** command. If cluster names are "cluster-1" and "cluster-2" valid values are: **cluster-1** - This consistency group is visible only at cluster-1. **cluster-2** - This consistency group is visible only at cluster-2. **cluster-1,cluster-2** - This consistency group is visible at both cluster-1 and cluster-2. Changing this property changes where the consistency group is visible, and may cause contexts to appear or disappear in the context tree.
"virtual-volumes"	Lists the virtual volumes that are members of the consistency group. Modifiable using the following commands: **consistency-group add-virtual-volumes** - Add one or more virtual volumes to a consistency group. **consistency-group remove-virtual-volumes** - Remove one or more virtual volumes from a consistency group.

Table 9 Display consistency group field descriptions (2 of 5)

Property	Description
Advanced properties	
"auto-resume-at-loser"	Determines whether IO automatically resumes at the detached cluster for the volumes in a consistency group when the cluster regains connectivity with its peer cluster. Relevant only for multi-cluster consistency groups that contain distributed volumes. Modifiable using the **set** command. Set this property to true to allow the volumes to resume I/O without user intervention (using the **resume-at-loser** command). **true** - I/O automatically resumes on the losing cluster after the inter-cluster link has been restored. **false** (default) - I/O must be resumed manually after the inter-cluster link has been restored. **Note:** Leave this property set to false to give administrators time to restart the application. Otherwise, dirty data in the host's cache is not consistent with the image on disk to which the winning cluster has been actively writing. Setting this property to true can cause a spontaneous change of the view of data presented to applications at the losing cluster. Most applications cannot tolerate this data change. If the host flushes those dirty pages out of sequence, the data image may be corrupted.
"default closeout-time"	Applicable only to asynchronous consistency groups. Modifiable using the **set** command. Maximum time for a delta to remain in the open state. When this value is positive, the firmware will attempt to close the open delta once it has been open for default-closeout-time seconds. When this value is zero, there is no time limit on closing the open delta. The delta is closed when it cannot accept more dirty data in the delta. Use this property to limit the maximum possible data loss in the event of an inter-cluster link outage. The ideal default close-out -time is equal to the time it takes to exchange a full delta set. **Default**: 30 seconds. **zero (0)** - There is no time limit on closing the open delta. The delta is closed when it is full and cannot accept more data.
"local-read-override"	Applicable only to asynchronous consistency groups. Modifiable using the **set** command. Instructs VPLEX whether to prefer reading non-dirty pages from cache or from back-end storage. When a read request is received by a director it will first check with the distributed cache component to see if that page is dirty (written to some director's cache, but not yet written to disk). If the page is dirty in any director's cache, the page is sent from that director to the reading director. This is the case when the two directors are in the same cluster *and* when the two directors are in different clusters. When a read request is received by a director, and the page is not dirty in any director's cache, and none of the directors in the same cluster have that page in cache, it has two ways to get the page: • Query the directors in the other cluster whether they have the page in their caches, or • Read the page from the underlying back-end storage **Note:** If none of the directors have the page in cache, the page is read from the underlying back-end storage. This property instructs VPLEX whether to prefer reading the page from directors in the other cluster, or to prefer reading the page from the back-end storage. **false** - The distributed cache will prefer to pull the page across the inter cluster link from the remote director's cache. **true** - (default) The distributed cache will prefer to read the page from the underlying back-end storage.
"maximum-queue-depth"	Applicable only to asynchronous consistency groups. Modifiable using the **set** command. Configures the maximum possible length of the delta processing queue. **Default**: 6 **Range**: 6 - 64. Maximum value is platform specific.

Table 9 Display consistency group field descriptions (3 of 5)

Property	Description
Recoverpoint properties	
recoverpoint-information	
vplex-rp-cg-alignment-indications	
vplex-rp-cg-alignment status	
Display-only properties	
active-clusters	List of clusters which have recently written to a member volume in the consistency group. Applicable only to asynchronous consistency groups. For synchronous consistency groups, this property is always empty ([ ]).
current-rollback-data	Applicable only to asynchronous consistency groups. Amount of data that would be discarded if that cluster were declared the winner during a link outage and rolled back its view of data to the last known consistent point. This includes all the data in the queue that has not yet been exchanged with the other cluster. This value depends on the current state of the system and changes over time.
current-queue-depth	Applicable only to asynchronous consistency groups. Number of deltas or stages in the asynchronous data queue at the enclosing cluster.
delta-size	Applicable only to asynchronous consistency groups. The maximum permitted size of a delta. Exception: the maximum-queue-size is reached, because I/O is being received faster than it can be exchanged between clusters and committed to back-end storage. The current open delta may exceed this size until the delta at the head of the queue has been committed.
max-possible-rollback-data	An estimate of the maximum amount data that can be lost if at a single cluster within a volume set if there are multiple director failures within the cluster, a total cluster failure, or a total system failure. Roughly the product of current-queue-depth, delta-size, and the largest number of directors at any one cluster.
operational status	Current status for this consistency group with respect to each cluster on which it is visible. **ok** - I/O can be serviced on the volumes in the consistency group. **suspended** - I/O is suspended for the volumes in the consistency group. The reasons are described in the "operational status: details". **degraded** - I/O is continuing, but there are other problems described in "operational status: details": **unknown** - The status is unknown, likely because of lost management connectivity.

Table 9 Display consistency group field descriptions (4 of 5)

Property	Description
operational status: details	If operational status is “ok” this field is empty: “[ ]”. Otherwise, it displays additional information, which may be any of the following: **cluster-departure** - Not all the visible clusters are in communication. **data-safe-failure** - A single director has failed. The volumes are still crash-consistent, and will remain so, unless a second failure occurs before the first is recovered. **member-volumes-unreachable** - This asynchronous consistency group has unreachable volumes. I/O is not suspended but the consistency group is degraded. **rebuilding-across-clusters** - One or more distributed member volumes is being rebuilt. At least one volume in the group is out of date at that cluster and is re-syncing. If the link goes out at this time the entire group is suspended. Use the rebuild status command to display which volume is out of date at which cluster. **rebuilding-within-cluster** - One or more local rebuilds is in progress at this cluster. **requires-resolve-conflicting-detach** - After the inter-cluster link is restored, two clusters have discovered that they have detached one another and resumed I/O independently. The clusters are continuing to service I/O on their independent versions of the data. The consistency-group resolve-conflicting-detach command must be used to make the view of data consistent again at the clusters. **requires-resume-after-data-loss-failure** - There have been at least two concurrent failures, and data has been lost. For example, a director fails shortly after the inter-cluster link fails, or when two directors fail at almost the same time. Use the consistency-group resume-after-data-loss-failure command to select a winning cluster and allow I/O to resume. **requires-resume-after-rollback** - A cluster has detached its peer cluster and rolled back the view of data, but is awaiting the consistency-group resume-after-rollback command before resuming I/O. Displayed: • For an asynchronous consistency group where both clusters have been writing, that is where both clusters are active. • At the winning side when a detach rule fires, or shortly after the consistency-group choose-winner command picks a winning cluster. **requires-resume-at-loser** - Displayed on the losing side when the inter-cluster link heals after an outage. After the inter-cluster link is restored, the losing cluster discovers that its peer was declared the winner and resumed I/O. Use the consistency-group resume-at-loser command to make the view of data consistent with the winner, and to resume I/O at the loser. **restore-link-or-choose-winner** - I/O is suspended at all clusters because of a cluster departure, and cannot automatically resume. This can happen if: • There is no detach-rule • If the detach-rule is 'no-automatic-winner', or • If the detach-rule cannot fire because its conditions are not met. For example, if more than one cluster is active at the time of an inter-cluster link outage, the 'active-cluster-wins' rule cannot take effect. When this detail is present, I/O will not resume until either the inter-cluster link is restored, or the user intervenes to select a winning cluster with the consistency-group choose-winner command. **temporarily-synchronous** - This asynchronous consistency group has temporarily switched to 'synchronous' cache mode because of a director failure. **unhealthy-devices** - I/O has stopped in this consistency group because one or more volumes is unhealthy and cannot perform I/O. **will-rollback-on-link-down** - If there were a link-down now, the winning cluster would have to roll back the view of data in order to resume I/O.
passive-clusters	Applicable only to asynchronous consistency groups. Clusters which have not recently written to a member volume. For synchronous consistency groups, the value of this attribute is always empty [ ].

Table 9 Display consistency group field descriptions (5 of 5)

Property	Description
potential-winner	Applicable only to asynchronous consistency groups. For an asynchronous consistency group on which the detach-rule is “active-cluster-wins”, shows which cluster, if any, would be declared the winning cluster if there were an inter-cluster link outage now.
virtual-volumes	List of the virtual volumes that are members of this consistency group.
write-pacing	Applicable only to asynchronous consistency groups. I/O from the host slows down to the consistency group when inter-cluster I/O approaches its upper limit. **inactive** - Write pacing it is not currently being applied to I/O. **active** - Write pacing is being applied to I/O. Activated when the underlying consistency group utilization reaches 70%.

Operate a consistency group

This section describes the procedures to:

- "Resolve a conflicting detach" on page 176
- "Resume I/O after rollback" on page 177
- "Resume I/O at the losing cluster" on page 179

Resolve a conflicting detach

Best practice is to allow I/O to continue at only one cluster. Allowing I/O to continue at both clusters will result in the complete re-syncing of one cluster from the other. All writes at the loser cluster will be lost.

When I/O continues at both clusters:

- The data images at the clusters diverge.
- Legs of distributed volumes are logically separate.

When the inter-cluster link is restored, the clusters learn that I/O has proceeded independently. I/O continues at both clusters until the administrator picks a "winning" cluster whose data image will be used as the source to resynchronize the data images.

In the following example, I/O has resumed at both clusters during an inter-cluster link outage. When the inter-cluster link is restored, the two clusters will come back into contact and learn that they have each detached the other and carried on I/O.

1. Use the **ls** command to display the consistency group's operational status at both clusters.

```
VPlexcli:/clusters/cluster-1/consistency-groups/cg1> ls

Attributes:
Name                  Value
--------------------  ----------------------------------------
active-clusters       [cluster-1, cluster-2]
cache-mode            asynchronous
detach-rule           no-automatic-winner
operational-status    [(cluster-1,{ summary:: ok, details::
[requires-resolve-conflicting-detach] }),
                      (cluster-2,{ summary:: ok, details::
[requires-resolve-conflicting-detach] })]
passive-clusters      []
recoverpoint-enabled false
storage-at-clusters   [cluster-1, cluster-2]
virtual-volumes       [dd1_vol, dd2_vol]
visibility            [cluster-1, cluster-2]


Contexts:
advanced  recoverpoint
```

2. Use the **resolve-conflicting-detach** command to select cluster-1 as the winner. The syntax for the command is:

```
consistency-group resolve-conflicting-detach
    [-c|--cluster] cluster
    [-g|--consistency-group context path
    [-f|--force]
```

[-c | --cluster] ***cluster*** - * The cluster whose data image will be used as the source to resynchronize the data images on both clusters.

[-g | --consistency-group] ***consistency-group*** - * The consistency group on which to resolve the conflicting detach.

```
VPlexcli:/clusters/cluster-1/consistency-groups/cg1> resolve-conflicting-detach -c cluster-1
This will cause I/O to suspend at clusters in conflict with cluster cluster-1, allowing you to
stop applications at those clusters. Continue? (Yes/No) Yes
```

Cluster-2's modifications to data on volumes in the consistency group since the link outage started are discarded.

Cluster-2's data image is then synchronized with the image at cluster-1.

I/O will suspend at cluster-2 if the auto-resume policy is false.

3. Use the **ls** command to verify the change in operation status:

```
VPlexcli:/clusters/cluster-1/consistency-groups/cg1> ls

Attributes:
Name                  Value
--------------------  ------------------------------------------------------------
active-clusters       [cluster-1, cluster-2]
cache-mode            asynchronous
detach-rule           no-automatic-winner
operational-status    [(cluster-1,{ summary:: ok, details:: [] }),
                     (cluster-2,{ summary:: suspended, details:: [requires-resume-at-loser] })]
passive-clusters      []
recoverpoint-enabled false
storage-at-clusters   [cluster-1, cluster-2]
virtual-volumes       [dd1_vol, dd2_vol]
visibility            [cluster-1, cluster-2]


Contexts:
advanced  recoverpoint
```

- At cluster-1, I/O continues, and the status is 'ok'.
- At cluster-2, the view of data has changed and so I/O is suspended.

4. Use the consistency-group resume-at-loser command to resume I/O to the consistency group on cluster-2.

Resume I/O after rollback

WARNING

In a Geo configuration, on a cluster that successfully vaulted and unvaulted, the user should contact EMC Engineering for assistance before rolling back the data prior to re-establishing communication with the non-vaulting cluster.

If the losing cluster is active at the onset of a cluster or inter-cluster-link outage, the distributed cache at the losing cluster contains dirty data.

Without that data, the winning cluster's data image is inconsistent. Resuming I/O at the winner requires rolling back the winner's data image to the last point where the clusters agreed.

This can cause a sudden change in the data image.

Many applications cannot tolerate sudden data changes, so the roll-back and resumption of I/O requires manual intervention.

The delay gives the administrator the chance to halt applications before changing the data image. The data image is rolled back as soon as a winner is chosen (either manually or automatically using a detach rule).

The **resume-after-rollback** command acknowledges that the application is ready for recovery (this may involve application failure and/or rebooting the host).

Note: It is recommended to reboot the hosts of affected applications.

1. Use the **ls** command to display the consistency group on the winning cluster during an inter-cluster link outage.

   Because the consistency group is asynchronous, I/O remains suspended.

```
VPlexcli:/clusters/cluster-1/consistency-groups/cg1> ls

Attributes:
Name                   Value
---------------------  ------------------------------------------
active-clusters        []
cache-mode             asynchronous
detach-rule            -
operational-status     [suspended, requires-resume-after-rollback]
passive-clusters       [cluster-1, cluster-2]
recoverpoint-enabled   false
storage-at-clusters    [cluster-1, cluster-2]
virtual-volumes        [dd1_vol]
visibility             [cluster-1, cluster-2

Contexts:
advanced  recoverpoint
```

2. Use the **resume-after-rollback** command to acknowledge that the application is ready for recovery.

```
VPlexcli:/clusters/cluster-1/consistency-groups/cg1> resume-after-rollback
--consistency-group cg1
This will change the view of data at cluster cluster-1, so you should ensure applications are
stopped at that cluster. Continue? (Yes/No) Yes
```

3. Use the **ls** command to display the change in operational status.

```
VPlexcli:/clusters/cluster-1/consistency-groups/cg1> ls

Attributes:
Name                   Value
---------------------  ----------------------
active-clusters        [cluster-1]
cache-mode             asynchronous
detach-rule            -
operational-status     [ok]
passive-clusters       [cluster-2]
recoverpoint-enabled   false
storage-at-clusters    [cluster-1, cluster-2]
virtual-volumes        [dd1_vol]
visibility             [cluster-1, cluster-2]

Contexts:
advanced  recoverpoint
```

Resume I/O at the losing cluster

During an inter-cluster link outage, an administrator may permit I/O to resume at one of the two clusters: the "winning" cluster.

I/O remains suspended on the "losing" cluster.

When the inter-cluster link heals, the winning and losing clusters re-connect, and the losing cluster discovers that the winning cluster has resumed I/O without it.

Unless explicitly configured otherwise (using the "auto-resume-at-loser" property), I/O remains suspended on the losing cluster. This prevents applications at the losing cluster from experiencing a spontaneous data change.

The delay allows the administrator to shut down applications.

After stopping the applications, the administrator can use the consistency-group resume-at-loser command to:

- Resynchronize the data image on the losing cluster with the data image on the winning cluster,
- Resume servicing I/O operations.

The administrator may then safely restart the applications at the losing cluster.

To restart I/O on the losing cluster:

1. Use the **ls** command to display the operational status of the target consistency group.

```
VPlexcli:/clusters/cluster-1/consistency-groups/cg1> ls

Attributes:
Name                  Value
--------------------  ------------------------------------------------------------
active-clusters       [cluster-1, cluster-2]
cache-mode            asynchronous
detach-rule           no-automatic-winner
operational-status    [(cluster-1,{ summary:: ok, details:: [] }),
                      (cluster-2,{ summary:: suspended, details:: [requires-resume-at-loser] })]
passive-clusters      []
recoverpoint-enabled  false
storage-at-clusters   [cluster-1, cluster-2]
virtual-volumes       [dd1_vol, dd2_vol]
visibility            [cluster-1, cluster-2]


Contexts:
advanced  recoverpoint
```

2. Use the consistency-group resume-at-loser to restart I/O on the losing cluster. The syntax for the command is:

```
consistency-group resume-at-loser
[-c|--cluster] cluster
[-s|--consistency-group] consistency-group
[-f|--force]
```

[-c|--cluster] ***cluster*** - The cluster on which to roll back and resume I/O.

[-g|--consistency-group] ***consistency-group*** - The consistency group on which to resynchronize and resume I/O.

```
VPlexcli:/clusters/cluster-1/consistency-groups/cg1> resume-at-loser -c cluster-2
This may change the view of data presented to applications at cluster cluster-2. You should
first stop applications at that cluster. Continue? (Yes/No) Yes
```

3. Use the **ls** command to verify the change in operational status:

```
VPlexcli:/clusters/cluster-1/consistency-groups/cg1> ls

Attributes:
Name                  Value
--------------------  ------------------------------------------------------------
active-clusters       [cluster-1, cluster-2]
cache-mode            asynchronous
detach-rule           no-automatic-winner
operational-status    [(cluster-1,{ summary:: ok, details:: [] }),
                       (cluster-2,{ summary:: ok, details:: [] })]
passive-clusters      []
recoverpoint-enabled false
storage-at-clusters   [cluster-1, cluster-2]
virtual-volumes       [dd1_vol, dd2_vol]
visibility            [cluster-1, cluster-2]


Contexts:
advanced  recoverpoint
```

10

VPLEX Witness

This chapter describes VPLEX Witness, including:

Introduction

Starting in GeoSynchrony 5.0, VPLEX Witness helps multi-cluster VPLEX configurations automate the response to cluster failures and inter-cluster link outages.

VPLEX Witness is an optional component installed as a VM on a customer host. The customer host must be deployed in a separate failure domain from either VPLEX cluster to eliminate the possibility of a single fault affecting both a cluster and VPLEX Witness.

VPLEX Witness connects to both VPLEX clusters over the management IP network:

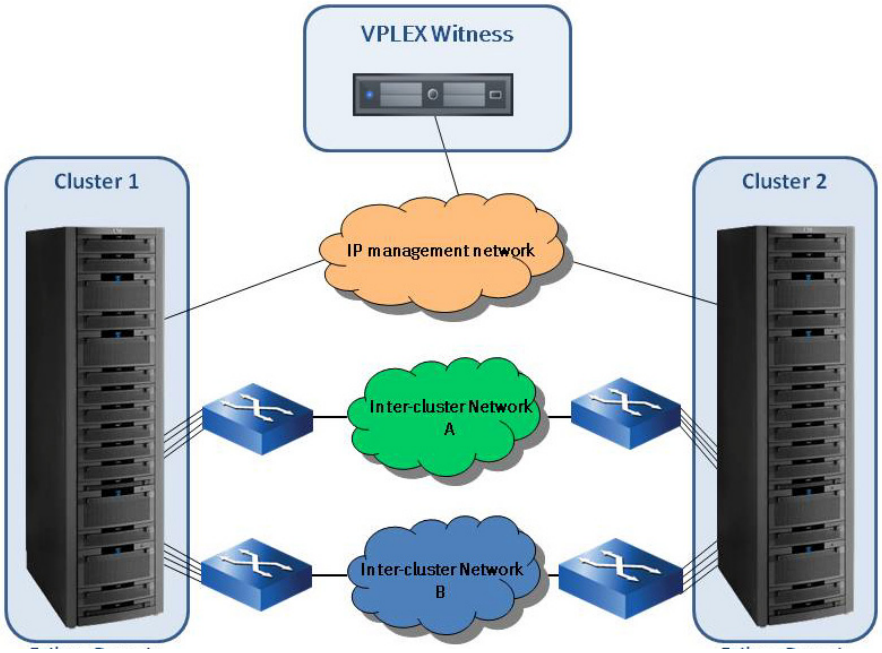


Figure 10 How VPLEX Witness is deployed

VPLEX Witness observes the state of the clusters, and thus can distinguish between a outage of the inter-cluster link and a cluster failure. VPLEX Witness uses this information to guide the clusters to either resume or suspend I/O.

Note: VPLEX Witness works in conjunction with consistency groups (see "Consistency groups" on page 139). VPLEX Witness guidance does not apply to local volumes and distributed volumes that are not members of a consistency group.

VPLEX Witness capabilities vary depending on whether the VPLEX is a Metro (synchronous consistency groups) or Geo (asynchronous consistency groups).

- In Metro systems, VPLEX Witness provides seamless zero RTO fail-over for storage volumes in synchronous consistency groups. See "Failures in Metro Systems: With VPLEX Witness" on page 186.
- In Geo systems, VPLEX Witness does not automate fail-over for asynchronous consistency groups and can only be used for diagnostic purposes.

Related Information

Additional information, including procedures to install, configure, enable, and disable VPLEX Witness is published in the following EMC documents:

Information	Reference
Commands to enable/disable/display VPLEX Witness	*EMC VPLEX CLI Guide*
Consistency groups	Chapter 9, "Consistency groups,"
Procedures to install/configure VPLEX Witness	EMC VPLEX generator
Procedures to troubleshoot VPLEX Witness	EMC VPLEX generator

Handling failures when VPLEX Witness is not deployed: detach rules

When VPLEX Witness is not deployed, detach rules determine at which cluster I/O continues during a cluster failure or inter-cluster link outage.

Note: The cluster on which I/O continues during a failure is called the *preferred cluster*.

VPLEX uses two levels of detach rules to determine which cluster is preferred: detach rules associated with individual distributed devices, and detach rules associated with consistency groups.

Detach rules associated with distributed devices - Every distributed device must have a detach rule. There are two device-level detach rules:

- **cluster-1-detaches**

  Cluster-1 is the preferred cluster. During a cluster failure or inter-cluster link outage, cluster-1 continues I/O and cluster-2 suspends I/O.

- **cluster-2-detaches**

  Cluster-2 is the preferred cluster. During a cluster failure or inter-cluster link outage, cluster-2 continues I/O and cluster-1 suspends I/O.

VPLEX Witness does not override device-level detach rules.

Detach rules associated with consistency groups - Every consistency group has a detach rule that applies to all member devices in the consistency group. If a distributed device is a member of a consistency group, the detach rule of the consistency group overrides the detach rule configured for the device.

There are three consistency group detach rules.

- **active-cluster-wins**

  This detach rule applies only to asynchronous consistency groups (VPLEX Geo configurations).

  If one cluster was active and one was passive at the time of the failure, the active cluster is preferred. See "When a cluster is active vs. passive" on page 153.

  The consistency group's "auto-resume-at-loser" property must be set to true for the active-cluster-wins rule to work.

- **winner *cluster-name* delay *seconds***

  This detach rule applies only to synchronous consistency groups (VPLEX Metro configurations).

The cluster specified by *cluster-name* is declared the preferred cluster if a failure lasts more than the number of seconds specified by *seconds*.

- **no-automatic-winner**

  Consistency groups with this detach rules are not guided by VPLEX Witness.

  The consistency group does not select the preferred cluster. The detach rules of the member devices determine the preferred cluster for that device.

  Manual intervention may be required to restart I/O on the suspended cluster.

Failures in Metro systems

VPLEX Metro configurations (synchronous consistency groups) have two consistency group level detach rules:

- "winner cluster-name delay seconds"
- "no-automatic-winner"

VPLEX Witness does not guide consistency groups with the "no-automatic-winner" detach rule. The remainder of this discussion applies only to synchronous consistency groups with the "winner cluster-name delay seconds" detach rule.

Synchronous consistency groups use write-through caching. Host writes to a distributed volume are acknowledged back to the host only after the data is written to the back-end storage at both VPLEX clusters.

IMPORTANT

VPLEX Witness does not automate fail-over for distributed volumes outside of consistency groups.

Failures in Metro Systems: Without VPLEX Witness

In Figure 11, the "winner cluster-name delay seconds" detach rule designates cluster-1 as the preferred cluster. That is, during an inter-cluster link outage or a cluster failure, I/O to the device leg at cluster-1 continues, and I/O to the device leg at cluster-2 is suspended.

Three common types of failures that illustrate how VPLEX responds without VPLEX Witness are described below:.

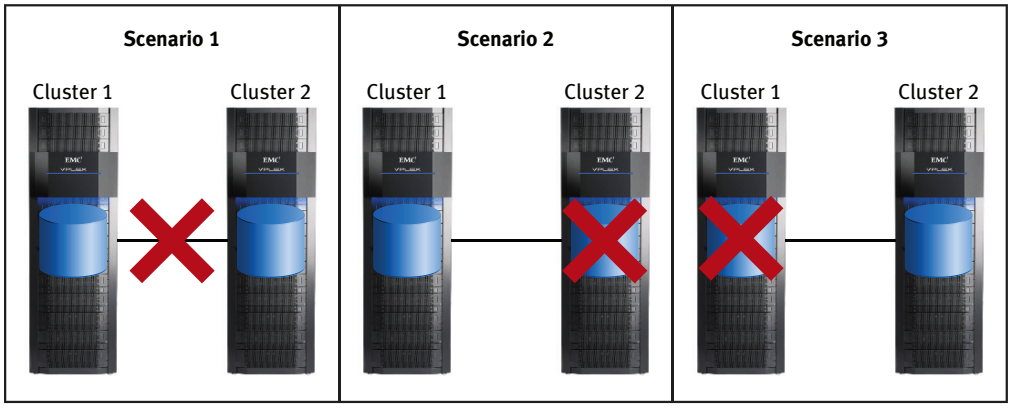


Figure 11 Failure scenarios in VPLEX Metro configurations without VPLEX Witness

Scenario 1 - Inter-cluster link outage. Both of the dual links between the clusters have an outage. Also known as a "cluster partition".

- The preferred cluster (cluster-1) continues I/O
- Cluster-2 suspends I/O

The existing detach rules are sufficient to prevent data unavailability. Writes at cluster-1 are logged. When the inter-cluster link is restored, a log rebuild copies only the logged changes to resynchronize the clusters.

Scenario 2 - Cluster-2 fails.

- Cluster-1 (the preferred cluster) continues I/O.

The existing detach rules are sufficient to prevent data unavailability. Volumes are accessible with no disruptions at cluster-1.

Writes at cluster-1 are logged. When cluster-2 is restored, and rejoins cluster-1, a log rebuild copies only the logged changes to resynchronize cluster-2.

Scenario 3 - Cluster-1 (the preferred cluster) fails.

- Cluster-2 suspends I/O (data unavailability)

VPLEX cannot automatically recover from this failure and suspends I/O at the only operating cluster.

Recovery may require manual intervention to re-enable I/O on cluster-2. In this case, however, it is crucial to note that if I/O is enabled on cluster-2 while cluster-1 is active and processing I/O, data corruption will ensue once the two clusters are synchronized.

VPLEX Witness addresses Scenario 3, where the preferred cluster fails, and the un-preferred cluster cannot continue I/O due to the configured detach rule-set.

Failures in Metro Systems: With VPLEX Witness

When VPLEX Witness is deployed in a VPLEX Metro configuration, failure of the preferred cluster (Scenario 3) does not result in data unavailability for distributed devices that are members of (synchronous) consistency groups.

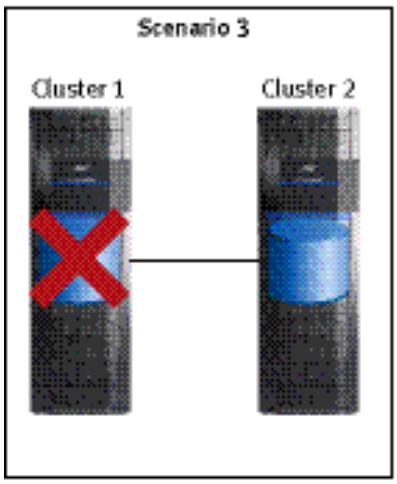


Figure 12 VPLEX Metro - preferred cluster fails

Instead, VPLEX Witness guides the surviving cluster to continue I/O, despite it's designation as the non-preferred cluster. I/O continues to all distributed volumes in all synchronous consistency groups that do not have the "no-automatic-winner" detach rule.

Host applications continue I/O on the surviving cluster without any manual intervention.

When the preferred cluster fails in a Metro configuration, VPLEX Witness provides seamless zero RTO fail-over to the surviving cluster.

VPLEX Witness failures

The VPLEX Witness Server VM connects to the VPLEX clusters over the management IP network. The deployment of VPLEX Witness adds a point of failure to the VPLEX deployment.

This section describes the impact of failures of the VPLEX Witness Server VM and the network connections between the VM and the clusters.

Loss of Contact with VPLEX Witness - The clusters are still in contact with each other, but one of the clusters (either preferred or non-preferred) has lost contact with VPLEX Witness. This can occur when there is a management connectivity failure between the VPLEX Witness host and the Management Server in the corresponding cluster. In this scenario:

- There is no change in I/O.
- The cluster that lost connectivity with VPLEX Witness sends a call-home notification.

Note that this scenario, depicted in Figure 13, is equivalent to a dual connection failure between each cluster and VPLEX Witness. This scenario may occur either as a result of dual connection failure or the physical failure of the host on which VPLEX Witness is deployed.

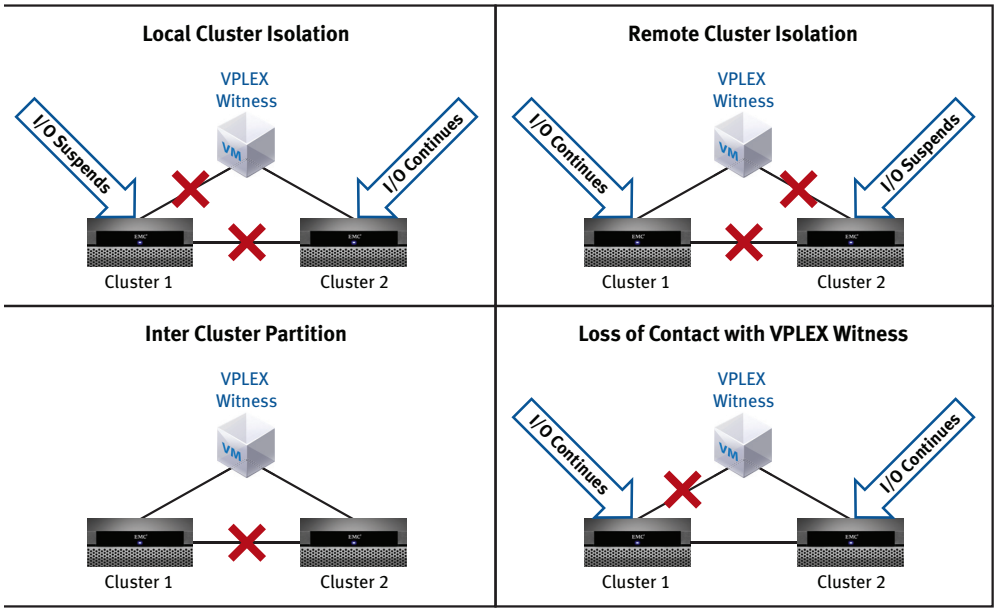


Figure 13 VPLEX Witness to VPLEX cluster connection failures

Connection failures relating to VPLEX Witness are managed as follows:

Local Cluster Isolation - The preferred cluster loses contact with both the remote cluster and VPLEX Witness.

- The preferred cluster cannot receive guidance from VPLEX Witness and suspends I/O.
- VPLEX Witness guides the non-preferred cluster to continue I/O.

Remote Cluster Isolation - The preferred cluster loses contact with the remote cluster and the non-preferred cluster loses contact with the VPLEX Witness. The preferred cluster is connected to VPLEX Witness.

- The preferred cluster continues I/O as it is still in contact with the VPLEX Witness.
- The non-preferred cluster suspends I/O, as it is neither in contact with the other cluster, nor can it receive guidance from VPLEX Witness.

Inter-Cluster Partition - Both clusters lose contact with each other, but still have access to the VPLEX Witness. VPLEX Witness preserves the detach rule failure behaviors:

- I/O continues on the preferred cluster.
- If the preferred cluster can not proceed because it has not fully synchronized, the cluster suspends I/O.

Failures in Geo systems

In VPLEX Geo configurations (asynchronous consistency groups) the two detach rules that apply are:

- "active-cluster-wins"
- "no-automatic-winner"

This discussion applies only to asynchronous consistency groups with the "active-cluster-wins" detach rule.

Failures in VPLEX Geo systems

Failures behaviors in VPLEX Geo configurations vary depending on whether I/O was occurring at one or both legs of a distributed device when the failure occurred.

This section describes a selection of failures when I/O is occurring at only one leg (active/passive) and both legs (active/active). See "When a cluster is active vs. passive" on page 153.

Active/passive

In an active/passive configuration, hosts at only one cluster write to legs of distributed volumes located at both clusters. Failures in active/passive scenarios are handled as follows:

- **Entire engine on the passive cluster fails** - No loss of data or access. If either director in the failed engine had deltas in "commit" or "exchange", part of the delta is lost at the passive cluster. These deltas are re-copied across the inter-cluster link from the active cluster.
- **Entire engine on the active cluster fails** - All data in open, closed, and exchanging deltas is discarded. I/O is suspended at both clusters. Manual intervention is required (**resume-after-data-loss-failure** command) to roll back the data image to the last committed delta (the last consistent point in time).
- **Passive cluster fails** - No loss of data or access. I/O continues on the active cluster. All writes are logged. When the passive cluster is restored, and rejoins, a log rebuild copies only the logged changes to resynchronize the passive cluster.
- **Active cluster fails** - Dirty data is lost, and the volumes are suspended at the passive cluster. Hosts at the active cluster experience a DU.

  All writes at the passive cluster are logged. When the active cluster is restored and rejoins VPLEX, a log rebuild copies only changes to the active cluster.

  Users can fail-over applications to the passive cluster using the **choose-winner** and **resume-after-rollback** commands.

  Alternatively, users can wait for the failed cluster to be restored. When the failed cluster is restored and rejoins, VPLEX recognizes that the failed cluster has lost data and rolls back to the last committed delta (last time consistent image). The user must manually re-enable I/O at the recovered cluster (**resume-at-loser** command).
- **Inter-cluster link outage** - No loss of data or access. I/O continues on the active cluster. All I/O is logged. On the passive cluster, local volumes are accessible to local hosts. When the inter-cluster link is restored, a log rebuild copies only the logged changes to resynchronize the passive cluster.

Active/active

In an active/active configuration, hosts at both clusters write to legs of distributed volumes at both clusters. Failures in active/active scenarios are handled as follows:

- **Entire engine on the either cluster fails** - All data in open, closed, and exchanging deltas is discarded. I/O is suspended at both clusters. Manual intervention (**resume-after-data-loss-failure** command), is required to roll back the data image to the last committed delta, which is the last consistent point in time.
- **Either cluster fails** - All data in open, closed, and exchanging deltas is discarded. I/O to volumes is suspended at both clusters. All writes are logged at the surviving cluster. When the failed cluster is restored and rejoins, a log rebuild copies only changes to the restored cluster.

  Users can fail-over applications to the surviving cluster using the **choose-winner** and **resume-after-rollback** commands. VPLEX rolls back the data image to the end of the last committed delta (last consistent point in time). Applications must be restarted or rolled back.

  Users can wait for the failed cluster to be restored. When the failed cluster is restored and rejoins, VPLEX recognizes that the failed cluster has lost data and rolls back to the last committed delta (last time consistent image). The user must manually re-enable I/O at the recovered cluster (**resume-at-loser** command).
- **Inter-cluster link outage** - I/O suspended at both clusters. I/O at both clusters is logged.

  Users can wait for the network outage to resolve. When the inter-cluster link is restored, a log rebuild copies only the logged changes to resynchronize the clusters.

  Alternatively, users can manually choose a cluster to continue I/O. Manual intervention (**resume-after-data-loss-failure** command) rolls back the data image to the last committed delta, which is the last consistent point in time. Applications are then re-started.

Multiple failures in VPLEX Geo systems

During recovery, a failed director's dirty cache is recovered from its protection partner and re-protected. If portions of committed and exchanged deltas have been lost, they are retrieved from the other cluster.

Exchanging deltas are not replicated within the cluster. Thus, a cluster failure can cause a loss of parts of deltas at that cluster. If the surviving cluster remains healthy, a complete copy of the deltas are at the other cluster.

These recovery operations create scenarios where an additional failure can result in data loss or data unavailability.

This section describes how multiple failures are handled.

- **Failed director's protection partner also fails** - If the failed director's protection partner fails before dirty cache recovery completes, VPLEX has lost data.

  This is both a DU and a DL.

  Recovery from an external backup is required.
- **Inter-cluster link outage during director failure recovery** - A single director failure followed by a inter-cluster link outage before failure recovery of the missing portions of the commit delta are copied.

  The cluster with the failed director did not complete committing the delta and the image on backend storage is not a time consistent image.

If a detach rule or user command causes the unhealthy cluster to become a winner in this state, the affected consistency groups stay suspended at that cluster.

If the unhealthy cluster (the one at which the original failure occurred) is not detached, the user may:

- Wait for the network to be restored.

  No data is lost.

- Accept a data rollback and resume I/O at the other, healthy cluster.

- **Multiple BE storage volume failures after a inter-cluster link outage** - This scenario occurs when a cluster is declared preferred during a inter-cluster link outage. When the link recovers, the preferred cluster updates the other cluster (logging rebuild). If the target disks on the cluster being updated fail, the consistency group suspends.

  Recovery from an external backup is required.

Data loss failure mode (DLFM)

In some extremely rare system-wide multi-failure scenarios the system may enter Data Loss Failure Mode, including data loss and potential loss of crash consistency on disk in both clusters.

This state applies to individual asynchronous consistency groups; that is, while one asynchronous consistency group may be in Data Loss Failure Mode, others on the same VPLEX Geo system may be unaffected.

Best practices to avoid DLFM

To avoid DLFM, follow these best practices when configuring your system:

- Mirror local DR1 legs to two local arrays with a RAID-1. Local mirrors minimize the risk of back-end visibility issues during array failures
- Follow best practices for high availability when deploying VPLEX Geo configurations. Refer to the *VPLEX Product Guide.*

Recover from DLFM

Refer to the troubleshooting procedure for recovering from DLFM in the VPLEX generator.

Failures in Geo Systems: With VPLEX Witness

The value of VPLEX Witness is different in VPLEX Geo configurations than it is in VPLEX Metro configurations. In Geo configurations, VPLEX Witness does not automate recovery from any failure scenario.

Clusters in Geo configurations do not comply with VPLEX Witness guidance. Rather, the clusters operate according to the detach rules applied to each asynchronous consistency group.

Information displayed in the VPLEX Witness CLI context can help determine the nature of a failure (for example, a cluster failure or inter-cluster link outage). Administrators use this information to make manual fail-over decisions.

The following example shows the output of a **ll components** command after `cluster-2` has failed:

```
VPlexcli:/cluster-witness> ll components/
/cluster-witness/components:
Name        ID  Admin State  Operational State                Mgmt Connectivity
----------  --  -----------  -------------------------------  -----------------
```

```
cluster-1   1   enabled      remote-cluster-isolated-or-dead  ok
cluster-2   2   unknown      -                                failed
server      -   enabled      cluster-unreachable              ok

VPlexcli:/cluster-witness> ll components/*
/cluster-witness/components/cluster-1:
Name                      Value
------------------------  ------------------------------------------------------
admin-state               enabled
diagnostic                WARNING: Current state of cluster-1 is
                          remote-cluster-isolated-or-dead (last state change: 0
                          days, 15 secs ago; last message from server: 0 days, 0
                          secs ago.)
id                        1
management-connectivity   ok
operational-state         remote-cluster-isolated-or-dead

/cluster-witness/components/cluster-2:
Name                      Value
------------------------  ------------------------------------------------------
admin-state               unknown
diagnostic                WARNING: Cannot establish connectivity with cluster-2
                          to query diagnostic information.
id                        2
management-connectivity   failed
operational-state         -

/cluster-witness/components/server:
Name                      Value
------------------------  ------------------------------------------------------
admin-state               enabled
diagnostic                WARNING: Current state is cluster-unreachable (last
                          state change: 0 days, 15 secs ago.) (last time of
                          communication with cluster-1: 0 days, 0 secs ago.)
                          (last time of communication with cluster-2: 0 days,
                          34 secs ago.)
id                        -
management-connectivity   ok
operational-state         cluster-unreachable
```

Install, enable, and manage VPLEX Witness

VPLEX Witness is to designed to improve availability in your storage network.

You should carefully consider how you install and enable VPLEX Witness.

Installation considerations

It is important to deploy the VPLEX Witness server VM in a separate failure domain than either cluster.

A failure domain is a set of entities effected by the same set of faults. The scope of the failure domain depends on the set of fault scenarios that can be tolerated in a given environment. For example:

- If the two clusters are deployed on different floors of the same data center, deploy the VPLEX Witness Server VM on a separate floor.
- If the two clusters are deployed in two different data centers, deploy the VPLEX Witness Server VM in the third data center.

WARNING

If the VPLEX Witness Server VM cannot be deployed in a failure domain independent from either of VPLEX cluster, VPLEX Witness should not be installed.

The following are additional recommendations for installing and deploying VPLEX Witness:

- Connect the VPLEX Witness Server VM to a power source that is independent of power sources supplying power to the clusters.
- Connect the VPLEX Witness Server VM to an Uninterruptable Power Supply (UPS).
- Enable the BIOS Virtualization Technology (VT) extension on the ESX host where the VPLEX Witness Server VM is installed to ensure performance of the VPLEX Witness Server VM.
- Secure access the ESX host server using a password.
- The IP Management network (connecting the cluster management servers and the VPLEX Witness Server VM) must be physically separate from the inter-cluster networks.
- Latency in the network between VPLEX Witness Server VM and the cluster management servers should not exceed 1 second (round trip).

Complete instructions to install and deploy the VPLEX Witness Server VM are available in the VPLEX generator.

CLI context

The VPLEX Witness software includes a client on each of the VPLEX clusters. VPLEX Witness does not appear in the CLI until the client has been configured.

Complete instructions to configure the VPLEX Witness client are available in the VPLEX generator.

Descriptions of the CLI commands to configure the VPLEX Witness client are available in the VPLEX CLI Guide.

Once VPLEX Witness is installed and configured on the clusters, a CLI context called "cluster-witness" appears in the CLI context tree:

```
VPlexcli:/cluster-witness> ls
Attributes:
Name                  Value
--------------        -------------
admin-state           enabled
private-ip-address    128.221.254.3
public-ip-address     10.31.25.45

Contexts:
components
```

Enable/disable VPLEX Witness client

Once VPLEX Witness client is configured on the clusters, the cluster-witness context is added to the CLI context tree, and you can enable and disable the client.

Enable the VPLEX Witness client on the clusters to provide guidance.

Disable the VPLEX Witness client on the clusters when:

- The VPLEX Witness Server VM fails, and is not expected to be restored quickly.
- Connectivity between both clusters and the VPLEX Witness Server VM fails, and is not expected to be restored quickly.

  There is no impact on I/O to distributed volumes. Both clusters send a call-home notification to indicate that they have lost connectivity with the VPLEX Witness Server. However, an additional failure (cluster failure or inter-cluster link outage) may result in a Data Unavailability.

CAUTION

When you disable the VPLEX Witness client on the clusters, all distributed volumes in consistency groups use their consistency group-level detach rules to determine which cluster is preferred.

Complete instructions to enable/disable the VPLEX Witness client are available in the VPLEX generator.

Descriptions of the CLI commands to enable/disable the VPLEX Witness client are available in the VPLEX CLI Guide.

Renew the VPLEX Witness security certificate

When VPLEX Witness is installed, a VPLEX Witness host security certificate is created.

All types of security certificates expire and must be periodically renewed. Host certificates (including the VPLEX Witness host security certificate) must be renewed every 2 years.

Complete instructions to renew the VPLEX Witness security certificate are available in the VPLEX generator.

Descriptions of the CLI commands to renew the VPLEX Witness security certificate are available in the VPLEX CLI Guide.

VPLEX Witness operation

This section describes the output of CLI commands during normal operations and in various failure states:

- "Normal operation"
- "Single failures (no data unavailability)"
- "Dual failures (no data unavailability)"
- "Single failures (data unavailability at one cluster)"
- "Dual failure (cluster isolation/DU at one cluster)"

Use the **ls** command in /cluster-witness context to display the state and IP addresses.

Use the **ll components/** command in /cluster-witness context to display status of components (VPLEX clusters and the VPLEX Witness Server).

Use the **ll components/*** command in /cluster-witness context to display detailed information (including diagnostics)

Table 10 lists the fields in the output of these commands.

Table 10 VPLEX Witness display fields (1 of 2)

Field	Description
Name	Name of component, For VPLEX clusters – name assigned to cluster. For VPLEX Witness server – "server".
id	ID of a VPLEX cluster. Always blank "-" for Witness server.
admin state	Identifies whether VPLEX Witness is enabled/disabled. Valid values are: **enabled** - VPLEX Witness functionality is enabled on this component. **disabled** - VPLEX Witness functionality is disabled on this component. **inconsistent** - All VPLEX Witness components are reachable over the management network but some components report their administrative state as disabled while others report it as enabled. This is a rare state which may result failure during enabling or disabling. **unknown** - This component is not reachable and its administrative state cannot be determined.
private-ip-address	Private IP address of the VPLEX Witness Server VM used for VPLEX Witness-specific traffic.
public-ip-address	Public IP address of the VPLEX Witness Server VM used as an endpoint of the IPsec tunnel.
diagnostic	String generated by CLI based on the analysis of the data and state information reported by the corresponding component. **WARNING: Cannot establish connectivity with VPLEX Witness Server to query diagnostic information.** - VPLEX Witness Server or one of the clusters is unreachable. **Local cluster-x hasn't yet established connectivity with the server** - The cluster has never connected to VPLEX Witness Server. **Remote cluster-x hasn't yet established connectivity with the server** - The cluster has never connected to VPLEX Witness Server. **Cluster-x has been out of touch from the server for X days, Y secs** - VPLEX Witness Server has not received messages from a given cluster for longer than 60 seconds. **VPLEX Witness server has been out of touch for X days, Y secs** - Either cluster has not received messages from VPLEX Witness Server for longer than 60 seconds. **VPLEX Witness is not enabled on component-X, so no diagnostic information is available** - VPLEX Witness Server or either of the clusters is disabled.

Table 10 VPLEX Witness display fields (2 of 2)

Operational State	Operational state of the corresponding component. For clusters – Healthy state is in-contact. All other states indicate a problem. For VPLEX Witness Server – Healthy state is clusters-in-contact. All other states indicate a problem. For VPLEX Witness Server: **clusters-in-contact** - Both clusters are in contact with each other over the inter-cluster network. **cluster-partition** - The clusters are partitioned from each other over the inter-cluster network, and the VPLEX Witness Server can still talk to each cluster. See NOTE: below. **cluster-partition** - The clusters are partitioned from each other over the inter-cluster network, and the VPLEX Witness Server can still talk to each cluster. See NOTE: below. **cluster-unreachable** - One cluster has either failed or become isolated (that is partitioned from its peer cluster and disconnected from the VPLEX Witness Server). See NOTE: below. **unknown** - VPLEX Witness Server does not know the states of one or both of the clusters and needs to learn them before it can start making decisions. VPLEX Witness Server assumes this state upon startup. --- **Note:** When the server's operational state is cluster-partition or cluster-unreachable, this may not reflect the current observation of the VPLEX Witness Server. The VPLEX Witness Server state and the guidance that it provides to the clusters based on its state is persistent -- if VPLEX Witness Server observes a failure (changes its state and provides guidance to the clusters), the VPLEX Witness Server maintains this state even if current observations change. VPLEX Witness Server maintains its failure state and guidance until both clusters and their connectivity fully recover. This policy is implemented in order to avoid potential data corruption scenarios due to split brain (that would be possible if this semantics were not followed). --- For VPLEX clusters: **in-contact** - This cluster is in contact with its peer over the inter-cluster network. Rebuilds may be in progress. Subject to other system-wide restrictions, I/O to all distributed virtual volumes in all synchronous consistency groups is allowed from the perspective of VPLEX Witness. **cluster-partition** - This cluster is not in contact with its peer and the VPLEX Witness Server has declared that two clusters partitioned. Subject to other system-wide restrictions, I/O to all distributed virtual volumes in all synchronous consistency groups (with specific preference rule set) is allowed from the perspective of VPLEX Witness. **remote-cluster-isolated-or-dead** - This cluster is not in contact with its peer and the VPLEX Witness Server has declared that the remote cluster (the peer) was isolated or dead. Subject to other system-wide restrictions, I/O to all distributed virtual volumes in all synchronous consistency groups (with specific preference rule set) is allowed from the perspective of VPLEX Witness. **local-cluster-isolated** - This cluster is not in contact with its peer and the VPLEX Witness Server has declared that the remote cluster (the peer) is the only proceeding cluster. This cluster must suspend I/O to all distributed virtual volumes in all synchronous consistency groups regardless of cluster preference. NOTE: When a cluster is isolated from both the remote cluster and VPLEX Witness Server, its state is unknown. When connectivity to VPLEX Witness server is restored, the state of this cluster changes to local-cluster-isolated if this cluster remains partitioned from the peer cluster. **unknown** - This cluster is not in contact with its peer over the inter-cluster network and is awaiting guidance from the VPLEX Witness Server. I/O to all distributed virtual volumes in all synchronous consistency groups is suspended regardless of cluster preference. NOTE: When its state is local-cluster-isolated, the cluster does not receive guidance from VPLEX Witness Server. If connectivity with VPLEX Witness Server is restored before the inter-cluster link is restored state is local-cluster-isolated.
Mgmt Connectivity	Reachability of the specified Witness component over the IP management network from the management server where the CLI command is run. **ok** - The component is reachable **failed** - The component is not reachable

Normal operation

During normal operation (no failures), the output of display commands from the cluster-witness CLI context are as follows:

```
VPlexcli:/cluster-witness> ls
Attributes:
Name                  Value
--------------        -------------
admin-state           enabled
private-ip-address    128.221.254.3
public-ip-address     10.31.25.45

Contexts:
components

VPlexcli:/cluster-witness> ll components/
/cluster-witness/components:
Name        ID  Admin State  Operational State    Mgmt Connectivity
----------  --  -----------  -------------------  -----------------
cluster-1   1   enabled      in-contact           ok
cluster-2   2   enabled      in-contact           ok
server      -   enabled      clusters-in-contact  ok

VPlexcli:/cluster-witness> ll components/*
/cluster-witness/components/cluster-1:
Name                      Value
------------------------  ------------------------------------------------------
admin-state               enabled
diagnostic                INFO: Current state of cluster-1 is in-contact (last
                          state change: 0 days, 13056 secs ago; last message
                          from server: 0 days, 0 secs ago.)
id                        1
management-connectivity   ok
operational-state         in-contact

/cluster-witness/components/cluster-2:
Name                      Value
------------------------  ------------------------------------------------------
admin-state               enabled
diagnostic                INFO: Current state of cluster-2 is in-contact (last
                          state change: 0 days, 13056 secs ago; last message
                          from server: 0 days, 0 secs ago.)
id                        2
management-connectivity   ok
operational-state         in-contact

/cluster-witness/components/server:
Name                      Value
------------------------  ------------------------------------------------------
admin-state               enabled
diagnostic                INFO: Current state is clusters-in-contact (last state
                          change: 0 days, 13056 secs ago.) (last time of
                          communication with cluster-2: 0 days, 0 secs ago.)
                          (last time of communication with cluster-1: 0 days, 0
                          secs ago.)
id                        -
management-connectivity   ok
operational-state         clusters-in-contact
```

Single failures (no data unavailability)

Single failures that do not impact I/O to distributed volumes include:

- Failure of the VPLEX Witness Server VM
- Failure of the link between VPLEX Witness Server VM and cluster-1
- Failure of the link between VPLEX Witness Server VM and cluster-2

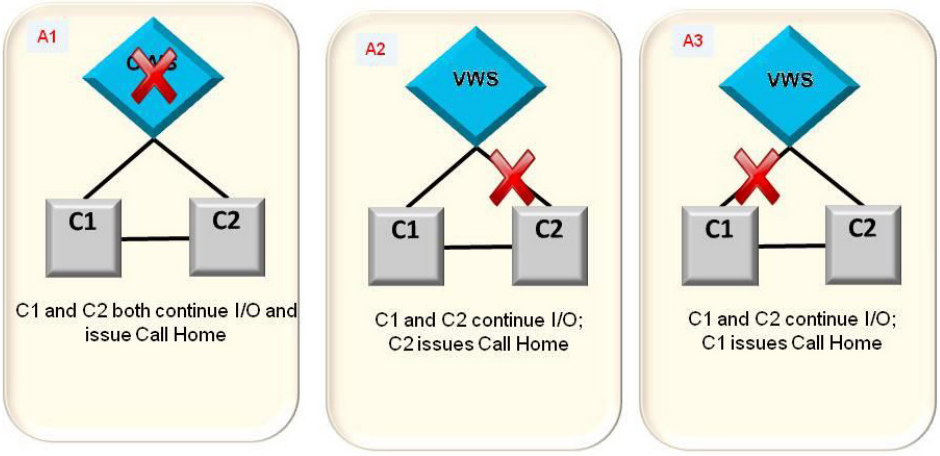


Figure 14 VPLEX Witness Server and VPLEX Witness Server-to-clusters connectivity failures

The cluster that lost connectivity with VPLEX Witness Server sends a call-home 30 seconds after connectivity is lost.

The following example shows the output of the **ll** commands when the VPLEX Witness Server VM loses contact with cluster-2 (A2 in Figure 14):

```
VPlexcli:/cluster-witness> ll components/
/cluster-witness/components:
Name         ID  Admin State  Operational State    Mgmt Connectivity
----------   --  -----------  -------------------  -----------------
cluster-1    1   enabled      in-contact           ok
cluster-2    2   enabled      in-contact           ok
server       -   enabled      clusters-in-contact  ok

VPlexcli:/cluster-witness> ll components/*
/cluster-witness/components/cluster-1:
Name                      Value
------------------------  ------------------------------------------------------------
admin-state               enabled
diagnostic                INFO: Current state of cluster-1 is in-contact (last
                          state change: 0 days, 13439 secs ago; last message
                          from server: 0 days, 0 secs ago.)
id                        1
management-connectivity   ok
operational-state         in-contact

/cluster-witness/components/cluster-2:
Name                      Value
------------------------  ------------------------------------------------------------
admin-state               enabled
diagnostic                INFO: Current state of cluster-2 is in-contact (last
                          state change: 0 days, 13439 secs ago; last message
                          from server: 0 days, 2315 secs ago.)
id                        2
management-connectivity   ok
operational-state         in-contact

/cluster-witness/components/server:
Name                      Value
------------------------  ------------------------------------------------------------
```

```
admin-state                enabled
diagnostic                 INFO: Current state is clusters-in-contact (last state
                           change: 0 days, 13439 secs ago.) (last time of
                           communication with cluster-1: 0 days, 0 secs ago.)
                           (last time of communication with cluster-2: 0 days,
                           2315 secs ago.)
id                         -
management-connectivity    ok
operational-state          clusters-in-contact
```

Dual failures (no data unavailability)

The failure of both links between VPLEX Witness Server VM and cluster-1 and cluster-2 also does not impact I/O to distributed volumes.

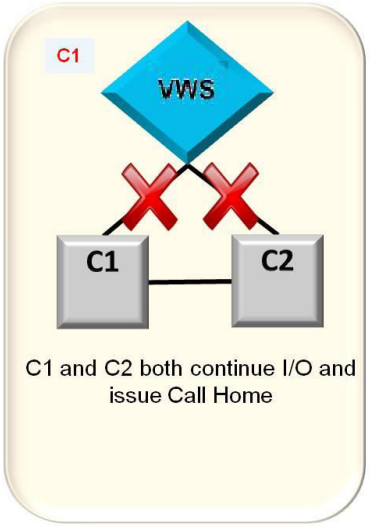


Figure 15 **Dual VPLEX Witness Server-to-clusters connectivity failures**

Both clusters send a call-home indicating that they have lost connectivity with the VPLEX Witness Server.

Note: If an additional inter-cluster link or cluster failure occurs, the system is at risk of data unavailability unless manual action is taken to disable VPLEX witness.

The following example shows the output of the **ll** commands when the VPLEX Witness Server VM loses contact with both clusters:

Note: The cluster-witness CLI context on cluster-2 shows the same loss of connectivity to the CSWS.

```
VPlexcli:/cluster-witness> ll components/
/cluster-witness/components:
Name        ID  Admin State  Operational State  Mgmt Connectivity
----------  --  -----------  -----------------  -----------------
cluster-1   1   enabled      in-contact         ok
cluster-2   2   enabled      in-contact         ok
server      -   unknown      -                  failed

VPlexcli:/cluster-witness> ll components/*
/cluster-witness/components/cluster-1:
Name                   Value
---------------------  ------------------------------------------------------
```

```
admin-state                enabled
diagnostic                 WARNING: Current state of cluster-1 is in-contact
                           (last state change: 0 days, 94 secs ago; last message
                           from server: 0 days, 34 secs ago.)
id                         1
management-connectivity    ok
operational-state          in-contact

/cluster-witness/components/cluster-2:
Name                       Value
-----------------------    ------------------------------------------------------
admin-state                enabled
diagnostic                 WARNING: Current state of cluster-2 is in-contact
                           (last state change: 0 days, 94 secs ago; last message
                           from server: 0 days, 34 secs ago.)
id                         2
management-connectivity    ok
operational-state          in-contact

/cluster-witness/components/server:
Name                       Value
-----------------------    ------------------------------------------------------
admin-state                unknown
diagnostic                 WARNING: Cannot establish connectivity with Cluster
                           Witness Server to query diagnostic information.
id                         -
management-connectivity    failed
operational-state          -
```

Single failures (data unavailability at one cluster)

Single failures that impact I/O to distributed volumes include:

- Failure of the inter-cluster link
- Failure of cluster-1
- Failure of cluster-2

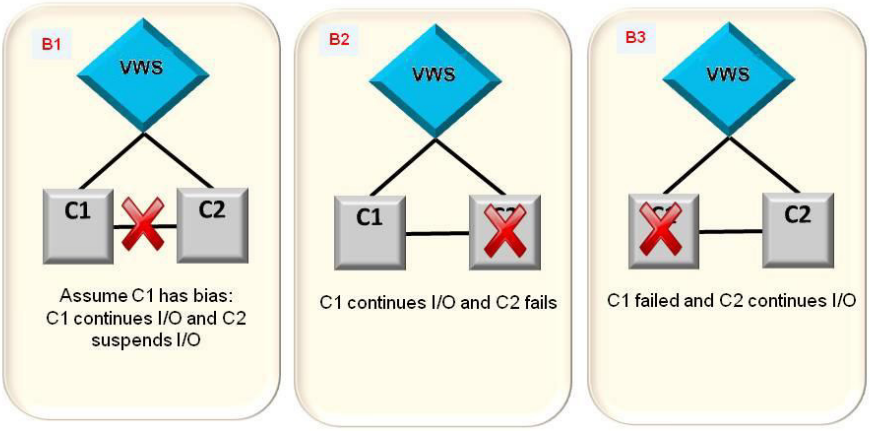


Figure 16 Single failures; inter-cluster link or one cluster resulting in data unavailability

When a single failure occurs, each cluster receives guidance from VPLEX Witness Server and sends a call-home indicating the guidance and changes its internal state.

The following examples show the output of the **ls** and **ll** commands following:

- Inter-cluster link failure (B1 in Figure 16)
- cluster-1 failure (B3 in in Figure 16)
- cluster-2 failure (B2 in Figure 16)

Inter-cluster link failure

The following example shows the output of the **ll** commands following a failure of the inter-cluster link (B1 in Figure 16):

```
VPlexcli:/cluster-witness> ll components/
/cluster-witness/components:
Name        ID  Admin State  Operational State  Mgmt Connectivity
----------  --  -----------  -----------------  -----------------
cluster-1   1   enabled      cluster-partition  ok
cluster-2   2   enabled      cluster-partition  ok
server      -   enabled      cluster-partition  ok

VPlexcli:/cluster-witness> ll components/*
/cluster-witness/components/cluster-1:
Name                    Value
----------------------  ------------------------------------------------------
admin-state             enabled
diagnostic              WARNING: Current state of cluster-1 is
                        cluster-partition (last state change: 0 days, 56 secs
                        ago; last message from server: 0 days, 0 secs ago.)
id                      1
management-connectivity ok
operational-state       cluster-partition

/cluster-witness/components/cluster-2:
Name                    Value
----------------------  ------------------------------------------------------
admin-state             enabled
diagnostic              WARNING: Current state of cluster-2 is
                        cluster-partition (last state change: 0 days, 57 secs
                        ago; last message from server: 0 days, 0 secs ago.)
id                      2
management-connectivity ok
operational-state       cluster-partition

/cluster-witness/components/server:
Name                    Value
----------------------  ------------------------------------------------------
admin-state             enabled
diagnostic              WARNING: Current state is cluster-partition (last
                        state change: 0 days, 57 secs ago.) (last time of
                        communication with cluster-1: 0 days, 0 secs ago.)
                        (last time of communication with cluster-2: 0 days, 0
                        secs ago.)
id                      -
management-connectivity ok
operational-state       cluster-partition
```

cluster-1 failure

The following example shows the output of the **ll** commands following a failure of cluster-1 (B3 in Figure 16):

```
VPlexcli:/cluster-witness> ll components/
Name       ID  Admin State  Operational State                Mgmt Connectivity
---------  --  -----------  ------------------------------  -----------------
cluster-1  1   unknown      -                               failed
cluster-2  2   enabled      remote-cluster-isolated-or-dead ok
server     -   enabled      cluster-unreachable             ok


VPlexcli:/cluster-witness> ll components/*
```

```
/cluster-witness/components/cluster-1:
Name                      Value
------------------------  ------------------------------------------------------
admin-state               unknown
diagnostic                WARNING: Cannot establish connectivity with cluster-1
                          to query diagnostic information.
id                        1
management-connectivity   failed
operational-state         -


/cluster-witness/components/cluster-2:
Name                      Value
------------------------  ------------------------------------------------------
admin-state               enabled
diagnostic                WARNING: Current state of cluster-2 is
                          remote-cluster-isolated-or-dead (last state change: 0
                          days, 49 secs ago; last message from server: 0 days, 0
                          secs ago.)
id                        2
management-connectivity   ok
operational-state         remote-cluster-isolated-or-dead


/cluster-witness/components/server:
Name                      Value
------------------------  ------------------------------------------------------
admin-state               enabled
diagnostic                WARNING: Current state is cluster-unreachable (last
                          state change: 0 days, 49 secs ago.) (last time of
                          communication with cluster-2: 0 days, 0 secs ago.)
                          (last time of communication with cluster-1: 0 days, 59
                          secs ago.)
id                        -
management-connectivity   ok
operational-state         cluster-unreachable
```

cluster-2 failure The following example shows the output of the **ll** commands following a failure of cluster-2 (B2 in Figure 16):

```
VPlexcli:/cluster-witness> ll components/
/cluster-witness/components:
Name        ID  Admin State  Operational State                Mgmt Connectivity
----------  --  -----------  ------------------------------  -----------------
cluster-1   1   enabled      remote-cluster-isolated-or-dead  ok
cluster-2   2   unknown      -                                failed
server      -   enabled      cluster-unreachable              ok

VPlexcli:/cluster-witness> ll components/*
/cluster-witness/components/cluster-1:
Name                      Value
------------------------  ------------------------------------------------------
admin-state               enabled
diagnostic                WARNING: Current state of cluster-1 is
                          remote-cluster-isolated-or-dead (last state change: 0
                          days, 15 secs ago; last message from server: 0 days, 0
                          secs ago.)
id                        1
management-connectivity   ok
operational-state         remote-cluster-isolated-or-dead

/cluster-witness/components/cluster-2:
Name                      Value
------------------------  ------------------------------------------------------
admin-state               unknown
```

```
diagnostic id                          2
management-connectivity  failed
operational-state        -

/cluster-witness/components/server:
Name                     Value
-----------------------  ------------------------------------------------------
admin-state              enabled
diagnostic               WARNING: Current state is cluster-unreachable (last
                         state change: 0 days, 15 secs ago.) (last time of
                         communication with cluster-1: 0 days, 0 secs ago.)
                         (last time of communication with cluster-2: 0 days,
                         34 secs ago.)
id                       -
management-connectivity  ok
operational-state        cluster-unreachable

diagnostic               WARNING: Cannot establish connectivity with cluster-2
                         to query diagnostic information
```

Note: Status can be used to determine that cluster-2 has failed and that Consistency Groups may need manual intervention.

Dual failure (cluster isolation/DU at one cluster)

Cluster isolation occurs when a cluster loses network connectivity to both its peer cluster and VPLEX Witness Server. Cluster isolation results in data unavailability on the isolated cluster.

VPLEX Witness server guides the cluster that is still connected to continue I/O.

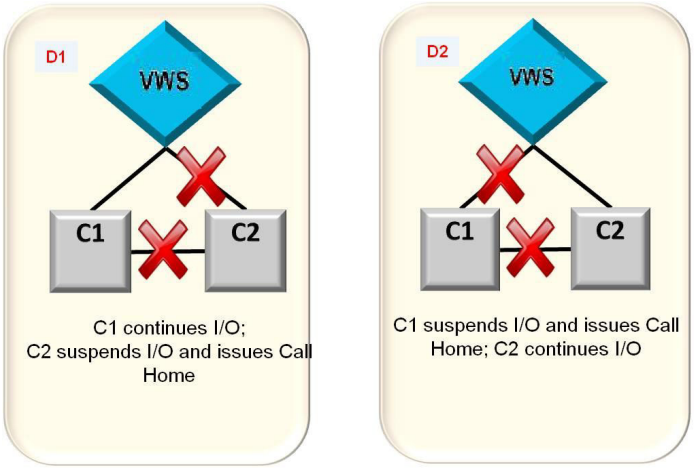


Figure 17 Cluster isolation resulting in data unavailability

In D1 in Figure 17, cluster-2 is isolated. VPLEX Witness Server guides cluster-1 to proceed with I/O on all distributed volumes in all consistency groups regardless of preference.

Cluster-2 suspends I/O to all distributed volumes in all consistency groups regardless of preference.

The following examples show the output of the **ll** commands:

- From cluster-1 when cluster-2 is isolated

- From cluster-2 when cluster-2 is isolated
- After the VPLEX Witness Server-cluster link is restored

From cluster-1 when cluster-2 is isolated

The following example shows the output of the **ll** command when cluster-2 is isolated, and cluster-1 remains connected to VPLEX Witness Server (D1 in Figure 17):

```
VPlexcli:/cluster-witness> ll components/
/cluster-witness/components:
Name         ID  Admin State  Operational State                Mgmt Connectivity
----------   --  -----------  ------------------------------   -----------------
cluster-1    1   enabled      remote-cluster-isolated-or-dead  ok
cluster-2    2   unknown      -                                failed
server       -   enabled      cluster-unreachable              ok

VPlexcli:/cluster-witness> ll components/*
/cluster-witness/components/cluster-1:
Name                     Value
-----------------------  ------------------------------------------------------
admin-state              enabled
diagnostic               WARNING: Current state of cluster-1 is
                         remote-cluster-isolated-or-dead (last state change: 0
                         days, 35 secs ago; last message from server: 0 days, 0
                         secs ago.)
id                       1
management-connectivity  ok
operational-state        remote-cluster-isolated-or-dead

/cluster-witness/components/cluster-2:
Name                     Value
-----------------------  ------------------------------------------------------
admin-state              enabled
diagnostic               WARNING: Cannot establish connectivity with cluster-2
                         to query diagnostic information.
id                       2
management-connectivity  failed
operational-state        -

/cluster-witness/components/server:
Name                     Value
-----------------------  ------------------------------------------------------
admin-state              enabled
diagnostic               WARNING: Current state is cluster-unreachable (last
                         state change: 0 days, 35 secs ago.) (last time of
                         communication with cluster-1: 0 days, 0 secs ago.)
                         (last time of communication with cluster-2: 0 days,
                         103 secs ago.) Remote cluster has been out of touch
                         from the server for 0 days, 103 secs.
id                       -
management-connectivity  ok
operational-state        cluster-unreachable
```

From cluster-2 when cluster-2 is isolated

CLI output for cluster-2 (which is isolated):

```
VPlexcli:/cluster-witness> ll components/
/cluster-witness/components:
Name         ID  Admin State  Operational State                Mgmt Connectivity
----------   --  -----------  ------------------------------   -----------------
cluster-1    1   unknown      -                                failed
cluster-2    2   unknown      -                                failed
server       -   unknown      -                                failed

VPlexcli:/cluster-witness> ll components/*
/cluster-witness/components/cluster-1:
Name                     Value
```

```
----------------------  ------------------------------------------------------
admin-state             unknown
diagnostic              WARNING: Cannot establish connectivity with cluster-1
                        to query diagnostic information.
id                      1
management-connectivity failed
operational-state       -

/cluster-witness/components/cluster-2:
Name                    Value
----------------------  ------------------------------------------------------
admin-state             unknown
diagnostic              WARNING: Cannot establish connectivity with cluster-1
                        to query diagnostic information.
id                      2
management-connectivity failed
operational-state       -

/cluster-witness/components/server:
Name                    Value
----------------------  ------------------------------------------------------
admin-state             unknown
diagnostic              WARNING: Cannot establish connectivity with Cluster
                        Witness Server to query diagnostic information.
id                      -
management-connectivity failed
operational-state       -
```

After the VPLEX Witness Server-cluster link is restored

If the link between cluster-2 and VPLEX Witness Server is restored, but the inter-cluster link is NOT restored, output of the **ll** commands is as follows:

```
VPlexcli:/cluster-witness> ll components/
/cluster-witness/components:
Name        ID  Admin State  Operational State                Mgmt Connectivity
----------  --  -----------  ------------------------------  -----------------
cluster-1   1   enabled      remote-cluster-isolated-or-dead  ok
cluster-2   2   enabled      local-cluster-isolated           ok
server      -   enabled      cluster-unreachable              ok

VPlexcli:/cluster-witness> ll components/*
/cluster-witness/components/cluster-1:
Name                    Value
----------------------  ------------------------------------------------------
admin-state             enabled
diagnostic              WARNING: Current state of cluster-1 is
                        remote-cluster-isolated-or-dead (last state change: 0
                        days, 357 secs ago; last message from server: 0 days,
                        0 secs ago.)
id                      1
management-connectivity ok
operational-state       remote-cluster-isolated-or-dead

/cluster-witness/components/cluster-2:
Name                    Value
----------------------  ------------------------------------------------------
admin-state             enabled
diagnostic              WARNING: Current state of cluster-2 is
                        local-cluster-isolated (last state change: 0 days, 9
                        secs ago; last message from server: 0 days, 0 secs
                        ago.)
id                      2
management-connectivity ok
operational-state       local-cluster-isolated
```

```
/cluster-witness/components/server:
Name                    Value
----------------------  ------------------------------------------------------
admin-state             enabled
diagnostic              WARNING: Current state is cluster-unreachable (last
                        state change: 0 days, 357 secs ago.) (last time of
                        communication with cluster-1: 0 days, 0 secs ago.)
                        (last time of communication with cluster-2: 0 days, 0
                        secs ago.)
id                      -
management-connectivity ok
operational-state       cluster-unreachable
```

11

Cache vaults

This chapter describes cache vaulting on VPLEX and explains how to recover after a cache vault.

About cache vaulting

VPLEX uses the individual director's memory systems to ensure durability of user data and critical system configuration data.

User data is made durable in one of two ways depending on the cache mode used for the data.

- **Write-through cache mode** leverages the durability properties of a back-end array by writing user data to the array and obtaining an acknowledgement for the written data before it acknowledges the write back to the host.
- **Write-back cache mode** ensures data durability by storing user data into the cache memory of the director that received the I/O, then placing a protection copy of this data on another director in the local cluster before acknowledging the write to the host. This ensures the data is protected in two independent memories. The data is later destaged to back-end storage arrays that provide the physical storage media.

If a power failure on a VPLEX Geo cluster (using write-back cache mode) should occur, the data in cache memory might be at risk. In the event of data at risk from power failure in a VPLEX Geo configuration, each VPLEX director copies its dirty cache data to the local solid state storage devices (SSDs) using a process known as cache vaulting. *Dirty cache pages* are pages in a director's memory that have not been written to back-end storage but were acknowledged to the host. Dirty cache pages also include the copies protected on a second director in the cluster. After each director vaults its dirty cache pages, VPLEX then shuts down the director's firmware.

Note: Although there is no dirty cache data in VPLEX Local or VPLEX Metro configurations, vaulting is still necessary to quiesce all I/O when data is at risk due to power failure. This is done to minimize the risk of metadata corruption.

Once power is restored, the VPLEX system startup program initializes the hardware and the environmental system, checks the data validity of each vault, and unvaults the data. The process of system recovery and unvault depends largely on the configuration of the system:

- In a VPLEX Local or VPLEX Metro configuration, the cluster unvaults without recovering any vault data because there was no data to vault.
- In a VPLEX Geo system, if the remote cluster proceeded on its own (either because it was the only active cluster at the time or because of administrative action), VPLEX discards the vault and does not restore memory. Additionally, the auto-resume-at-loser parameter affects whether the recovering cluster starts processing I/O. By default this parameter is set to false for asynchronous consistency groups. This means that by default the recovering cluster discards its vault and then suspends and waits for manual intervention.
- In a VPLEX Geo system, if the remote cluster waited, the vault is recovered, the two clusters get back in touch and continue processing I/O.

When you resume operation of the cluster, if any condition is not safe, the system does not resume normal status and calls home for diagnosis and repair. This allows EMC Customer Support to communicate with the VPLEX system and restore normal system operations.

Vaulting can be used in two scenarios:

- *Data at risk due to power failure:* VPLEX monitors all components that provide power to the VPLEX cluster. If it detects AC power loss in accordance with the specifications in "Power failures that cause vault," in order to avoid a possible data loss it takes a conservative approach and initiates a cluster wide vault if the power loss exceeds 30 seconds.
- *Manual emergency cluster shutdown:* When unforeseen circumstances require an unplanned and immediate shutdown, it is known as an *emergency cluster shutdown.* You can use a CLI command to manually start vaulting if an emergency shutdown is required.

WARNING

When performing maintenance activities on a VPLEX Geo configuration, service personnel must not remove the power in one or more engines unless both directors in those engines have been shutdown and are no longer monitoring power. Failure to do so, leads to data unavailability in the affected cluster. To avoid unintended vaults, always follow official maintenance procedures.

Under normal conditions, the SPS batteries can support two consecutive vaults; this ensures, that the system can resume I/O immediately after the first power failure, and that it can still vault if there is a second power failure.

For information on the redundant and backup power supplies in VPLEX, refer to the Hardware Overview chapter of the *EMC VPLEX Product Guide*.

Related Information

For additional information on cache vaulting:

Information	Reference
Related Commands	*EMC VPLEX CLI Guide*
Information about consistency groups	Chapter 9, "Consistency groups,"
Procedures to shutdown after a cache vault	EMC VPLEX generator at EMC Online Support
Procedures to troubleshoot after a cache vault	EMC VPLEX generator at EMC Online Support

Power failures that cause vault

Note: Vaulting is a mechanism to prevent data loss when an external environmental condition causes a cluster failure. For example, if a power failure lasts longer than 30 seconds in a VPLEX Geo configuration, then each VPLEX director copies its dirty cache data to the local solid state storage devices (SSDs). Known as vaulting, this process protects user data in cache if that data is at risk due to power loss. After each director vaults its dirty cache pages, then VPLEX shuts down the director's firmware.

Vaulting is evolving rapidly with each release of GeoSynchrony. The events and/or conditions that trigger cache vaulting vary depending by release as follows:

Release 5.0.1:

- Vaulting is introduced.
- On all configurations, vaulting is triggered if all following conditions are present:
  - AC power is lost (due to power failure, faulty hardware, or power supply is not present) in power zone A from engine X,
  - AC power is lost (due to power failure, faulty hardware, or power supply is not present) in power zone B from engine Y,

(X and Y would be the same in a single engine configuration but they may or may not be the same in dual or quad engine configurations.)

- Both conditions persist for more than 30 seconds.

Release 5.0.1 Patch:

- On all configurations, vaulting is triggered if all the following conditions are present:
  - AC power is lost (due to power failure or faulty hardware) in power zone A from engine X,
  - AC power is lost (due to power failure or faulty hardware) in power zone B from engine Y,

    (X and Y would be the same in a single engine configuration but they may or may not be the same in dual or quad engine configurations.)
  - Both conditions persist for more than 30 seconds.

Release 5.1:

- In a VPLEX Geo configuration with asynchronous consistency groups, vaulting is triggered if all the following conditions are present:
  - AC power is lost (due to power failure or faulty hardware) or becomes "unknown" in a director from engine X,
  - AC power is lost (due to power failure or faulty hardware) or becomes "unknown" in director from engine Y,

    (X and Y would be the same in a single engine configuration but they may or may not be the same in dual or quad engine configurations.)
  - Both conditions persist for more than 30 seconds.
- In a VPLEX Local or VPLEX Metro configuration, vaulting is triggered if all the following conditions are present:
  - AC power is lost (due to power failure or faulty hardware) or becomes "unknown" in the minimum number of directors required for the cluster to be operational.
  - Condition persist for more than 30 seconds.

Note: UPS power conditions do not trigger any vaulting.

The vaulting process

Figure 18 shows the process VPLEX uses when it starts writing the dirty cache pages to its vault.

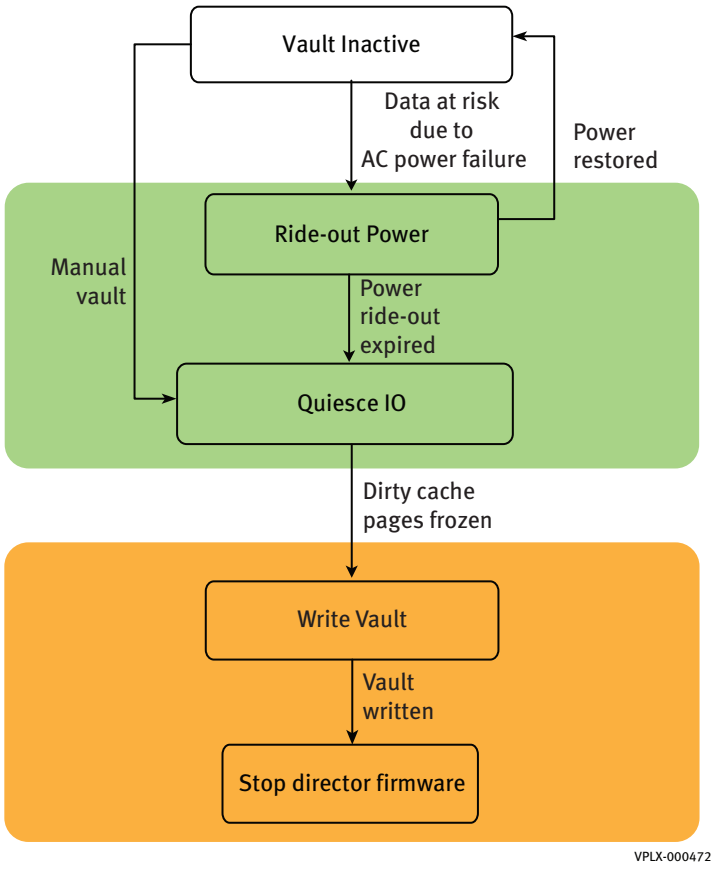


Figure 18 Cache vaulting process flow

When a cluster detects a condition described in "Power failures that cause vault," VPLEX triggers the cluster to enter a 30 second ride-out phase. This delays the (irreversible) decision to vault, allowing for a timely return of AC input to avoid vaulting altogether. During the ride-out phase, all mirror rebuilds and migrations pause, and the cluster disallows new configuration changes on the local cluster, to prepare for a possible vault.

If the power is restored prior to the 30 second ride-out, all mirror rebuilds and migrations resume, and configuration changes are once again allowed.

Power ride-out is not necessary when a manual vault has been requested. However, similar to the power ride-out phase, manual vaulting stops any mirror rebuilds and migrations and disallows any configuration changes on the local cluster.

Once the cluster has decided to proceed with vaulting the dirty cache, the vaulting cluster quiesces all I/Os and disables inter-cluster links to isolate itself from the remote cluster. These steps are required to freeze the director's dirty cache in preparation for vaulting.

Once the dirty cache (if any) is frozen, each director in the vaulting cluster isolates itself from the other directors and starts writing. When finished writing to its vault, the director stops its firmware.

This entire process is completed within the time parameters supported by the stand-by power supplies.

It is important to ensure that the cluster is shutdown in an organized fashion and to save any remaining battery charge so that recharge completes faster when the cluster is restarted.

Refer to the following procedures (published in the EMC VPLEX generator at EMC Online Support) for the procedures to safely shutdown and restart a VPLEX cluster after a vault:

- Startup after a Vault on VPLEX Local
- Startup after a Vault on VPLEX Metro or VPLEX Geo System

Recovery after vault

Once the cluster is ready to recover from the conditions that caused the vault, the cluster is powered up.

Note: If the directors in the cluster come up with degraded power such that it may put the data at risk, the cluster will either shutdown the firmware if vault recovery has not begun (to protect the vaulted data), or initiate vaulting immediately after vault recovery has finished

Figure 19 shows the process used to unvault during a recovery process.

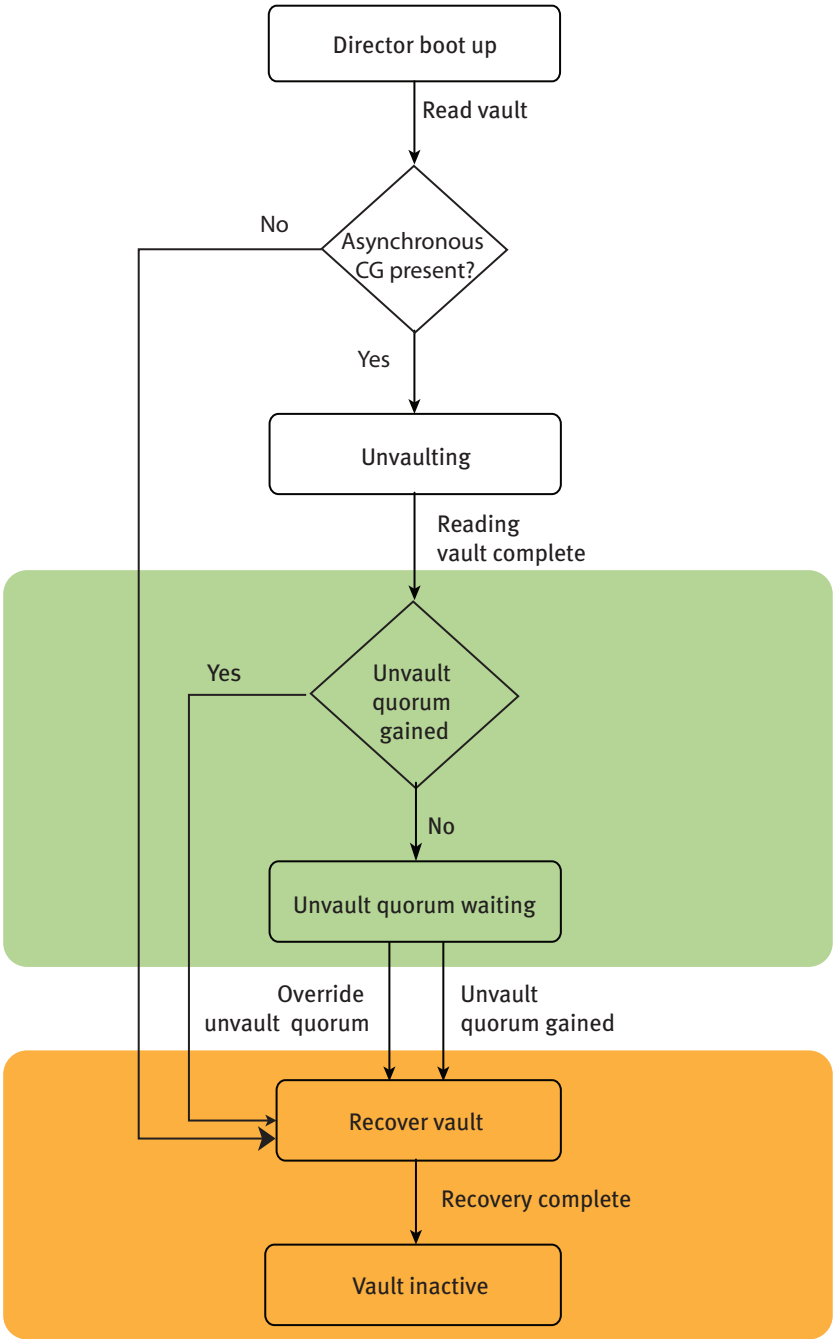


Figure 19 Unvaulting cache process

At the start of cluster recovery, VPLEX checks to see if there are any configured asynchronous consistency groups. If there are none (as would be the case in VPLEX Local and VPLEX Metro configurations), the entire unvault recovery process is skipped.

As the directors boot up, each director reads the vaulted dirty data from its respective vault disk. Once the directors have completed reading the vault, each director evaluates if the unvaulted data can be recovered.

The cluster then evaluates if it has gained unvault quorum. The *unvault quorum* is the set of directors that vaulted their dirty cache data during the last successful cluster wide vault. In order to recover their vaulted data, which is required to preserve cache coherency and avoid data loss, these directors must boot and rejoin the cluster. If the unvault quorum is achieved the cluster proceeds to recover the vault.

If the cluster determines that it has not gained unvault quorum, it waits indefinitely for the required directors to boot up and join the cluster. During the waiting period, the cluster remains in the *unvault quorum wait state*.

After 30 minutes in the unvault quorum wait state, the cluster generates a call home indicating the current state of the cluster and indicating that manual intervention is needed to allow the cluster to process I/O.

Once the cluster enters the unvault quorum wait state it cannot proceed to the recovery phase until any of the following events happen:

- The directors required to gain unvault quorum become available
- You issue the **override unvault quorum** command and agree to accept a possible data loss

Refer to the VPLEX generator troubleshooting procedures for cache vaulting for instructions on how recover in this scenario. See the *VPLEX CLI Guide* for details on the use of the **override unvault quorum** command.

Successful Recovery

VPLEX Geo can handle one invalid or missing vault because each director has vaulted a copy of each dirty cache page of its protection partner. The cache can be recovered as long as the original dirty cache vault or its protected copy is available.

An *invalid* vault can be caused by:

- A director not successfully completing write of the vault
- A director containing stale (older) vault data

A vault can be *missing* because:

- A director failed during unvault
- A director never came up during cluster power up

If the cluster determines it has sufficient valid vaults, it proceeds with recovery of the vaulted data into the distributed cache. In this scenario the unvaulted cluster looks like a cluster that has recovered after an inter-cluster link outage as no data is lost on the vaulting cluster. VPLEX behavior following this recovery process depends on how the detach rules were configured for each asynchronous consistency group.

Refer to Chapter 9, “Consistency groups.”

Unsuccessful Recovery

If the cluster determines that more than one invalid vault is present, the cluster discards the vault and reports a data loss. In this scenario the unvaulted cluster looks like a cluster that has recovered after a cluster failure. The cluster still waits for all

configured directors to boot and rejoin the cluster. The volumes are marked as **recovery-error** and refuse I/O. If one volume of a consistency group is marked **recovery-error**, all other volumes of that consistency group must also refuse I/O.

12

RecoverPoint

EMC's RecoverPoint provides continuous data protection and remote replication for on-demand protection and recovery at any point in time.

This chapter provides an overview of the RecoverPoint product family and how RecoverPoint can be deployed with VPLEX Local and Metro configurations.

EMC RecoverPoint provides comprehensive data protection by continuous replication (splitting) of host writes. With RecoverPoint, applications can be recovered to any point in time. Replicated writes can be written to local volumes to provide recovery from operational disasters, to remote volumes to provide recovery from site disasters, or both.

RecoverPoint supports 3 types of splitters:

- Host OS-based splitters
- Intelligent fabric-based splitters (SANTap and Brocade)
- Storage-based splitters (CLARiiON CX4, VNX series, and Symmetrix VMAX)

VPLEX includes a RecoverPoint splitter. The splitter is built into VPLEX such that VPLEX volumes can have their I/O replicated by RecoverPoint Appliances (RPAs) to volumes located in VPLEX or on one or more heterogeneous storage arrays.

Note: For GeoSynchrony 5.1, RecoverPoint integration is offered for VPLEX Local and VPLEX Metro configurations (not for Geo).

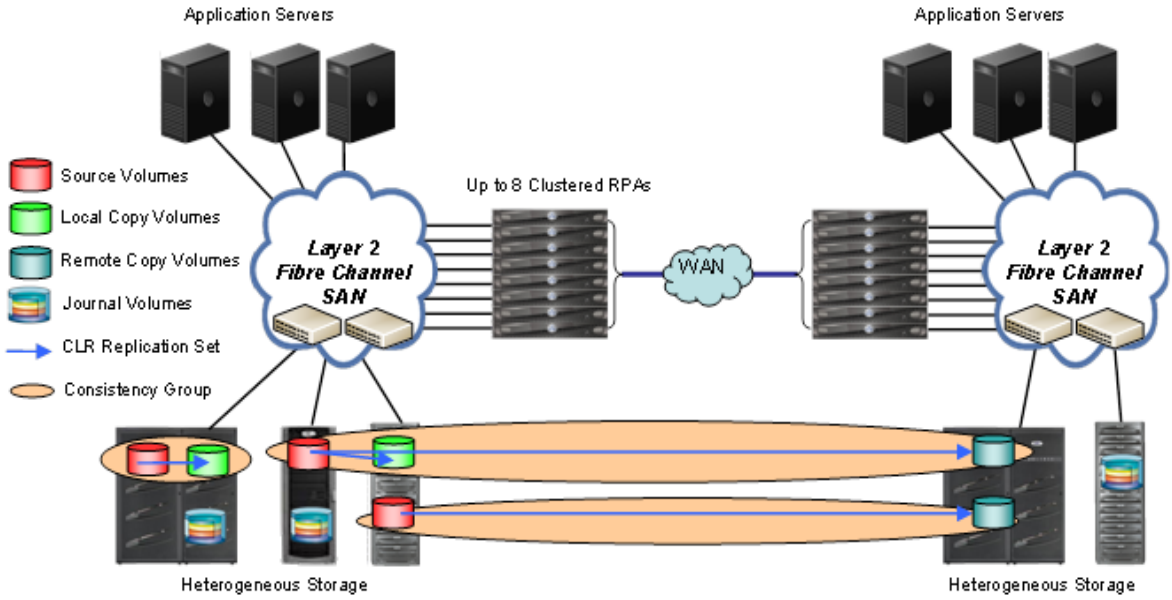


Figure 20 RecoverPoint architecture

The VPLEX splitter enables VPLEX volumes in a VPLEX Local or VPLEX Metro to mirror I/O to a RecoverPoint Appliance (RPA) performing continuous data protection (CDP), continuous remote replication (CRR), or concurrent local and remote data protection (CLR).

RecoverPoint terminology and concepts

This section introduces basic terms and concepts that you need to understand RecoverPoint.

RPA RecoverPoint Appliance. The hardware that manages all aspects of data protection. One RPA can manage multiple storage groups, each with differing policies.

A minimum of two and a maximum of eight RPAs are installed at each site, located in the same facility as the host and storage. The set of RPAs installed at each site is referred to as an "RPA cluster". If one RPA in a cluster fails, the functions provided by the failed RPA are automatically moved to one or more of the remaining RPAs.

The RPAs at the production site transfer the split I/O to the replica site.

The RPAs at the replica site distribute the data to the replica storage.

In the event of failover, these roles can be reversed. The same RPA can serve as the production RPA for one consistency group and the replica RPA for another.

volumes

All RecoverPoint volumes can be hosted on VPLEX. In practice, some volumes may be hosted on VPLEX and other hosted on non-VPLEX storage. For example, the repository volume for an existing RPA cluster cannot be moved. If you are installing VPLEX into an existing RecoverPoint configuration, the repository volume is already configured on non-VPLEX storage.

The following types of volumes are required in all RecoverPoint configurations:

- **Repository volume** - A volume dedicated to RecoverPoint for each RPA cluster. The repository volume serves all RPAs of the particular RPA cluster and the splitter associated with that cluster. The repository volume stores configuration information about the RPAs and RecoverPoint consistency groups.

  There is one repository volume per RPA cluster.
- **Production volumes** - Volumes that are written to by the host applications. Writes to production volumes are split such that they are sent to both the normally designated volumes and RPAs simultaneously.
- **Replica volumes** - Volumes to which production volumes replicate.
- **Journal volumes** - Volumes that contain data waiting to be distributed to target replica volumes and copies of the data previously distributed to the target volumes. Journal volumes allow convenient rollback to any point in time, enabling instantaneous recovery for application environments.

Snapshot/PIT

A point-in-time copy that preserves the state of data at an instant in time, by storing only those blocks that are different from an already existing full copy of the data.

Snapshots are also referred to as Point In Time (PIT). Snapshots stored at a replica journal represent the data that has changed on the production storage since the closing of the previous snapshot.

Image access

User operation on a replica journal to enable read/write access to a selected PIT at a replica. There are four image access modes:

- **Logged (physical) access** - Used for production recovery, failover, testing, and cloning a replica.
- **Direct access** - This access mode can only be enabled after logged access, or virtual access with roll, are enabled. Used for extensive processing with a high write-rate, when image access is needed for a long period of time (and may not have the journal space to support all of the data written to the image access log in this time), and when it is not required to save the history in the replica journal (the replica journal is lost after direct access).
- **Virtual (instant) access** - Used for single file recovery or light testing. Used to gain access to the replica data immediately, or when I/O performance is not important.

- **Virtual (instant) access with roll** - Used for production recovery, failover, or processing with a high write-rate. Used when the PIT is far from the current PIT (and would take too long to access in logged access mode).

IMPORTANT

In the current release, virtual (instant) access and virtual (instant) access with roll are not supported by the VPLEX splitter.

Bookmark A label applied to a snapshot so that the snapshot can be explicitly called (identified) during recovery processes (during image access).

Bookmarks are created through the CLI or GUI and can be created manually, by the user, or automatically, by the system. Bookmarks created automatically can be created at pre-defined intervals or in response to specific system events. Parallel bookmarks are bookmarks that are created simultaneously across multiple consistency groups.

RecoverPoint configurations

RecoverPoint supports three replication configurations:

- Continuous data protection (CDP)
- Continuous remote replication (CRR)
- Concurrent local and remote data protection (CLR)

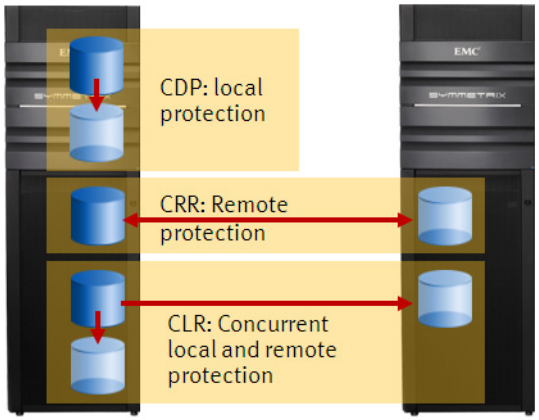


Figure 21 RecoverPoint configurations

Continuous data protection (CDP)

In a CDP configuration, RecoverPoint continuously replicates data within the same site.

Every write is kept in the journal volume, allowing recovery to any point in time. By default, snapshot granularity is set to per second, so the exact data size and contents are determined by the number of writes made by the host application per second. If necessary, the snapshot granularity can be set to per write.

CDP configurations include:

- Standard CDP - all components (splitters, storage, RPAs, and hosts) are located at the same site.

- Stretch CDP - the production host is located at the local site, splitters and storage are located at both the bunker site and the local site, and the RPAs are located at the bunker site. The repository volume and both the production and local journals are located at the bunker site.

Continuous remote replication (CRR)

In CRR configurations, data is transferred between a local and a remote site over Fibre Channel or a WAN. The RPAs, storage, and splitters are located at both the local and the remote site.

By default, the replication mode is set to asynchronous, and the snapshot granularity is set to dynamic, so the exact data size and contents are determined by the policies set by the user and system performance. This provides protection to application consistent and other specific points in time.

Synchronous replication is supported when the local and remote sites are connected using Fibre Channel.

Concurrent local and remote data protection (CLR)

A combination of both CDP and CRR. In a CLR configuration, RecoverPoint replicates data to both a local and a remote site simultaneously, providing concurrent local and remote data protection.

The CDP copy is normally used for operational recovery, while the CRR copy is normally used for disaster recovery.

RecoverPoint/VPLEX configurations

RecoverPoint can be configured on VPLEX Local or Metro systems as follows:

- "VPLEX Local and RecoverPoint CDP"
- "VPLEX Local and RecoverPoint CRR/CLR"
- "VPLEX Metro and RecoverPoint CDP at one site"
- "VPLEX Metro and RecoverPoint CRR/CLR"

In VPLEX Local systems, RecoverPoint can replicate local volumes.

In VPLEX Metro systems, RecoverPoint can replicate local volumes and distributed RAID 1 volumes.

Virtual volumes can be replicated locally (CDP), remotely (CRR), or both (CLR).

Distances between production sources and replication volumes vary based on the recovery objectives, inter-site bandwidth, latency, and other limitations outlined in the EMC Simple Support Matrix (ESSM) for RecoverPoint.

VPLEX Local and RecoverPoint CDP

In VPLEX Local/RecoverPoint CDP configurations, I/O is split to replica volumes that are located at the same site.

RPAs are deployed with the VPLEX cluster.

This configuration supports unlimited points in time, (with granularity up to a single write) for local VPLEX virtual volumes. The CDP replica volume can be a VPLEX virtual volume or any other heterogeneous storage supported by RecoverPoint.

Application event aware based rollback is supported for Microsoft SQL, Microsoft Exchange and Oracle database applications.

Users can quickly return to any point-in-time, in order to recover from operational disasters.

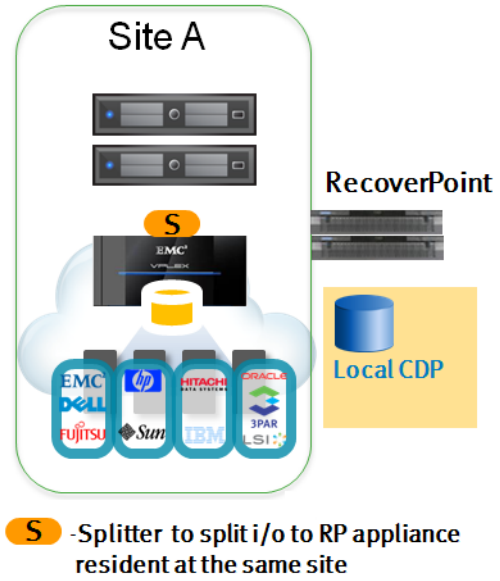


Figure 22 VPLEX Local and RecoverPoint CDP

VPLEX Local and RecoverPoint CRR/CLR

In VPLEX Local/RecoverPoint CRR/CLR configurations, I/O is split to replica volumes located both at the site where the VPLEX cluster is located and a remote site.

RPAs are deployed at both sites.

If the primary site (the VPLEX cluster site) fails, customers can recover to any point in time at the remote site. Recovery can be automated through integration with MSCE and VMware SRM.

This configuration can simulate a disaster at the primary site to test RecoverPoint disaster recovery features at the remote site.

Application event-aware based rollback is supported for Microsoft SQL, Microsoft Exchange and Oracle database applications.

The remote site can be an independent VPLEX cluster:

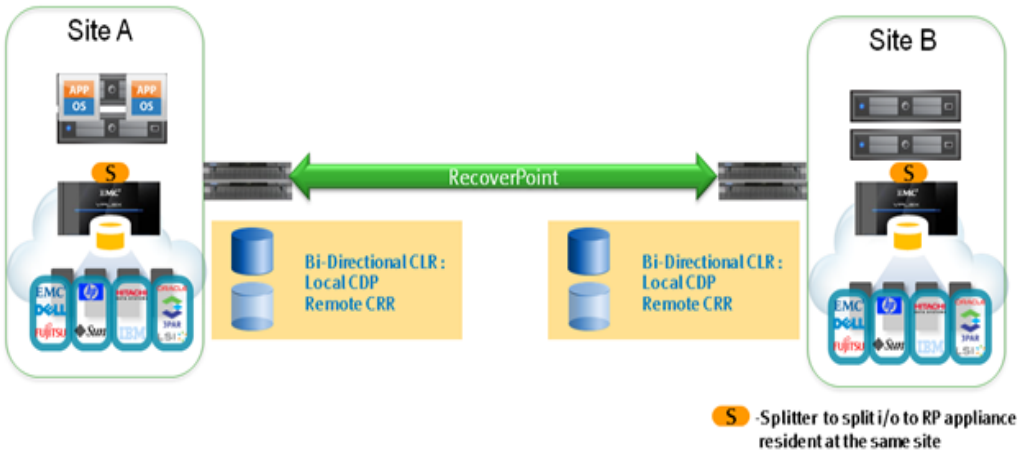


Figure 23 VPLEX Local and RecoverPoint CLR - remote site is independent VPLEX cluster

or, the remote site can be an array-based splitter:

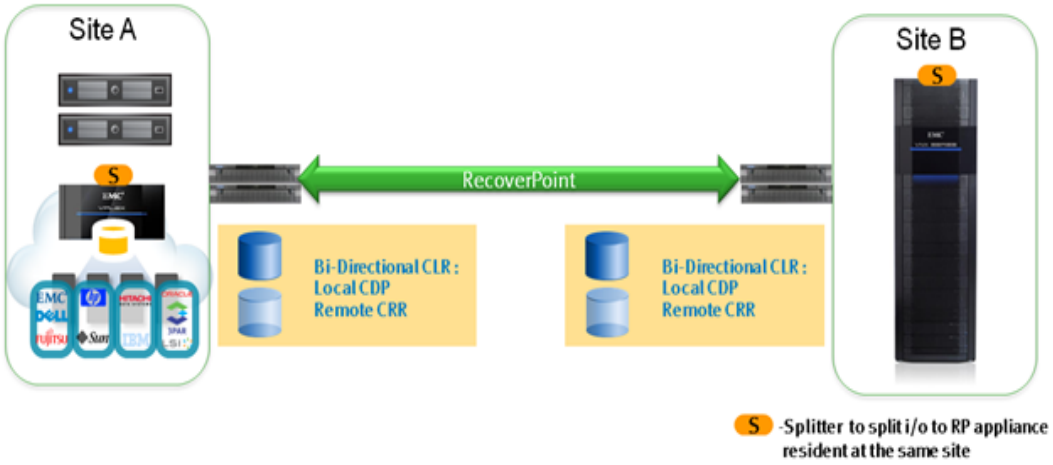


Figure 24 VPLEX Local and RecoverPoint CLR - remote site is array-based splitter

VPLEX Metro and RecoverPoint CDP at one site

In VPLEX Metro/RecoverPoint CDP configurations, I/O is split to replica volumes located at only one VPLEX cluster.

RPAs are deployed at one VPLEX cluster:

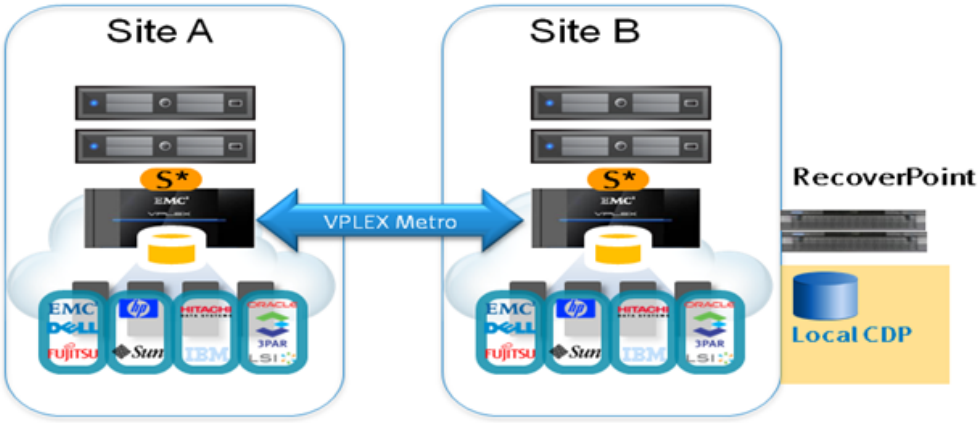


Figure 25 VPLEX Metro and RecoverPoint CDP

VPLEX Metro/RecoverPoint CDP configurations support unlimited points in time on VPLEX distributed and local virtual volumes.

Users can quickly return to any point-in-time, in order to recover from operational disasters.

VPLEX Metro and RecoverPoint CRR/CLR

In VPLEX Metro/RecoverPoint CRR/CLR configurations, I/O is:

- Written to both VPLEX clusters (as part of normal VPLEX operations).
- Split on one VPLEX cluster to replica volumes located both at the cluster and at a remote site.

RPAs are deployed at one VPLEX cluster and at a third site.

The third site can be an independent VPLEX cluster:

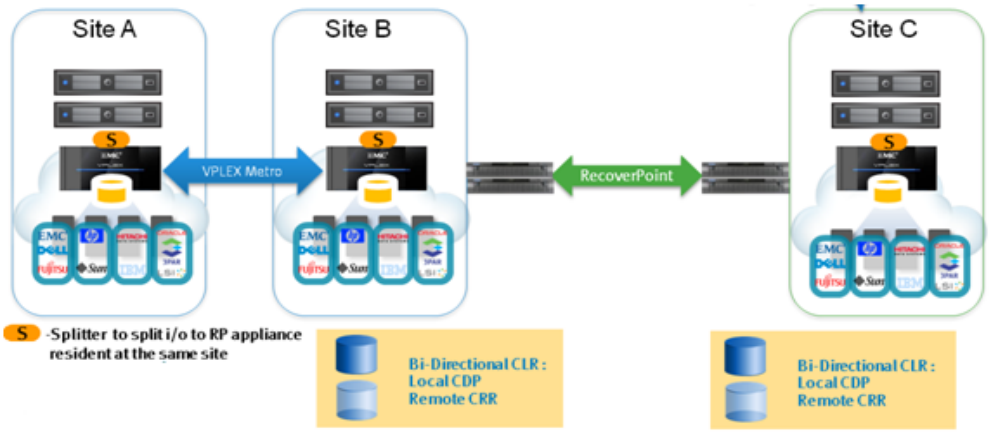


Figure 26 VPLEX Metro and RecoverPoint CLR - remote site is independent VPLEX cluster

or, the remote site can be an array-based splitter:

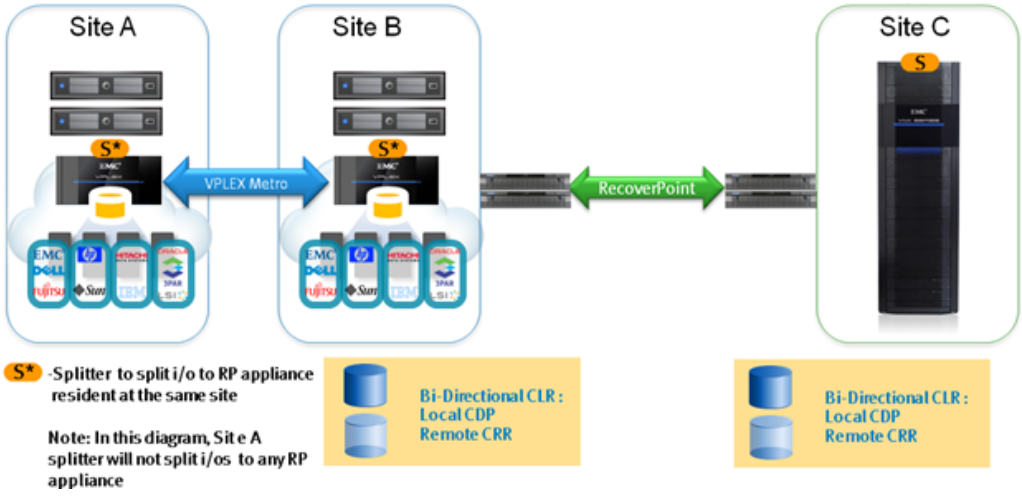


Figure 27 VPLEX Metro and RecoverPoint CLR/CRR - remote site is array-based splitter

Although the RecoverPoint splitter is resident in all VPLEX clusters, only one cluster in a VPLEX Metro can have RPAs deployed.

This configuration supports unlimited points in time, (with granularity up to a single write) for local and distributed VPLEX virtual volumes.

- RecoverPoint Appliances can be deployed at only one VPLEX cluster in a Metro configuration.
- All RecoverPoint-protected volumes must be on the preferred cluster, as designated by VPLEX consistency group-level detach rules.
- Customers can recover from operational disasters by quickly returning to any PIT on the VPLEX cluster where the RPAs are deployed or at the third site.
- Application event aware based rollback is supported on VPLEX Metro distributed/local virtual volumes for Microsoft SQL, Microsoft Exchange and Oracle database applications.
- If the VPLEX cluster fails, customers can recover to any point in time at the remote (third) site. Recovery at remote site to any point in time can be automated through integration with MSCE and VMware Site Recovery Manager (SRM). See vCenter Site Recovery Manager support for VPLEX on page 228.
- This configuration can simulate a disaster at the VPLEX cluster to test RecoverPoint disaster recovery features at the remote site.

Shared VPLEX splitter

The VPLEX splitter can be shared by multiple RecoverPoint clusters. This allows data to be replicated from a production VPLEX cluster to multiple RecoverPoint clusters.

Figure 28 Shared VPLEX splitter

Up to 4 RecoverPoint RPA clusters can share a VPLEX splitter.

Shared RecoverPoint RPA cluster

The RecoverPoint RPA cluster can be shared by multiple VPLEX sites:

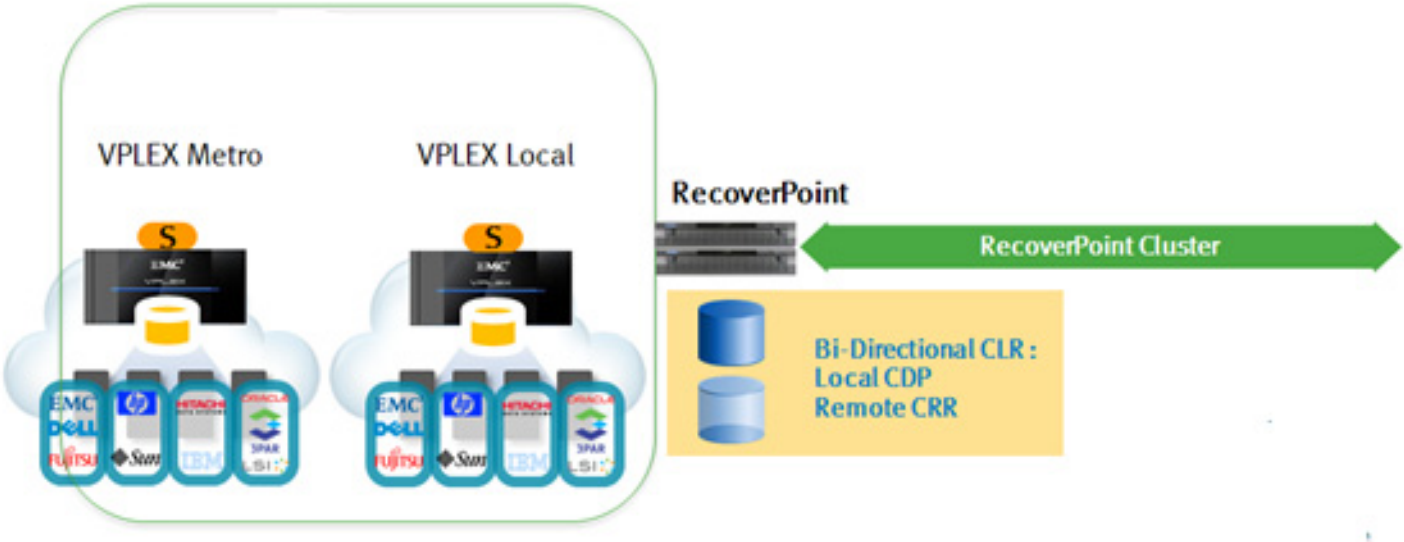


Figure 29 Shared RecoverPoint RPA cluster

RecoverPoint replication with CLARiiON

VPLEX and RecoverPoint can be deployed in conjunction with CLARiiON-based RecoverPoint splitters, in both VPLEX Local and VPLEX Metro environments.

In the configuration depicted in Figure 30, host writes to VPLEX Local virtual volumes are written to both legs of RAID 1 devices. The VPLEX splitter "splits" the writes, sending one copy to the usual back-end storage, and one copy across a WAN to a CLARiiON array at a remote disaster recovery site:

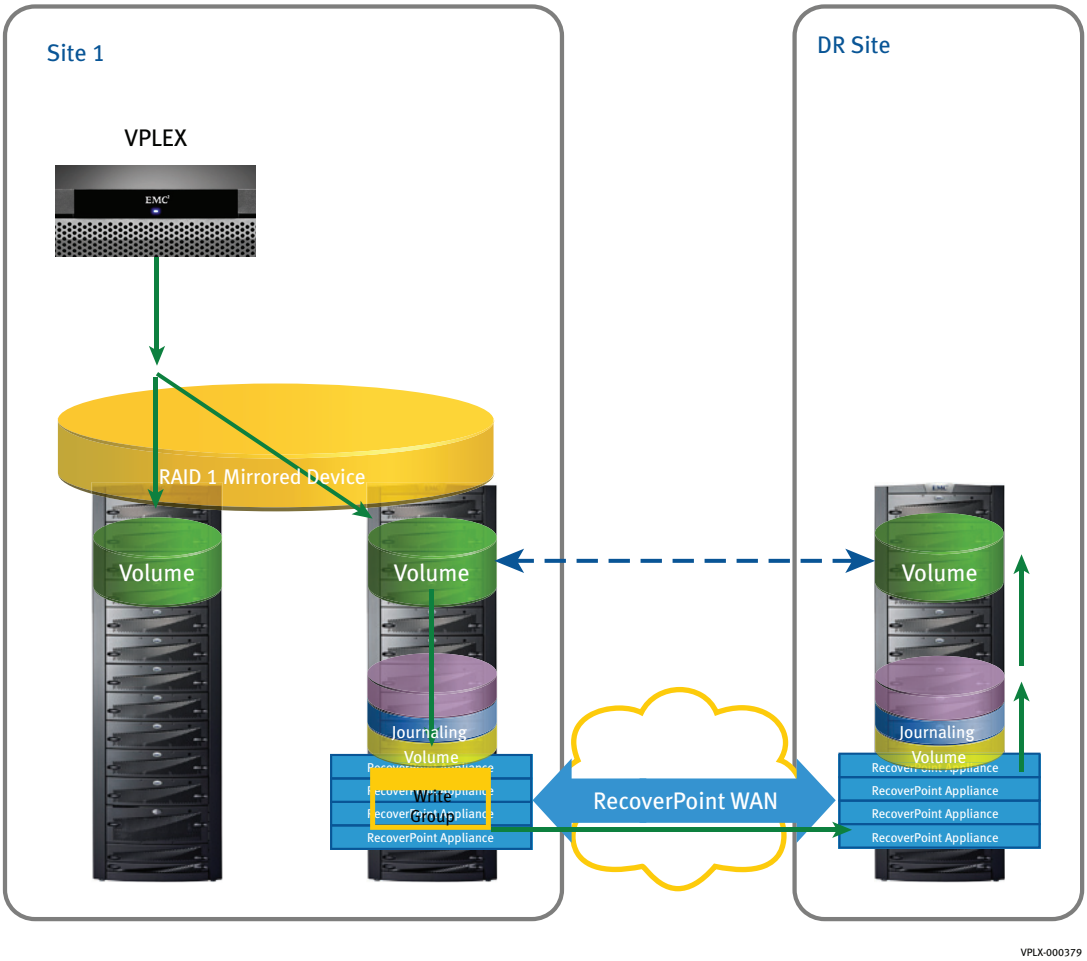


Figure 30 Replication with VPLEX Local and CLARiiON

In the configuration depicted in Figure 31, host writes to VPLEX Metro distributed virtual volumes are similarly split, sending one copy to each of the VPLEX clusters, and a third copy across a WAN to a CLARiiON array at a remote disaster recovery site:

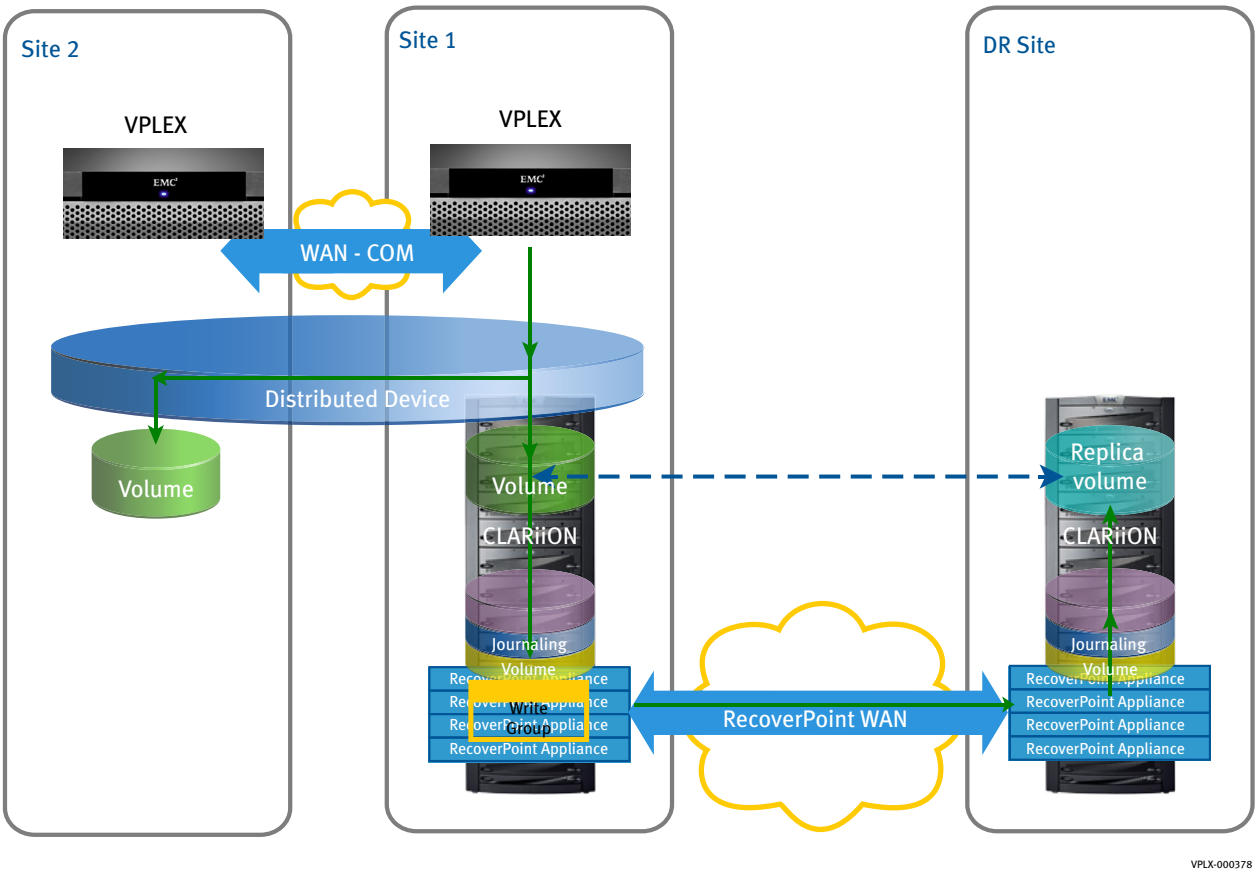


Figure 31 Replication with VPLEX Metro and CLARiiON

vCenter Site Recovery Manager support for VPLEX

With RecoverPoint replication, you can add Site Recovery Manager support to VPLEX.

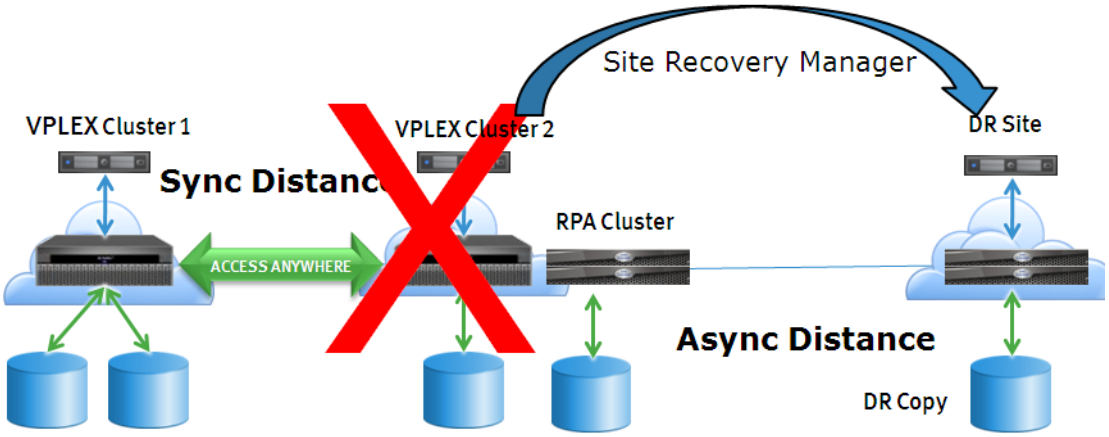


Figure 32 Support for Site Recovery Manager

When an outage occurs in VPLEX Local or VPLEX Metro configurations, the virtual machine(s) can be restarted at the DR Site with automatic synchronization to the VPLEX configuration when the outage is over.

RecoverPoint CLI context

RecoverPoint does not appear in the VPLEX CLI until the client has been configured.

Complete instructions to install RecoverPoint are available in the VPLEX generator.

Descriptions of the CLI commands to add RecoverPoint are available in the VPLEX CLI Guide.

Use the **rp rpa-cluster-add** command to add a RecoverPoint RPA cluster:

```
VPlexcli:/> rp rpa-cluster add -o 10.6.210.79 -u admin -c cluster-1
Enter rpa-cluster administrative password: Admin-password
Enter rpa-cluster administrative password again for verification:
   Admin-password
```

Once the RecoverPoint cluster is added, a CLI context called "recoverpoint" appears in the CLI context tree:

```
VPlexcli:/> ll /recoverpoint/

/recoverpoint:
Name          Description
------------  ------------------------------------------------------------
rpa-clusters  Contains all the 'clusters' of RecoverPoint Appliances registered
              in the system.

VPlexcli:/> ll /recoverpoint/rpa-clusters

/recoverpoint/rpa-clusters:
RPA Host     VPLEX Cluster  RPA Site  RPA ID  RPA Version
-----------  -------------  --------  ------  -----------
10.6.210.75  cluster-1      Site1     RPA 1   3.5(l.26)

VPlexcli:/> ll /recoverpoint/rpa-clusters/10.6.210.75

/recoverpoint/rpa-clusters/10.6.210.75:

Attributes:
Name                    Value
----------------------  -----------
admin-username          admin
config-changes-allowed  true
rp-health-indications   []
rp-health-status        ok
rp-software-serial-id   -
rpa-host                10.6.210.75
rpa-id                  RPA 1
rpa-site                Site1
rpa-version             3.5(l.26)
vplex-cluster           cluster-1


Contexts:
Name     Description
-------  -----------
volumes  -


VPlexcli:/> ll /recoverpoint/rpa-clusters/10.6.210.75/volumes

/recoverpoint/rpa-clusters/10.6.210.75/volumes:
Name                    RPA    RP Type     RP Role     RP     VPLEX       Capacity
----------------------  Site   ----------  ----------  Group  Group       --------
----------------------  -----  ----------  ----------  -----  ----------  --------
RP_Repo_Vol_vol         Site1  Repository  -           -      RPTestCg    12G
```

```
demo_DR1_vol             Site1  Replication  Production   cg1    DemoSyncCG  5G
                                             Source
demo_Replica_vol         Site1  Replication  Local        cg1    Demo_Repli  5G
                                             Replica
demo_prodjournal_1_vol   Site1  Journal      -            cg1    RPTestCg    5G
demo_prodjournal_2_vol   Site1  Journal      -            cg1    RPTestCg    5G
demo_prodjournal_3_vol   Site1  Journal      -            cg1    RPTestCg    5G
.
.
.
VPlexcli:/> ll /recoverpoint/rpa-clusters/10.6.210.75/volumes/demo_DR1_vol

/recoverpoint/rpa-clusters/10.6.210.75/volumes/demo_DR1_vol:
Name                         Value
---------------------------  -------------------------------
rp-consistency-group         cg1
rp-consistency-group-copy    prod
rp-replication-set           RSet 1
rp-role                      Production Source
rp-type                      Replication
rpa-site                     Site1
size                         5G
uid                          6000144000000010e03ec4a5dfffd8c3
vplex-cluster                cluster-1
vplex-consistency-group      DemoSyncCG
```

Configuration/operation guidelines

In VPLEX Metro configurations, RecoverPoint Appliances (RPAs) can be configured at only one VPLEX cluster. Data from the cluster where the RPAs are configured is replicated to the peer VPLEX cluster (by VPLEX), and to a third site (by RecoverPoint).

Virtual image access (see Image access on page 219) is not supported.

Device migrations between two VPLEX clusters are not supported if one leg of the device is replicated by RecoverPoint.

If a VPLEX director and a RecoverPoint Appliance are restarted at the same time, a full sweep may occur.

An RPA cluster supports up to 5,000 volumes exported to it from all storage arrays that it can see, and up to 280,000 ITL paths.

RecoverPoint repository volume

One RecoverPoint repository volume is required for each RPA cluster attached to VPLEX. The repository volume serves all RPAs of the particular cluster and splitters associated with that cluster.

The repository volume should be configured on a VPLEX virtual volume with the following attributes:

- 5.72 GB. Capacity beyond 5.72 GB is not used.
- RAID-1 device based on volumes in two different arrays
- Provisioned on non-thin storage
- Accessible to all RPAs at the site
- A local volume on the VPLEX cluster where the RPAs are configured
- Cannot be distributed

When VPLEX is installed into an existing RecoverPoint configuration, the repository volume is already configured. There is no procedure to move a non-VPLEX repository volume. When installing VPLEX into an existing RecoverPoint configuration, continue using the non-VPLEX repository volume. When the storage array that hosts the repository volume is refreshed, move the repository volume to VPLEX-hosted storage.

Once the repository volume is in VPLEX, it can be moved as needed.

Replica volumes

Performance of replica volumes should be the same or faster than their associated production volumes.

- RecoverPoint CDP requires one replica volume on the production site for each production volume.
- RecoverPoint CRR requires one replica volume on the remote site for each production volume.
- RecoverPoint CLR requires one replica volume on production site AND one on the remote site for each production volume.

Journal volumes

Performance is crucial for journal volumes.

Configure journal volumes on the fastest storage available.

RecoverPoint journal volumes must be local volumes on the VPLEX cluster where the RPAs are configured.

RecoverPoint journal volumes cannot be distributed volumes.

Refer to the *EMC RecoverPoint Administrator's Guide* for information about sizing journal volumes.

VPLEX volumes

- Only VPLEX volumes that are members of a VPLEX consistency group with the **recoverpoint-enabled** attribute set to "true" can be replicated by RecoverPoint. See recoverpoint-enabled on page 152.
- Virtual volumes that are used by RecoverPoint cannot be expanded.

  Configure virtual volume size accordingly.
- Only local and distributed volumes can be replicated using RecoverPoint - not remote volumes.
- The following volumes may not be added to a VPLEX storage view created to support RecoverPoint:
  - Remote volumes
  - Volumes already in a different RecoverPoint storage view
  - Volumes in VPLEX consistency groups whose members are in a different RecoverPoint storage view

Note: A RecoverPoint cluster may take up to 2 minutes to take note of changes to VPLEX consistency groups.

Wait for 2 minutes after making the following changes before creating or changing a RecoverPoint consistency group:

- Adding or removing virtual volumes to/from a VPLEX consistency group
- Enabling/disabling the recoverpoint-enabled property of a VPLEX consistency group
- Changing the detach rule of a VPLEX consistency group

VPLEX consistency groups

- Only VPLEX volumes that are members of VPLEX consistency groups with the **recoverpoint-enabled** attribute set to "true" can be replicated by RecoverPoint. See recoverpoint-enabled on page 152.
- VPLEX supports a maximum of 128 recoverpoint-enabled consistency groups.
- Changes to VPLEX consistency groups may take up to 2 minutes to register in RecoverPoint.
- The cluster at which RPAs are configured must be the preferred cluster as designated by the VPLEX consistency group detach rules. See detach-rule on page 150.

Administrators and VPLEX Witness can override the preferred cluster designation during inter-cluster link outages. See Failures in Metro Systems: With VPLEX Witness on page 186.

For Production source volumes, this allows normal high availability behavior. Writes to the preferred cluster are logged for the duration of the link outage. When the link is restored, a log rebuild copies only the logged changes to the splitter.

No snapshots are kept during the rebuild.

- All Production source volumes for a given RecoverPoint consistency group should be in one VPLEX consistency group. All the Replica volumes for a given RecoverPoint consistency group should be in another VPLEX consistency group (not the same VPLEX consistency group as the Production source volumes).

  If a RecoverPoint consistency group with a Production source volume and a Replica volume fails over, the Replica becomes the Production source. Having the replica in a separate VPLEX consistency group assures that the hosts that the replica now services (as the Production source volume) is write-order ensured from a VPLEX standpoint.
- Replica volumes and Production source volumes may be on non-VPLEX storage, or on a separate VPLEX system.
- If a virtual volume is in a RP consistency group with the same role as other virtual volumes in the VPLEX consistency group, it should be added to the VPLEX consistency group.

VPLEX NDU

IMPORTANT

Confirm that the new version of VPLEX GeoSynchrony is compatible with the version of RecoverPoint. Refer to the *EMC VPLEX Release Notes* for the GeoSynchrony release.

When a VPLEX cluster is being upgraded, it puts any RecoverPoint system it can talk to into maintenance mode.

When the upgrade is complete, VPLEX takes the RecoverPoint system out of maintenance mode.

When two VPLEX systems are connected to the same RecoverPoint cluster, the VPLEX systems cannot be upgraded at the same time. Upgrade on the second VPLEX fails until upgrade on the first VPLEX completes.

CAUTION

Do not upgrade VPLEX systems connected to the same RecoverPoint replication environment at the same time.

Storage failures

- If a storage volume (at the VPLEX cluster where the RPAs are configured) fails, the host is still writing to the volume, and those writes are not included in the snapshots of the VPLEX consistency group, and won't arrive until the rebuild occurs.

- The RecoverPoint site on a replica volume always becomes a winner (regardless of VPLEX Witness or site restart), even if this triggers a conflicting detach (winner at both sides).

  In the case of a conflicting detach, as long as there is an active splitter (i.e. replication is not disabled), its host cluster must be the "datawin" target if the administrator chooses to resolve the conflict

- In the case of storage failure of the mirror leg of a replica volume at the RecoverPoint site, there must be a rebuild to correct the data on the replica. This will put the replica volume into tracking mode (overriding fail-all), and rebuild I/O is allowed to proceed.

Logged image access

During logged image access (see Image access on page 219), the user selects a snapshot (PIT). The RecoverPoint system reverts the replica volume to the selected PIT using the following steps:

- Stores the current state of the replica storage in the replica journal
- Reads historical data from the replica journal
- Re-writes the replica storage to reflect that historical data

When logged image access is disabled, this process is reversed.

When logged image access is first enabled, the RecoverPoint GUI displays "Enabling Logged Access". Once logged image access is enabled, the display changes to "Logged access (*n*% full)".

Click the Journal tab => target copy to display how much data remains to be written and remaining time to complete the writes.

Time to complete depends on the amount of data and on the performance of the journal volume.

For example:

- To write 963GB of data;
- At a rate of 91MB/sec

Expect the time to complete as ~3.5 hours.

Zoning

Refer to *RecoverPoint Deploying with VPLEX™ Technical Notes* for detailed information. The following information is overview and guidance only.

Zoning is specific to site and topology. Best practice is to use the RPA Fibre Channel ports as both initiators and targets to achieve maximum performance, redundancy, and optimal use of resource.

CAUTION

If, due to Initiator-Target LUN (ITL) limitations or other non-RecoverPoint considerations, you need to zone RPA Fibre Channel ports in either the initiator zone or the target zone, but not both, there are minor differences in performance and availability.

Initiator-target separation is not supported when:

- Multiple splitter types are configured at the same RPA site
- Using remote replication over FibreChannel
- Using distributed RecoverPoint consistency groups

CAUTION

When using Fibre Channel ports that can be both initiators and targets, best practice is to add all initiator and target ports to the same zone.

About zoning: VPLEX and RecoverPoint upgrades

If you are upgrading a VPLEX system from release 5.0.x or earlier and your RecoverPoint system is at Release 3.4 or earlier, ensure that the VPLEX backend and RecoverPoint systems are NOT zoned together.

Zone the two systems only after both are upgraded; VPLEX to 5.1 or later, and RecoverPoint to 3.5 or later.

If a Release 5.1 VPLEX system is zoned to a pre-3.5 RecoverPoint system, the VPLEX system will not recognize when RecoverPoint is upgraded to Release 3.5.

About zoning: RPAs to VPLEX FE ports

RecoverPoint requires read and write access to VPLEX virtual volumes.

All RecoverPoint Fibre Channel ports must be zoned to at least two front-end ports on each VPLEX director of the VPLEX where splitting occurs.

All RecoverPoint appliances should be zoned to the same VPLEX front-end ports.

The following zoning rules apply:

- Each RPA must have at least two paths to the VPLEX cluster.
- For non-disruptive upgrades of RecoverPoint, each RecoverPoint appliance must have at least one path to an A director and one path to a B director.
- Each RecoverPoint appliance should have at least two physical connections to the front-end fabric switch.
- Each RPA should be zoned to provide paths to every virtual volume via at least two directors.

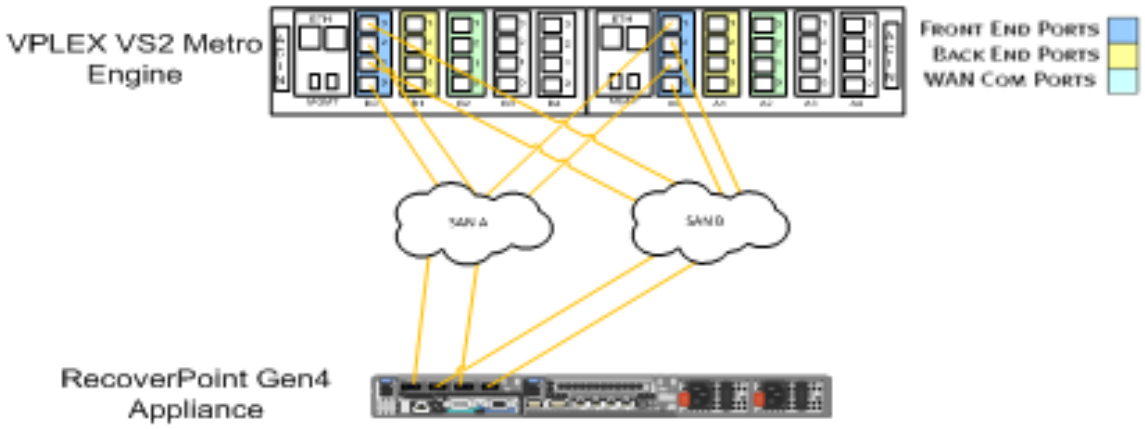


Figure 33 Figure 1RPA to VPLEX Front End zoning

About zoning: RPAs to VPLEX BE ports

The RecoverPoint VPLEX splitter splits all incoming I/Os from both the front-end ports and from the remote VPLEX cluster (in the Metro configuration). The VPLEX splitter sends a copy of each write to the RPAs over the back-end ports.

Thus, every RecoverPoint HBA must be zoned to at least two VPLEX back-end ports on each director of the VPLEX cluster where the RPA cluster is installed.

All RPAs should be zoned to the same VPLEX back-end ports. The RPAs are seen by VPLEX as storage targets (receiving writes); therefore, the appliances must be zoned according to the rules of zoning storage arrays:

- Each director in the VPLEX cluster must have redundant I/O paths to every RPA.
- Each director must have redundant physical connections to the back-end fabric switches.

Note: Each director supports a maximum of four paths to any one RPA.

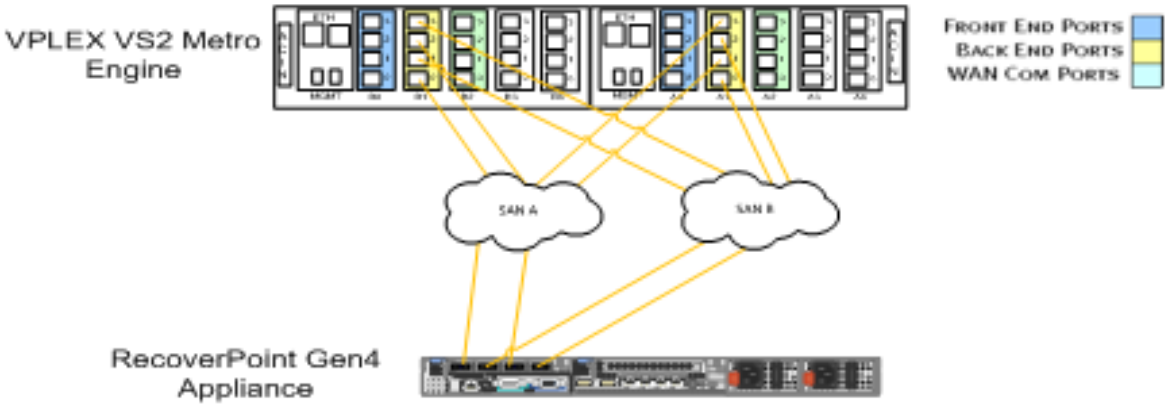


Figure 34 RPA to VPLEX Back End zoning

About zoning: Distributed RecoverPoint consistency groups

Distributed RecoverPoint consistency groups require that all RecoverPoint Fibre Channel ports in the fabric can communicate with each other.

To do so, they must all be in one zone.

Note: RecoverPoint-to-VPLEX back-end zoning usually satisfies this requirement.

About zoning: Storage-to-VPLEX

For VPLEX to access physical storage, arrays must be zoned to the VPLEX back-end as follows:

- Each director must have redundant I/O paths to every back-end storage array.

  Best practice is that each director has redundant physical connections to the back-end storage fabrics.

- Each storage array must have redundant controllers.

  Best practice is that at least two ports of each controller are connected to the back-end fabric.

  Each director supports a maximum of 4 paths per volume.

Management tools

RecoverPoint can be managed using either a Command Line Interface (CLI) or Graphical User Interface (GUI).

RecoverPoint CLI

Use the RecoverPoint CLI to manage and monitor activities interactively, or through scripts.

There are two main modes of work in a CLI session:

CLI mode - Get help and interact with the system using CLI commands. CLI mode includes:

- **help mode** - retrieve information regarding each CLI command, its parameters and usage.
- **interactive mode** - guide the user when running single commands, allowing them to view each command parameter and its possible values while running the command.
- **expert mode** - input multiple parameters and their values for a single command.

Script mode - Interact with the system using scripts containing CLI commands.

To open a session with a RecoverPoint cluster or RPA and communicate using the CLI, create an SSH connection to either the site management IP (preferred, for a RecoverPoint cluster) or IP address of a specific RPA configured during a RecoverPoint installation.

The tools to work with the RecoverPoint CLI vary depending on which mode is used to access the CLI.

- For CLI mode, download, install the free SSH connection utility PuTTY for Linux or Unix.
- For Script mode, use the SSH utility that comes with the operating system, to run scripts and commands.

For more information about the command-line interface, see the *EMC RecoverPoint Command Line Interface Reference Guide*.

RecoverPoint GUI

The RecoverPoint Graphical User Interface (GUI) is a java-based Web application. The GUI enables the user to monitor and manage a single RecoverPoint cluster.

To open a session with a RecoverPoint cluster or RPA:

- Open a browser window
- Type the site management IP (to connect to a RecoverPoint cluster) or IP address of a specific RPA configured during a RecoverPoint installation, into the address bar.
- Supply your RecoverPoint login credentials.

13

Performance and monitoring

This chapter describes RPO/RTO and the procedures to create and operate performance monitors.

About performance

This chapter describes 2 topics related to performance on VPLEX systems:

- **Configuration** - Modifiable parameters to maximize performance and to manage Recovery Point Objective (RPO) and Recovery Time Objective (RTO) for VPLEX Geo systems.
- **Monitoring** - Tools and techniques to monitor VPLEX's performance, and to identify and diagnose problems.

RPO and RTO

Recovery Point Objective (RPO): RPO is the time interval between the point of failure of a storage system and the expected point in the past to which the storage system is capable of recovering customer data.

RPO is a maximum amount of data loss that can be tolerated by the application after a failure. The value of the RPO is highly dependent upon the recovery technique used. For example, RPO for backups is typically days; for asynchronous replication minutes; and for mirroring or synchronous replication seconds or instantaneous.

Recovery Time Objective (RTO): RTO is the time duration within which a storage solution is expected to recover from failure and begin servicing application requests.

RTO is the longest tolerable application outage due to a failure of a storage system. RTO is a function of the storage technology. It may measure in hours for backup systems, minutes for a remote replication, and seconds (or less) for a mirroring.

Calculating RPO in VPLEX Geo systems

VPLEX Geo systems, distributed volumes must be members of consistency groups with asynchronous cache-mode.

Two additional parameters specific to asynchronous consistency groups impact the RPO:

- "default closeout-time" on page 154
- "maximum-queue-depth" on page 154

The default closeout-time and maximum-queue-depth properties work together to configure the maximum possible data loss in the event of an inter-cluster link outage.

In general:

RPO = roll_back_data (MB) / drain_rate (MB/s)

The maximum value for roll_back_data is computed as:

(max_queue_depth - 2) * delta_size * N

- 2 is subtracted from the maximum_queue_depth to exclude the WRITING and COMMIT deltas which are never discarded (rolled back) during failure recovery. It is the unprotected deltas in states OPEN, CLOSED, or EXCHANGED that may be lost.
- N is the number of directors at a cluster exporting volumes in Asynchronous Consistency Groups.

- The maximum drain rate is the rate at which cache pages can be exchanged and written to back-end storage, the rate at which deltas can be protected. The maximum drain rate is a function of the inter-cluster WAN speed and backend storage-array performance.

Table 11 Parameters impacting RPO in Geo configurations

Parameter	Default	Maximum	Configurable?
Number of clusters	2	2	-
Number of asynchronous consistency groups	0 - 16	16	-
Maximum queue depth	6	64	Yes
Delta size	16 MB	16 MB	No
Closeout time	30 seconds	0 (no closeout time)	Yes

Note: Table 11 assumes the same number of engines configured in each cluster.

To calculate RPO, apply the drain rate. For example:

Assuming a drain rate of 100 MB/s:

- 1 engine clusters = 4 GB maximum loss = ~42 seconds
- 2 engine clusters = 8 GB maximum loss = ~ 1 minute 22 seconds
- 4 engine clusters = 16 GB maximum loss = ~2 minutes 44 seconds

Table 12 shows the impact of increasing the maximum queue depth from the default value of 6 to the maximum value of 64 for systems with the maximum number of asynchronous consistency groups (16) configured:

Table 12 Maximum roll back loss for Geo with 16 asynchronous consistency groups

Engines/ cluster	Total # directors	Max queue depth	Max roll back loss
1	4	6 (default)	4.0 GB
1	4	64	62.0 GB
2	8	6	8.0 GB
2	8	64	124.0 GB
4	16	6	16.0 GB
4	16	64	248.0 GB

About performance monitoring

Performance monitors collect and displays statistics to determine how a port or volume is being used, how much I/O is being processed, CPU usage, and so on.

Performance monitoring is supported in both the VPLEX CLI and GUI, and falls into three general types:

- Current load monitoring allows administrators to watch CPU load during upgrades, I/O load across the inter-cluster WAN link, and front-end vs. back-end load during data mining or back up.

  Current load monitoring is supported in the GUI.

- Long term load monitoring collects data for capacity planning and load balancing.

  Long term load monitoring is supported by monitors created in the CLI and/or perpetual monitors.

- Troubleshooting monitoring helps identify bottlenecks and resource hogs.

  Troubleshooting monitors are supported by monitors created in the CLI and/or perpetual monitors.

Note: In the GUI, performance statistics are displayed per cluster. To view statistics for both clusters in a Metro or Geo, connect to both clusters.

Custom monitors

You can use the CLI to create custom monitors to collect and display selected statistics for selected targets.

See "Monitor performance using the CLI" on page 245.

Preconfigured monitors

Use the **report create-monitors** command to create three pre-configured monitors for each director.

See "Pre-configured performance monitors" on page 254.

Perpetual monitors

Starting in Release 5.0, GeoSynchrony includes perpetual monitors that gather a standard set of performance statistics every 30 seconds.

Note: Perpetual monitors do not collect per volume statistics.

Perpetual monitor files are collected as part of collect-diagnostics. Collect-diagnostics is per cluster, so in Metro or Geo configurations, you must run the command from both VPLEX management servers.

Output of perpetual monitors is captured in the file smsDump_<date>.zip inside the base collect-diagnostics zip file.

Within smsDump_<date>.zip file, monitor files are in clilogs/.

You can also copy the perpetual files from the management server. They are located in /var/log/VPlex/cli/. There is one perpetual monitor file per director, identifiable by the keyword "PERPETUAL"

For example: director-1-1-A_PERPETUAL_vplex_sys_perf_mon.log

Performance monitoring using the Unisphere for VPLEX GUI

The performance monitoring dashboard provides a customized view into the performance of your VPLEX system. You decide which aspects of the system's performance to view and compare.

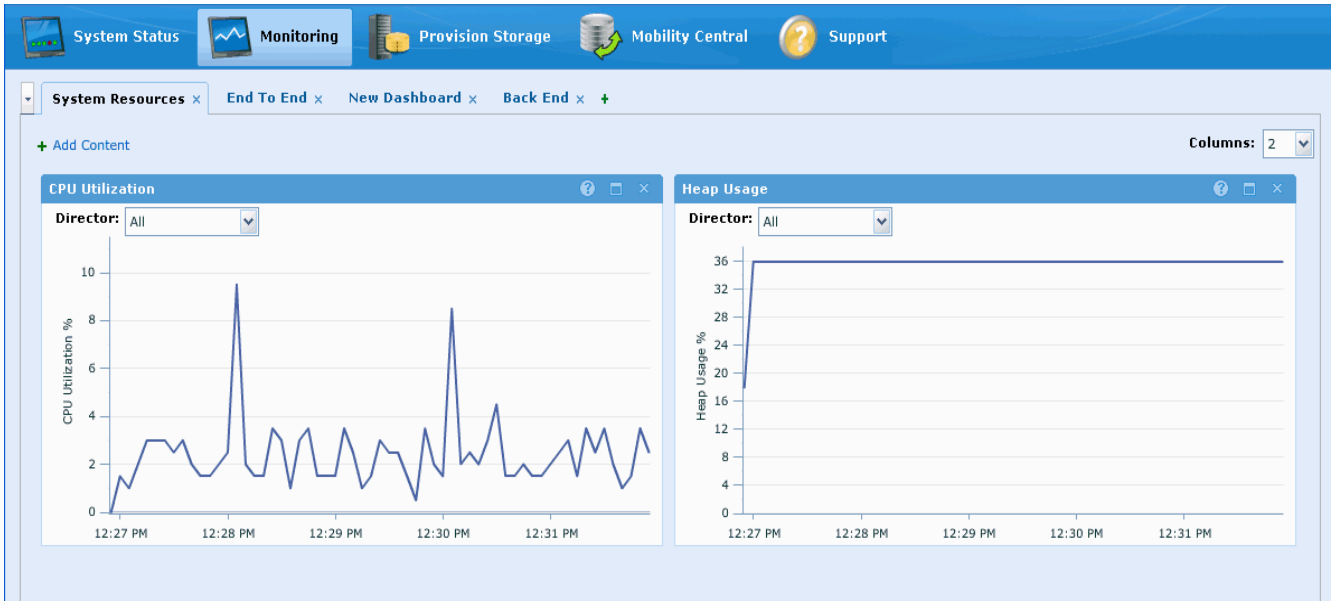


Figure 35 Performance monitoring dashboard

Performance information for the current 5-minute window is displayed as a set of charts, including:

- **WAN Link Performance chart -** Shows the WAN link performance for the cluster to which you are connected. Use this chart to monitor link performance to help determine the bandwidth requirements for your specific environment, gather statistical data over time, monitor network traffic during peak periods, or to plan data mobility jobs to avoid peak usage times.
- **WAN Link Usage chart** - Shows the WAN link usage for the cluster to which you are connected. Use this chart to monitor link usage to help determine the bandwidth requirements for your specific environment, gather statistical data over time, monitor network traffic during peak periods, or to plan data mobility jobs to avoid peak usage times. The chart allows you to separately monitor the amount of bandwidth used for normal system operations, writes to distributed volumes, and mobility jobs.
- **Back-end Latency chart** - Provides details of the back-end latency statistics for your VPLEX system in graphical form over time. The chart allows you to view current or historical performance data that you can use to monitor peaks in workload, detect performance issues, or view what was happening in the system when a specific problem occurred.
- **Front-end Latency chart** - Provides details of the front-end latency statistics for your VPLEX system in graphical form over time. The chart allows you to view current or historical performance data that you can use to monitor peaks in workload, detect performance issues, or view what was happening in the system when a specific problem occurred.

- **Rebuild Status chart** - Displays the status of any rebuilds that are running on your VPLEX system.
- **CPU Utilization chart** - Provides a time-based view of the utilization load on the primary director CPU on your VPLEX system. By default, the chart shows an averaged view of the utilization loads of all the directors in your VPLEX system.
- **Front-end Aborts chart** - Displays the number of aborts per second over time for directors on your VPLEX system. By default, the chart shows averaged front-end aborts for the VPLEX system.
- **Subpage Writes chart** - Displays the percentage of subpage writes over time for directors on your VPLEX system. By default, the chart shows an averaged subpage writes chart for the VPLEX system.
- **Front-End Throughput chart** - Displays the front-end I/Os per second over time for directors on your VPLEX system. By default, the chart shows the total front-end throughput for the VPLEX system.
- **Front-End Bandwidth chart** - Displays the quantity of front-end reads and writes per second over time for directors on your VPLEX system. By default, the chart shows the total front-end bandwidth for the VPLEX system.

Performance monitoring using the VPLEX CLI

Use the CLI to create custom monitors to help diagnose performance issues.

Two CLI objects collect and display performance statistics:

- **monitors** - Gather the specified statistic from the specified target at the specified interval.
- **monitor sinks** - Direct the output to the desired destination. Monitor sinks include the console, a file, or a combination of the two.

Note: SNMP statistics do not require a monitor or monitor sink. Use the **snmp-agent configure** command to configure and start the SNMP agent. Refer to "Performance statistics retrievable by SNMP" on page 272.

Monitor performance using the CLI

This section describes the steps to create a custom monitor using the VPLEX CLI.

About file rotation and timestamps

Starting in Release 5.1, the log files created by a monitor's file sink are automatically rotated when they reach a size of 10 MB. The 10MB file is saved as *filename*.csv.*n* where *n* is a number 1 - 10, and output is saved in a new file named *filename*.csv.*n*+1.

The .csv files are rotated up to 10 times.

In the following example, a monitor has exceeded 10MB of output. The initial 10MB are stored in *filename*.csv.1. Subsequent output is stored in *filename*.csv.

```
service@sms-cluster-1:/var/log/VPlex/cli> ll my-data.csv*
   -rw-r--r-- 1 service users 2910722 2012-03-06 21:23 my-data.csv
   -rw-r--r-- 1 service users 10566670 2012-03-06 21:10 my-data.csv.1
```

If the second file exceeds, 10MB:

- The previous filename.csv.1 is changed to filename.csv.2
- The filename.csv changes to filename.csv.1
- Subsequent output is stored in filename.csv

Up to 10 such rotations, and numbered .csv files are supported.

When the file sink is removed or the monitor is destroyed, output to the .csv file stops, and the current .csv file is is timestamped. For example:

```
service@sms-cluster-1:/var/log/VPlex/cli> ll my-data.csv*
   -rw-r--r-- 1 service users 10566670 2012-03-06 21:23 my-data.csv.1
   -rw-r--r-- 1 service users  5637498 2012-03-06 21:26
   my-data.csv_20120306092614973
```

Procedure overview: create a monitor using the CLI

To create and operate a monitor using the CLI, use the following general steps:

1. Determine the type of statistic to collect from the target object.

   Use the **monitor stat-list** ***category*** and/or the **monitor stat-list *** command to display the statistics to include in the monitor.

   Refer to the tables in "Statistics" on page 260 for lists of statistics by category.

   Note whether the statistic you want to collect requires a target to be specified.

   Note: Specify only one type of target per monitor. For example, you cannot create a monitor that includes both port and storage-volumes as targets.

2. Determine how often the monitor should collect statistics.
3. Use the **monitor create** command to create a monitor.
4. Use the **monitor add-sink** commands to add one or more sinks to the monitor.

   Add a console sink to send performance data to the VPLEX management console.

   Add a file sink to send performance data to a specified file.
5. Repeat Steps 3 and 4 for each director.

6. The monitor begins operation (polling and collecting performance data) when the sink is added to the monitor.

   To disable automatic polling without deleting the monitor or its sink(s), do one of the following:

   - Use the **set** command to change the monitor's **period** attribute to 0.
   - Use the **set** command to change the sink's **enabled** attribute to false.

7. Use the **monitor collect** command to update/collect statistics immediately without waiting for the monitor's next automatic collection.

8. Monitor output.

   Console sinks display monitor output on the console.

   For file sinks, navigate to /var/log/VPlex/cli/ on the management server and use the **tail -f** ***filename*** to display the output,

   or:

   Send output to a csv file, open the file in Microsoft Excel and create a chart.

WARNING

Do NOT edit the CSV file in Microsoft Excel, and then save the file. Excel removes the seconds field, resulting in duplicate timestamps. Use Excel to look at the CSV files, but don't save any edits.

9. Use the **monitor destroy** command to remove the monitor.

Create a monitor

Use the **monitor create** command to create a monitor and specify the statistics collected by the monitor.

The syntax for the command is:

```
monitor create
   [-p|--period] collection-period
   [-n|--name] monitor-name
   [-d|--director] context-path,context-path
   [-s|--stats] stat,stat,stat,stat
   [-t|--targets] context-path,context-path
```

Required arguments

--name - Name of the monitor.

--stats ***stat,stat,stat*** - One or more statistics to monitor, separated by commas. Examples of statistics (*stat*s) include:

- **be aborts**
- **be resets**
- **be timeouts**

See the Online Help for a complete list of available performance monitor statistics.

Optional arguments

-- period - Frequency at which this monitor collects statistics. Valid arguments are an integer followed by:

ms - milliseconds (period is truncated to the nearest second)
s - seconds (Default)

min - minutes
h - hours
0 - Disables automatic polling

The default period is 30 seconds.

--director ***context path, context path***... - Context path(s) to one or more directors to display statistics for. Separated by commas.

--targets ***context path, context path***... - Context path(s) to one or more targets to display statistics for, separated by commas. Applicable only to statistics that require a target.

Example: default period, no targets

Create a simple monitor with the default period, and no targets:

```
VPlexcli:/monitoring> monitor create --name TestMonitor --director Director-2-1-B --stats
director.fe-read,director.fe-write
Successfully created 1 monitor(s) out of 1.
```

Example: 10 seconds, directors

Create a monitor to collect statistics from the director category on /engines/engine-1-1/directors/director-1-1-A every 10 seconds:

```
VPlexcli:/monitoring> monitor create --name DirStats --period 10s --director
/engines/engine-1-1/directors/director-1-1-A --stats director.*
```

Example: all statistics for all volumes

Create a monitor to collect statistics on all storage volumes at cluster-1:

```
VPlexcli:/monitoring> monitor create --name SVStats-Cluster1 --director
/engines/engine-1-1/directors/director-1-1-A --stats storage-volume.* --targets
/clusters/cluster-1/storage-elements/storage-volumes/*
```

Example: fe statistics for a specified fe port

Create a monitor to collect all front-end statistics on front-end port FC01:

```
VPlexcli:/monitoring> monitor create --name FE-FC01-stats --director
/engines/engine-1-1/directors/director-1-1-A --stats fe-prt.* --targets
/engines/engine-1-1/directors/director-1-1-A/hardware/ports/A0-FC01
```

Example: udt port statistics for a specified director

Create a performance monitor to collect UDT port statistics for ports on the specified director:

```
VPlexcli:/monitoring> monitor create --name Director-1-1-A-UDT --director
/engines/engine-1-1/directors/director-1-1-A --stats director.udt*
```

Example: port-level WAN statistics

Create a performance monitor to collect port-level WAN-COM statistics:

```
VPlexcli:/monitoring> monitor create --name WAN_A2-XG00 --director
/engines/engine-1-1/directors/director-1-1-A --stats ip-com-port.* --targets
/engines/engine-1-1/directors/director-1-1-A/hardware/ports/A2-XG00
```

Example: Local COM latency:

Create a perfomance monitor to monitor local COM latency for a specified director:

```
VPlexcli:/> monitor create --name local-cluster --stats "com-cluster-io.*" --director
director-1-1-A --targets "/clusters/cluster-1"
```

Example: Remote cluster latency:

Create a perfomance monitor to monitor latency to the remote cluster:

```
VPlexcli:/> monitor create --name remote-cluster --stats "com-cluster-io.*" --director
director-1-1-A --targets "/clusters/cluster-2"
```

Example: Send CAW statistics to the management server

- Create a performance monitor to collect statistics for CompareAndWrite (CAW) operations, miscompares, and latency for the specified virtual volume on director-1-1-B.
- Add a file sink to send output to the specified directory on the management server.

```
VPlexcli:/monitoring/directors/director-1-1-B/monitors> monitor create --name fe_lu_caw
--director /engines/engine-1-1/directors/director-1-1-B --stats fe-lu.caw* --targets
/clusters/cluster-1/virtual-volumes/VAAI_Vol1_Device_vol

VPlexcli:/monitoring/directors/director-1-1-B/monitors> cd director-1-1-B_fe_lu_caw/sinks

VPlexcli:/monitoring/directors/director-1-1-B/monitors/director-1-1-B_fe_lu_caw/sinks>
monitor add-file-sink /var/log/VPlex/cli/director-1-1-B_fe_lu_caw
```

Add/delete monitor sinks

Every monitor must have at least one sink, and may have multiple sinks. There are two types of sink:

console - Sends output to VPLEX management server console.

file - Sends output to the specified file.

Add a console sink

Use the **monitor add-console-sink** command to add a console sink to an existing monitor.

CAUTION

Console monitors display the selected statistics on the VPLEX Management Console, interrupting any other input/output to/from the console. Refer to "Enable/disable sinks" on page 253 for the command to disable a console sink.

The syntax for the command is:

```
add-console-sink
   [-o|--format] {csv|table}
   [-m|--monitor] monitor-name
   --force
```

The default format for console sinks is 'table'.

To add a console sink with output formatted as table (the default output format):

```
VPlexcli:/> monitor add-console-sink --monitor Director-2-1-B_TestMonitor
```

Navigate to the monitor context and use the **ll console** command to display the sink:

```
VPlexcli:/cd monitoring/directors/Director-2-1-B/monitors/Director-2-1-B_TestMonitor/sinks
VPlexcli:/monitoring/directors/Director-2-1-B/monitors/Director-2-1-B_TestMonitor/sinks> ll
```

```
Name     Enabled  Format  Sink-To
-------  -------  ------  -------
console  true     table   console
```

```
VPlexcli:/monitoring/directors/Director-2-1-B/monitors/Director-2-1-B_TestMonitor/sinks> ll
console
```

```
/monitoring/directors/Director-2-1-B/monitors/Director-2-1-B_TestMonitor/sinks/console:
Name     Value
-------  -------
enabled  true
format   table
sink-to  console
type     console
```

Add a file sink

Use the **monitor add-file-sink** command to add a file sink to an existing monitor.

The syntax for the command is:

```
add-file-sink
   [-n|--name] sink-name
   [-o|--format] {csv|table}
   [-m|--monitor] monitor-name
   [-f|--file] file
   --force
```

The default format for file sinks is csv (comma-separated values).

The default name of the new sink is 'file".

The default location for the sink output is /var/log/VPlex/cli.

To add a file sink to send output to the specified .csv file:

```
VPlexcli:/monitoring/directors/director-1-1-A/monitors> monitor add-file-sink  --monitor
director-1-1-A_stats --file /var/log/VPlex/cli/director_1_1_A.csv
```

Navigate to the monitor sinks context and use the **ll** ***sink-name*** command to display the sink:

```
VPlexcli:/cd monitoring/directors/Director-1-1-A/monitors/director-1-1-A_stats/sinks
VPlexcli:/monitoring/directors/Director-1-1-A/monitors/director-1-1-A_stats/sinks> ll file

/monitoring/directors/Director-1-1-A/monitors/director-1-1-A_stats/sinks/file:
Name     Value
-------  ------------------------------
enabled  true
format   csv
sink-to  /var/log/VPlex/cli/director_1_1_A.csv
type     file
```

Delete a monitor sink

Use the **monitor remove sink** command to remove a sink from a monitor:

```
VPlexcli:/monitoring/directors/director-2-1-B/monitors/director-2-1-B_TestMonitor> monitor
remove-sink console
```

Delete a monitor

Use the **monitor destroy** ***monitor*** command to delete a specified monitor.

For example:

```
VPlexcli:/monitoring/directors/director-1-1-B/monitors> monitor destroy
director-1-1-B_TestMonitor

WARNING: The following items will be destroyed:

Context
------------------------------------------------------------------------------
/monitoring/directors/director-1-1-B/monitors/director-1-1-B_TestMonitor

Do you wish to proceed? (Yes/No) y
```

Display monitors

Use the **ls /monitoring/directors/*/monitors** command to display the names of all monitors configured on the system:

```
VPlexcli:/> ls /monitoring/directors/*/monitors

/monitoring/directors/director-1-1-A/monitors:
DEFAULT_director-1-1-A_PERPETUAL_vplex_sys_perf_mon_v8
director-1-1-A_Billy35_FE_A0-FC00_stats
director-1-1-A_director-fe-21112011
director-1-1-A_diskReportMonitor
.
.
.
/monitoring/directors/director-1-1-B/monitors:
DEFAULT_director-1-1-B_PERPETUAL_vplex_sys_perf_mon_v8
.
.
.
```

Use the **ll /monitoring/directors/*director*/monitors** command to display summarized information about all the monitors for the specified context/object:

```
VPlexcli:/> ll /monitoring/directors/director-1-1-A/monitors

/monitoring/directors/director-1-1-A/monitors:
Name                           Ownership  Collecting  Period  Average  Idle  Bucket  Bucket  Bucket  Bucket
                               ---------  Data        ------  Period   For   Min     Max     Width   Count
------------------------------ ---------  ----------  ------  -------  ----  ------  ------  ------  ------
director-1-1-A_FE_A0-FC00      false      false        5s      -        -     -       -       -       64
director-1-1-A_director-fe     false      false        5s      -        -     -       -       -       64
director-1-1-A_ipcom-21112011  false      false        5s      -        -     -       -       -       64
director-1-1-A_portReportMon   false      false        5s      -        -     -       -       -       64
.
.
.
```

Use the **ll /monitoring/directors/*director*/monitors/*monitor-name*** command to display detailed information about all the specified monitor:

```
VPlexcli: ll /monitoring/directors/Director-2-1-B/monitors/Director-2-1-B_volumeReportMonitor

Attributes:
Name             Value
---------------  ------------------------------------------------------------
average-period   -
bucket-count     64
bucket-max       -
bucket-min       -
bucket-width     -
collecting-data  true
firmware-id      9
idle-for         5.44days
ownership        true
period           0s
statistics       [virtual-volume.ops, virtual-volume.read,
                 virtual-volume.write]
```

```
targets          DR1_C1-C2_1gb_dev10_vol, DR1_C1-C2_1gb_dev11_vol,
                 DR1_C1-C2_1gb_dev12_vol, DR1_C1-C2_1gb_dev13_vol,
                 DR1_C1-C2_1gb_dev14_vol, DR1_C1-C2_1gb_dev15_vol,
                 DR1_C1-C2_1gb_dev16_vol, DR1_C1-C2_1gb_dev17_vol,
                 DR1_C1-C2_1gb_dev18_vol, DR1_C1-C2_1gb_dev19_vol, ... (1300
                 total)


Contexts:
Name   Description
-----  ------------------------------------------------------------------------
sinks  Contains all of the sinks set up to collect data from this performance
       monitor.
```

Use the **ll /monitoring/directors/*director*/monitors/*monitor-name*/sinks** command to display the sinks associated with the specified monitor:

```
VPlexcli: ll
/monitoring/directors/Director-2-1-B/monitors/Director-2-1-B_volumeReportMonitor
/sinks

/monitoring/directors/bob70/monitors/bob70_volumeReportMonitor/sinks:
Name  Enabled  Format  Sink-To
----  -------  ------  ------------------------------------------------------
file  true     csv     /var/log/VPlex/cli/reports/volumeReportMonitor_bob70.csv
```

Table 13 Monitor and sink field descriptions (1 of 2)

Field	Description
average-period	The actual average sampling period.
bucket-count	The number of buckets used for gathering statistics.
bucket-max	The upper bound for bucketed statistics being monitored.
bucket-min	The lower bound for bucketed statistics being monitored.
bucket-width	The width of the range of values that a given bucket represents. Use this to adjust the upper bound of monitoring.
collecting-data	Whether or not this performance monitor is collecting data. A monitor collects data if it has at least one enabled sink.
firmware-id	The firmware ID of the monitor.
idle-for	The elapsed time since this performance monitor was accessed in the firmware.
name	A director-wide unique name for this performance monitor intended to be meaningful to the user.
ownership	Whether or not this monitor was created in this instance of VPlex Management Console.

Table 13 **Monitor and sink field descriptions (2 of 2)**

Field	Description
period	Sampling period in seconds.
statistics	List of performance statistics that are being monitored.
targets	List of targets that apply to the monitored performance statistics. A target can be a port, storage-volume, or virtual volume. **Note:** Not all statistics require targets.
Monitor sink display fields	
Name	For file sinks, the name of the created sink context. Default is 'file'.
Enabled	Whether the monitor sink is enabled or disabled.
Format	The required output format. Can be 'csv' or 'table'. Default is 'csv' for file sinks and "table" for console sinks.
Sink-To	For file sinks, the filename to sink data to.

Enable/disable/change polling

Polling (collection of the specified statistics) begins when the first sink is added to a monitor. Polling occurs automatically at the interval specified by the monitor's **period** attribute.

Use the **set** command to change the polling period.

Use the **monitor collect** command to run a collection immediately, before its defined polling interval.

Use the **set** command to disable, or modify automatic polling for a monitor.

In the following example:

- The **set** command changes the **period** attribute to 0, disabling automatic polling
- The **ll** command displays the change:

```
VPlexcli:/monitoring/directors/director-2-1-B/monitors/director-2-1-B_TestMonitor> set period
0

VPlexcli:/monitoring/directors/director-2-1-B/monitors/director-2-1-B_TestMonitor> ll

Attributes:
Name             Value
---------------  ------------------------------------------------------------
average-period   -
bucket-count     64
bucket-max       -
bucket-min       -
bucket-width     -
collecting-data  false
firmware-id      4
idle-for         5.78min
ownership        true
period           0s
.
.
.
```

To re-enable polling, use the **set** command to change the **period** attribute to a non-zero value.

Enable/disable sinks

Use the **set** command to enable/disable a monitor sink.

To disable a monitor sink:

```
VPlexcli:/monitoring/directors/director-2-1-B/monitors/director-2-
1-B_TestMonitor/sinks/console> set enabled false

VPlexcli:/monitoring/directors/director-2-1-B/monitors/director-2-
1-B_TestMonitor/sinks/console> ll
Name     Value
-------  -------
enabled  false
format   table
sink-to  console
type     console
```

To enable a monitor sink:

```
VPlexcli:/monitoring/directors/director-2-1-B/monitors/director-2-
1-B_TestMonitor/sinks/console> set enabled true

VPlexcli:/monitoring/directors/director-2-1-B/monitors/director-2-
1-B_TestMonitor/sinks/console> ll
Name     Value
-------  -------
enabled  true
format   table
sink-to  console
type     console
```

Force an immediate poll

Use the **monitor collect** ***monitor*** command to force an immediate poll and collection of performance data without waiting for the automatic poll interval.

For example:

```
VPlexcli:/> monitor collect
/monitoring/directors/director-2-1-B/monitors/director-2-1-B_TestMonitor

VPlexcli:/>
Source:                            director-2-1-B_TestMonitor
Time:                                     2010-07-01 10:05:55
director.be-ops (counts/s):
.
.
.
```

Pre-configured performance monitors

Pre-configured monitors are an easy way to create monitors to collect information used to diagnose common problems.

The **report create-monitors** command creates three monitors for each director in the system. Monitors are named:

- Cluster_*n*_Dir_*nn*_diskReportMonitor
- Cluster_*n*_Dir_*nn*_portReportMonitor
- Cluster_*n*_Dir_*nn*_volumeReportMonitor

The **period** attribute for the new monitors is set to 0 (automatic polling is disabled).

The **report poll-monitors** command is used to force a poll.

Each monitor has one file sink. The file sinks are enabled.

By default, output files are located in:

/var/log/VPlex/cli/reports/

on the management server. Output filenames are in the following format:

<Monitor name>_Dir_*nn*.csv

- Pre-configured disk monitors collect the following statistics:
  - storage-volume.per-storage-volume-read-latency
  - storage-volume.per-storage-volume-write-latency
- Pre-configured port monitors collect the following statistics:
  - be-prt.read
  - be-prt.write
  - fe-prt.ops
  - fe-prt.read
  - fe-prt.write
- Pre-configured volume monitors collect the following statistics:
  - virtual-volume.ops
  - virtual-volume.read
  - virtual-volume.write

Create pre-configured monitors

In the following example:

- The **report create-monitors** command creates a diskReportMonitor, portReportMonitor, and volumeReportMonitor for each director,
- The **ll //monitoring/directors/*/monitors** command displays the new monitors:

```
VPlexcli:/> report create-monitors
Creating monitor diskReportMonitor on Director Cluster_1_Dir1A monitoring 981 targets, file
/var/log/VPlex/cli/reports/diskReportMonitor_Cluster_1_Dir1A.csv.
Successfully created 1 monitor(s) out of 1.

time: 11 sec
Creating monitor volumeReportMonitor on Director Cluster_1_Dir1A monitoring 30 targets, file
/var/log/VPlex/cli/reports/volumeReportMonitor_Cluster_1_Dir1A.csv.
Successfully created 1 monitor(s) out of 1.

time: 0 sec
Creating monitor portReportMonitor on Director Cluster_1_Dir1A monitoring 16 targets, file
/var/log/VPlex/cli/reports/portReportMonitor_Cluster_1_Dir1A.csv.
Successfully created 1 monitor(s) out of 1.Creating monitor diskReportMonitor on Director
Cluster_1_Dir1A monitoring 981 targets, file
/var/log/VPlex/cli/reports/diskReportMonitor_Cluster_1_Dir1A.csv.
Successfully created 1 monitor(s) out of 1.
.
.
.
VPlexcli:/> ll //monitoring/directors/*/monitors

/monitoring/directors/Cluster_1_Dir1A/monitors:
Name                                 Ownership  Collecting  Period  Average  Idle For  Bucket  Bucket   Bucket  Bucket
                                     ---------  Data        ------  Period   --------  Min     Max      Width   Count
-----------------------------------  ---------  ----------  ------  -------  --------  ------  -------
Cluster_1_Dir1A_diskReportMonitor    true       true        0s      -        7.1min    100     1600100  25000   64
Cluster_1_Dir1A_portReportMonitor    true       true        0s      -        6.88min   -       -        -       64
Cluster_1_Dir1A_volumeReportMonitor  true       true        0s      -        6.9min    -       -        -       64


/monitoring/directors/Cluster_1_Dir1B/monitors:
Name                                 Ownership  Collecting  Period  Average  Idle      Bucket  Bucket   Bucket  Bucket
                                     ---------  Data        ------  Period   For       Min     Max      Width   Count
-----------------------------------  ---------  ----------  ------  -------  --------  ------  -------
Cluster_1_Dir1B_diskReportMonitor    true       true        0s      -        6.88min   100     1600100  25000   64
Cluster_1_Dir1B_portReportMonitor    true       true        0s      -        6.68min   -       -        -       64
Cluster_1_Dir1B_volumeReportMonitor  true       true        0s      -        6.7min    -       -        -       64
.
.
.
```

An empty .csv file is created on the management server for each of the monitors:

```
service@ManagementServer:/var/log/VPlex/cli/reports> ll
total 36
.
.
.
-rw-r--r-- 1 service users    0 2010-08-19 13:55 diskReportMonitor_Cluster_1_Dir1A.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:55 diskReportMonitor_Cluster_1_Dir1B.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:56 diskReportMonitor_Cluster_2_Dir_1A.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:55 diskReportMonitor_Cluster_2_Dir_1B.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:55 diskReportMonitor_Cluster_2_Dir_2A.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:56 diskReportMonitor_Cluster_2_Dir_2B.csv
-rw-r--r-- 1 service users    5 2010-08-13 15:04 portPerformance_cluster-1.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:55 portReportMonitor_Cluster_1_Dir1A.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:55 portReportMonitor_Cluster_1_Dir1B.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:56 portReportMonitor_Cluster_2_Dir_1A.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:55 portReportMonitor_Cluster_2_Dir_1B.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:55 portReportMonitor_Cluster_2_Dir_2A.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:56 portReportMonitor_Cluster_2_Dir_2B.csv
-rw-r--r-- 1 service users    5 2010-08-13 15:04 volumePerformance_cluster-1.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:55 volumeReportMonitor_Cluster_1_Dir1A.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:55 volumeReportMonitor_Cluster_1_Dir1B.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:56 volumeReportMonitor_Cluster_2_Dir_1A.csv
```

```
-rw-r--r-- 1 service users    0 2010-08-19 13:55 volumeReportMonitor_Cluster_2_Dir_1B.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:55 volumeReportMonitor_Cluster_2_Dir_2A.csv
-rw-r--r-- 1 service users    0 2010-08-19 13:56 volumeReportMonitor_Cluster_2_Dir_2B.csv
```

Operate the pre-configured monitors

The monitors created by the **report create-monitors** command have their **period** attribute set to 0 seconds (automatic polling is disabled).

Use the **report poll-monitors** command to force an immediate poll and collection of performance data for monitors created by the **report create-monitors** command.

```
VPlexcli:/> report poll-monitors
Collecting data for director Cluster_2_Dir_1B monitor Cluster_2_Dir_1B_diskReportMonitor.
Collecting data for director Cluster_2_Dir_1B monitor Cluster_2_Dir_1B_portReportMonitor.
Collecting data for director Cluster_2_Dir_1B monitor Cluster_2_Dir_1B_volumeReportMonitor.
.
.
.
```

Output is written to the .csv files located in /var/log/VPlex/cli/reports/ on the management server.

Monitoring consistency groups using the CLI

Two groups of statistics are associated with consistency groups:

- "Write pacing (wrt-pacing) statistics" monitor write pacing (throttling) of I/O
- "Consistency-group (cg) statistics" monitor the performance of asynchronous consistency-groups.

These statistics apply only to consistency groups with cache mode set to asynchronous.

Because these statistics are per-consistency-group, a valid consistency-group must be specified as a target when creating a monitor of this type.

If the target consistency-group is synchronous (either when the monitor is created or if its cache mode is subsequently changed to synchronous), all statistics in this group will read "no data".

If the cache-mode is changed (back) to asynchronous, the monitor will behave as expected and the displayed values will reflect the consistency-group's performance.

Monitoring with SNMP

Simple Network Management Protocol (SNMP) is a protocol for managing and monitoring network-attached devices.

The VPLEX SNMP agent:

- Supports retrieval of performance-related statistics as published in the VPLEX-MIB.mib.
- Runs on the management server and fetches performance related data from individual directors using a firmware specific interface.
- Provides SNMP MIB data for directors for the local cluster only.
- Runs on Port 161 of the management server and uses the UDP protocol.
- Supports the following SNMP commands:
  - SNMP Get

- SNMP Get Next
- SNMP get Bulk

VPLEX MIBs are located on the management server in the /opt/emc/VPlex/mibs directory:

```
service@ManagementServer:/opt/emc/VPlex/mibs> ls
VPlex.mib  VPLEX-MIB.mib
```

VPLEX-MIB.mib contains the VPLEX MIB tree:

```
vplex   MODULE-IDENTITY
          LAST-UPDATED "201008250601Z"    -- Aug 25, 2010 6:01:00 AM
          ORGANIZATION "EMC Corporation"
          CONTACT-INFO
                "EMC Corporation
                176 South Street
                Hopkinton, MA 01748 USA
                Phone: 1-800-424-EMC2
                Web  : www.emc.com
                email: support@emc.com"
          DESCRIPTION
                "EMC VPLEX MIB Tree."
          REVISION "201008250601Z"      -- Aug 25, 2010 6:01:00 AM
          DESCRIPTION
                "Initial version."
          -- 1.3.6.1.4.1.1139.21
          ::= { enterprises 1139 21 }
```

VPlex.mib contains 4 VPLEX notifications:

```
vplex MODULE-IDENTITY
    LAST-UPDATED "201002020925Z"
    ORGANIZATION "EMC Corporation"
    CONTACT-INFO
         "postal:   EMC Corporation
                    176 South Street
                    Hopkinton, MA 01748 USA
          phone:    1-800-424-EMC2
          web:      www.emc.com
          email:    support@emc.com"
    DESCRIPTION  "The EMC V-Plex MIB tree."
    REVISION     "201002020925z"
    DESCRIPTION  "Second draft"
          -- 1.3.6.1.4.1.1139.21
          ::= { enterprises 1139 21 }
```

In order to utilize SNMP, the SNMP agent must be configured and started on the VPLEX cluster.

See "Performance statistics retrievable by SNMP" on page 272 for the VPLEX statistics that can be monitored using SNMP.

Call-home events and SNMP trap

When a call-home event is generated and sent from VPLEX, a corresponding SNMP trap is also sent from all configured SNMP trap services that have been started on the VPLEX cluster.

The SNMP trap service can be created and started using the **snmp-traps create** and **set started true** commands respectively in the /call-home/snmp-traps context in the VPLEX CLI. See "Configure SNMP trap in call-home" on page 258 for details.

Configure SNMP

Use the **snmp-agent configure** command to configure and start the SNMP agent:

```
VPlexcli:/> snmp-agent configure
The community string is already configured to be: private.
Choosing to continue will change the existing community string.

Do you want to continue? (yes/no)yes

What community string should the agent use? [private]: public
```

Use the **snmp-agent status** command to verify that the SNMP agent is running:

```
VPlexcli:/> snmp-agent status
SNMP Agent Service status is: Running
```

Configure SNMP trap in call-home

When call-home events are sent, the corresponding SNMP trap can also be sent. The following CLI example shows how to configure the SNMP trap in **call-home.** Multiple SNMP traps can be configured to send SNMP traps to different remote hosts.

```
VPlexcli:/> cd notifications/call-home/snmp-traps/
VPlexcli:/notifications/call-home/snmp-traps> create myTrap
VPlexcli:/notifications/call-home/snmp-traps> cd myTrap/
VPlexcli:/notifications/call-home/snmp-traps/myTrap> ls
Name Value
---------------- ------
community-string public
remote-host -
remote-port 162
started false


VPlexcli:/notifications/call-home/snmp-traps/myTrap> set
community-string myVplex
VPlexcli:/notifications/call-home/snmp-traps/myTrap> set
remote-host 1.1.1.1
VPlexcli:/notifications/call-home/snmp-traps/myTrap> set started
true
VPlexcli:/notifications/call-home/snmp-traps/myTrap> ls
Name Value
---------------- -------
community-string myVplex
remote-host 1.1.1.1
remote-port 162
started true
```

Stop/start SNMP

Use the **snmp-agent stop** command to stop the SNMP agent without removing it from VPLEX:

```
VPlexcli:/> snmp-agent stop
SNMP agent has been stopped.
```

Use the **snmp-agent start** command to restart the SNMP agent:

```
VPlexcli:/> snmp-agent start
SNMP agent has been started.
```

Unconfigure SNMP

Use the **snmp-agent unconfigure** command to stop the SNMP agent and unconfigure it:

```
VPlexcli:/> snmp-agent unconfigure
SNMP agent has been unconfigured.
```

Statistics

VPLEX collects and reports three types of statistics:

- **counters** - Monotonically increasing value (analogous to a car's odometer)

  Counters are used to count bytes, operations, and errors.

  Often reported as a rate such as counts/second or KB/second.

- **readings** - Instantaneous value (analagous to a car's speedometer)

  Readings are used to display CPU utilization, memory utilization.

  Value can change every sample.

- **period-average** - Average of a series calculated over the last sample period. If:

  current_reading_sum is the sum of all readings for the particular statistic since the monitor's creation, and

  previous_reading_sum is the count of all readings for the statistic since the monitor's creation

  period-average =

  (current_reading_sum - previous_reading_sum)/

  (current_reading_count - previous_reading_count)

- **buckets** - Histogram of 'bucketized' counts.

  Buckets are used to track latencies, determine median, mode, percentiles, minimums and maximums.

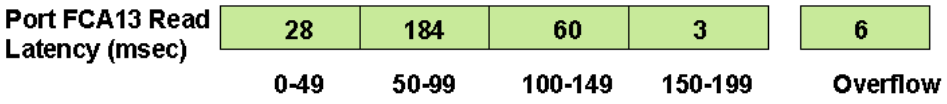


Figure 36 Bucket Statistic

Many statistics require a target port or volume to be specified. Output of the **monitor stat-list** command identifies which statistics need a target defined, and the type of target required when a monitor is created.

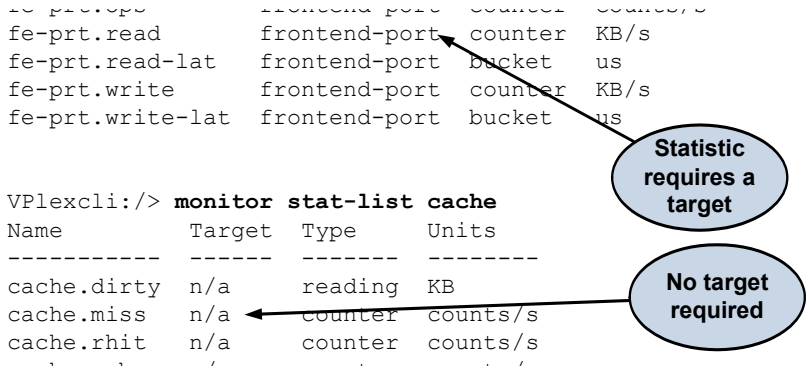


Figure 37 Monitoring targets

Display available statistics

Statistics are grouped into sub-categories.

Use the **monitor stat-list** command followed by the <Tab> key to display the statistics sub-categories. For example:

```
VPlexcli:/> monitor stat-list

be-prt, cache, cg, director, directory, fc-com-port, fe-director,
   fe-lu, fe-prt, ip-com-port, ramf, rdma, storage-volume,
   virtual-volume, wrt-pacing
```

Use the **--categories** *categories* option to display the statistics available in the specified category. For example:

```
VPlexcli:/monitoring> monitor stat-list --categories director
Name                    Target  Type     Units
----------------------  ------  -------  --------
director.be-aborts      n/a     counter  counts/s
director.be-ops         n/a     counter  counts/s
director.be-ops-read    n/a     counter  counts/s
director.be-ops-write   n/a     counter  counts/s
director.be-read        n/a     counter  KB/s
.
.
.
```

Use the * wildcard to display all statistics for all categories.

For example:

```
VPlexcli:/> monitor stat-list *
Name                                            Target             Type     Units
----------------------------------------------- -----------------  -------  --------
be-prt.read                                     backend-port       counter  KB/s
be-prt.write                                    backend-port       counter  KB/s
cache.dirty                                     n/a                reading  KB
cache.miss                                      n/a                counter  counts/s
cache.rhit                                      n/a                counter  counts/s
cache.subpg                                     n/a                counter  counts/s
cg.closure                                      consistency-group  bucket   us
cg.delta-util                                   consistency-group  reading  %
cg.drain-lat                                    consistency-group  bucket   us
cg.exch-bytes                                   consistency-group  counter  KB/s
cg.exch-lat                                     consistency-group  bucket   us
cg.exch-pages                                   consistency-group  counter  counts/s
cg.input-bytes                                  consistency-group  counter  KB/s
cg.input-ops                                    consistency-group  counter  counts/s
cg.inter-closure                                consistency-group  bucket   us
cg.outOfDate-counter                            consistency-group  counter  counts/s
cg.pipe-util                                    consistency-group  reading  %
cg.write-bytes                                  consistency-group  counter  KB/s
cg.write-lat                                    consistency-group  bucket   us
cg.write-pages                                  consistency-group  counter  counts/s
.
.
.
```

Statistics tables

The following tables list the statistics in each category:

Table 14 Back-end fibre channel port (be-prt) statistics

Statistic	Type	Description
be-prt.read type: counter, units: bytes/second, arguments: port#	Back-end port reads	Number of bytes read through the specified FC port.
be-prt.write type: counter, units: bytes/second, arguments: port#	Back-end port writes	Number of bytes written through the specified FC port.

Table 15 Cache statistics

Statistic	Type	Description
cache.dirty type: reading, units: bytes, arguments: none	Cache dirty	Number of modified pages in cache. Always 0 for writethrough.
cache.miss type: counter, units: counts/second, arguments: none	Cache miss	Operations to a page that was not already in cache.
cache.rhit type: counter, units: counts/second, arguments: none	Cache remote hits	Number of hits on another director. If this number is high, it indicates inefficiency at the driver level. Optimize your multipath driver.
cache.subpg type: counter, units: counts/second, arguments: none	Cache subpage	Number of operations smaller than the page size or misaligned. A large number indicates a performance problem. Align the file system.

Table 16 Director statistics (1 of 2)

Statistic	Type	Description
director.async-write type: counter, units: bytes/second, arguments: none	Asynchronous write	Total I/O to all asynchronous consistency groups at a director
director.be-aborts type: counter, units: counts/second, arguments: none	Back-end operations	Number of aborted I/O operations through the director's back-end ports.
director.be-ops type: counter, units: counts/second, arguments: none	Back-end operations	Number of I/O operations through the director's back-end ports.
director.be-ops-read type: counter, units: counts/second, arguments: none	Back-end reads	Number of reads by the director's back-end ports.
director.be-ops-write type: counter, units: counts/second, arguments: none	Back-end writes	Number of writes through the director's back-end ports.
director.be-read type: counter, units: bytes/second, arguments: none	Back-end reads	Number of bytes read by the director's back-end ports.
director.be-write type: counter, units: bytes/second, arguments: none	Back-end writes	Number of bytes written by the director's back-end ports.
director.busy type: reading; units: percentage, arguments: none	CPU	Percentage of CPU usage.
director.dr1-rbld-recv type: counter, units: bytes/second, arguments: none	Rebuild bytes received	Number of bytes received by this node from remote node(s) for rebuild traffic (reads and/or writes).
director.dr1-rbld-sent type: counter, units: bytes/seconds, arguments: none	Rebuild bytes sent	Number of bytes sent by this node to remote node(s) for rebuild traffic (reads and/or writes)
director.fe-ops type: counter, units: counts/second, arguments: none	Front-end operations	Number of I/O operations through the director's front-end ports.
director.fe-ops-act type: reading, units: counts, arguments: none	Front-end operations active	Number of active outstanding I/O operations on the director's front-end ports.

Table 16 **Director statistics (2 of 2)**

Statistic	Type	Description
director.fe-ops-q type: reading, units: counts, arguments: none	Front-end operations queued	Number of queued outstanding I/O operations on the director's front-end ports.
director.fe-ops-read type: counter, units: counts/second, arguments: none	Front-end reads	Number of reads on the director's front-end ports.
director.fe-ops-write type: counter, units: counts/second arguments: none	Front-end writes	Number of writes on the director's front-end ports.
director.fe-read type: counter, units: bytes/second, arguments: none	Front-end reads	Number of bytes read from the director's front-end ports.
director.fe-write type: counter, units: bytes/second, arguments: none	Front-end writes	Number of bytes written to the director's front-end ports.
director.heap-used type: reading; units: percentage, arguments: none	Memory	Percentage of memory usage on the director.
director.per-cpu-busy type: reading, units: percentage, arguments: none	CPU busy	The total utilization (user and system) of each CPU in the director.
director.tcp-recv type: counter, units: bytes/second, arguments: none	TCP bytes received	Number of bytes received on the specified director's TCP ports.
director.tcp-send type: counter, units: bytes/second, arguments: none	TCP bytes sent	Number of bytes sent on the specified director's TCP ports.
director.udt-conn-drop type: counter, units: counts/second, arguments: none	UDT connections dropped	Number of UDT connections dropped on the director's UDT ports.
director.udt-pckt-retrans type: counter, units: counts/second, arguments: none	Packets resent	Number of packets resent on the director's UDT ports.
director.udt-recv-bytes type: counter, units: bytes/second, arguments: none	UDT bytes received	Number of packets received on the director's UDT ports.
director.udt-recv-drops type: counter, units: counts/second, arguments: none	UDT received packets dropped	Number of received packets dropped on the director's UDT ports.
director.udt-recv-pckts type: counter, units: counts/second, arguments: none	UTDT received packets	Number of packets received on the director's UDT ports.
director.udt-send-bytes type: counter, units: bytes/second, arguments: none	UDT sent bytes	Number of bytes sent on the director's UDT ports.
director.udt-send-drops type: counter, units: count/second, arguments: none	UDT sent packets dropped	Number of sent packets dropped on the director's UDT ports.
director.udt-send-pckts type: counter, units: count/second, arguments: none	USDT sent packets	Number of packets sent t on the director's UDT ports.

Table 17 Directory statistics

Statistic	Type	Description
directory.ch-remote	Cache coherence	Internal EMC use only
directory.chk-total	Cache coherence	Internal EMC use only
directory.dir-total	Cache coherence	Internal EMC use only
directory.dr-remote	Cache coherence	Internal EMC use only
directory.ops-local	Cache coherence	Internal EMC use only
directory.ops-rem	Cache coherence	Internal EMC use only

Table 18 Front-end director (fe-director) statistics

Statistic	Type	Description
fe-director.aborts type: counter, units: counts/second, arguments: none	Front-end operations	Number of aborted I/O operations through the director's front-end ports.
fe-director.caw-lat type: bucket, units: microsecond, arguments: none	CompareAndWrite operations latency	CompareAndWrite latency in microseconds on the specified director's front-end ports. The latency bucket is reduced to three buckets from 0 to maximum instead of 64 latency buckets collected within the VPLEX firmware.
fe-director.read-lat type: bucket, units: microsecond, arguments: none	Front-end director read latency	Read latency distribution in microseconds on the director's front-end ports.
fe-director.write-lat type: bucket, units: microsecond, arguments: none	Front-end director write latency	Write latency distribution in microseconds on the director's front-end ports.

Table 19 Front-end volume (fe-lu) statistics (1 of 2)

Statistic	Type	Description
fe-lu.caw-lat type: bucket, units: microsecond, arguments:volume-id	CompareAndWrite operations latency	CompareAndWrite latency in microseconds on the specified front-end volume.
fe-lu.caw-mis type: counter, units: counts/second, arguments: volume-id	CompareAndWrite miscompares	Number of CompareAndWrite miscompares on the specified front-end volume.
fe-lu.caw-ops type: counter, units: counts/second, arguments: volume-id	CompareAndWrite operations	Number of CompareAndWrite operations on the specified front-end volume.
fe-lu.ops type: counter, units: counts/second, arguments: volume-id	Front-end volume operations	Number of I/O operations on the specified front-end volume.

Table 19 Front-end volume (fe-lu) statistics (2 of 2)

Statistic	Type	Description
fe-lu.read type: counter, units: bytes/second, arguments: volume-id	Front-end volume reads	Number of reads on the specified front-end volume.
fe-lu.read-lat type: bucket, units: microsecond, arguments: volume-id	Front-end volume read latency	Read latency distribution in microseconds on the specified front-end volume.
fe-lu.write type: counter, units: bytes/second, arguments: volume-id	Front-end volume writes	Number of writes on the specified front-end volume.
fe-lu.write-lat type: bucket, units: microsecond, arguments: volume-id	Front-end volume write latency	Write latency distribution in microseconds on the specified front-end volume.

Table 20 Front-end port (fe-prt) statistics

Statistic	Type	Description
fe-prt.caw-lat type: bucket, units: microsecond, arguments:port#	CompareAndWrite operations latency	CompareAndWrite latency in microseconds on the specified front-end port.
fe-prt.caw-mis type: counter, units: counts/sec, arguments: port#	CompareAndWrite miscompares	Number of CompareAndWrite miscompares on the specified front-end port.
fe-prt.caw-ops type: counter, units: counts/sec, arguments: port#	CompareAndWrite operations	Number of CompareAndWrite operations on the specified front-end port.
fe-prt.ops type: counter, units: counts/sec, arguments: port#	Front-end port operations	Number of I/O operations on the specified front-end FC port.
fe-prt.read type: counter, units: bytes/sec, arguments: port#	Front-end port reads	Number of bytes read from the specified front-end FC port.
fe-prt.read-lat type: bucket, units: microsecond, arguments: port#	Front-end port read latency	Read latency distribution in microseconds on the specified front-end FC port.
fe-prt.write type: counter, units: bytes/second, arguments: port#	Front-end port writes	Number of bytes written to the specified front-end FC port.
fe-prt.write-lat type: bucket, units: microsecond, arguments: port#	Front-end port write latency	Write latency distribution in microseconds on the specified front-end FC port.

Table 21 Remote RAID (ramf) statistics

Statistic	Type	Description
ramf.cur-op type: reading, units: counts, arguments: none	Current op count	Instantaneous count of remote RAID operations.
ramf.exp-op type: counter, units: counts/second, arguments: none	Remote operations	Total number of remote IOPS.
ramf.exp-rd type: counter, units: bytes/second, arguments: none	Remote reads	Remote reads from another cluster to a disk or LUN at the local cluster.
ramf.exp-wr type: counter, units: bytes/second, arguments: none	Remote writes	Remote writes from another cluster to a disk or LUN at the local cluster.
ramf.imp-op type: counter, units: counts/second, arguments: none	Imported ops	Number of operations that have been requested by a given director, regardless of remote target.
ramf.imp-rd type: counter, units: bytes/second, arguments: none	Imported reads	Reads from the local cluster to a disk or LUN at a remote cluster.
ramf.imp-wr type: counter, units: bytes/second, arguments: none	Imported writes	Writes from the local cluster to a disk or LUN at a remote cluster.

Table 22 Remote data memory access (rdma) statistics

Statistic	Type	Description
rdma.cur-ops type: reading, units: counts, arguments: none	RDMA ops	Instantaneous count of RDMA ops outstanding.
rdma.read type: counter, units: bytes/second, arguments: none	RDMA reads	Number of RDMA read operations.
rdma.write type: counter, units: bytes/second, arguments: none	RDMA writes	Number of RDMA write operations.

Table 23 Storage-volume statistics

Statistic	Type	Description
storage-volume.per-storage-volume-read-latency type: bucket, units: microsecond, arguments: volume-id	Volume read latency	Read latency distribution in microseconds on the specified storage volume.
storage-volume.per-storage-volume-write-latency type: bucket, units: microsecond, arguments: volume-id	Volume write latency	Write latency distribution in microseconds on the specified storage volume.
storage-volume.read-latency type: bucket, units: microsecond, arguments: none	Average volume read latency	Average read latency distribution in microseconds on all storage volumes.
storage-volume.write-latency type: bucket, units: microsecond, arguments: none	Average volume write latency	Average write latency distribution in microseconds on all storage volumes.

Table 24 Virtual-volume statistics

Statistic	Type	Description
virtual-volume.dirty type: reading, units: counts, arguments: volume-id	Volume dirty	Number of modified pages in cache for the specified virtual volume.
virtual-volume.ops type: counter, units: counts/second, arguments: volume-id	Volume operations	Total number of I/O operations for the specified virtual volume.
virtual-volume.read type: counter, units: bytes/second, arguments: volume-id	Volume reads	Number of reads in bytes for the specified virtual volume.
virtual-volume.write type: counter, units: bytes/second, arguments: volume-id	Volume writes	Number of writes in bytes for the specified virtual volume.

Table 25 Write pacing (wrt-pacing) statistics

Statistic	Description
wrt-pacing.avg-delay type: reading, units: millisecond, arguments: consistency-group	The average delay incurred on the throttled I/Os since the last iostat report.
wrt-pacing.avg-pdrain type: reading, units: bytes/second, arguments: consistency-group	Average drain rate of the consistency group pipe since the last iostat report.
wrt-pacing.avg-pinput type: reading, units: bytes/second, arguments: consistency-group	Average input rate of the consistency group pipe since the last iostat report.
wrt-pacing.is-throttling type: reading, units: counts, arguments: consistency-group	Has any I/O for this consistency group been throttled (delayed) since the last iostat report? **1-** True. I/O has been throttled at least once since the last iostat report. **0** - False. I/O has been never throttled.
wrt-pacing.peak-putil type: reading, units: percent, arguments: consistency-group	Maximum pipe utilization observed since the last iostat report.

Table 26 Consistency-group (cg) statistics

Statistic	Description
cg.closure type: bucket, units: microsecond, arguments: consistency-group	A distribution of time taken to complete closure requests over the last sampling period.
cg.delta-util type: reading, units: percentage, arguments: consistency-group	The average capacity utilization of closed deltas (how full the deltas were when a closure was requested) over the last sampling period.
cg.drain-lat type: bucket, units: microsecond, arguments: consistency-group	A distribution of time taken to drain deltas out of a asynchronous consistency-group pipe over the last sampling period. The time taken to drain a delta is the sum of the exchange, write and commit latencies.
cg.exch-bytes type: counter, units: bytes/second, arguments: consistency-group	The number of bytes received from other cluster(s) at this director over the last sampling period.
cg.exch-lat type: bucket, units: microsecond, arguments: consistency-group	A distribution of time spent in the exchange phase when processing deltas over the last sampling period.
cg.exch-pages type: counter, units: counts/second, arguments: consistency-group	The number of pages received from other cluster(s) at this director over the last sampling period.
cg.input-bytes type: counter, units: bytes/second, arguments: consistency-group	The number of incoming bytes to the asynchronous consistency-group pipe over the last sampling period.
cg.iput-ops type: counter, units: counts/second, arguments: consistency-group	The number of inbound-data operations to an asynchronous consistency-group pipe over the last sampling period.
cg.inter-closure type: bucket, units: microsecond, arguments: consistency-group	A distribution of time between consecutive closure requests over the last sampling period.
cg.outOfDate-counter type: counter, units: counts/second, arguments: consistency-group	The number of writes over the last sampling period for which an underlying DR1 leg was out of date.
cg.pipe-util type: reading, units: percentage, arguments: consistency-group	The average utilization (number of outstanding deltas) of the asynchronous consistency-group pipe over the last sampling period.
cg.write-bytes type: counter, units: bytes/second, arguments: consistency-group	The number of bytes written to the back end at this director over the last sampling period.
cg.write-lat type: bucket, units:microsecond, arguments: consistency-group	A distribution of duration spent in the write phase when processing deltas over the last sampling period.
cg.write-pages type: counter, units: counts/second, arguments: consistency-group	The number of pages written to the back end at this director over the last sampling period.

Table 27 RecoverPoint volume statistics (rp-spl-vol)

Statistic	Description
rp-spl-vol.write-active type: reading, units: counts, arguments: vol-id	The current number of outstanding writes in the splitter for a specific volume.
rp-spl-vol.-write-ops type: counter, units: counts/second, arguments: vol-id	Number of writes that have been processed by the splitter for a specific volume.
rp-spl-vol.-write type: counter, units: bytes/second, arguments: vol-id	The quantity of write data that has been split for a specific volume.
rp-spl-vol.write-lat type: bucket, units: microsecond, arguments: vol-id	Latency from when a write is sent to the splitter to when it is sent down the normal BE I/O stack. Measures SCSI round-trip time to get the write data into the memory of the RPA.

Table 28 RecoverPoint splitter statistics (rp-spl-node)

Statistic	Description
rp-spl-node.write-active type: reading, units: counts, arguments: none	Current number of outstanding writes in the splitter. Tracks split I/Os from when they enter the splitter until they finish in the BE per director..
rp-spl-node.write-ops type: counter, units: counts/second, arguments: none	Number of writes that have been processed by the splitter.
rp-spl-node.write type: counter, units: bytes/second, arguments: none	Quantity of write data that has been split.
rp-spl-node.write-lat type: bucket, units: microsecond, arguments: none	Latency from when a write is sent to the splitter to when it is sent down the normal BE I/O stack. Measures SCSI round-trip time to get the write data into the memory of the RPA..

Table 29 IP WAN COM (ip-com-port) statistics

Statistic	Type	Description
ip-com-port.avg-latency type: reading, units: microsecond, arguments: port-name	IP WAN COM port average latency	Total latency/number of Keepaliveack round trips, including wire and COM latency in microseconds on this IP WAN COM port. **Note:** Not useful for describing wire latency. These latency values reflect the quality of service over a WAN COM port. Values include: wire latency, COM latency increases, and retransmitted packets due to packet drops on the wire. Retransmitted packets increase the maximum and average latency values. These values normalize once the packet drops are no longer occurring.
ip-com-port.max-latency type: reading, units: microsecond, arguments: port-name	IP WAN COM port maximum latency	Keepaliveack round trip, including wire and COM latency in microseconds on this IP WAN COM port.
ip-com-port.min-latency type: reading, units: microsecond, arguments: port-name	IP WAN COM port minimum latency	Keepaliveack round trip, including wire and COM latency in microseconds on this IP WAN COM port.
ip-com-port.pckt-retrans type: counter, units: counts/second, arguments: port-name	IP WAN COM Port packets resent	Number of packets resent on this IP WAN COM port.
ip-com-port.recv-bytes type: counter, units: bytes/second, arguments: port-name	IP WAN COM Port bytes received	Number of bytes received through UDT on this IP WAN COM port.
ip-com-port.recv-drops type: counter, units: counts/second, arguments: port-name	IP WAN COM Port received packets dropped	Number of received packets dropped on this IP WAN COM port.
ip-com-port.recv-pckts type: counter, units: counts/second, arguments: port-name	IP WAN COM Port packets received	Number of packets received through UDT on this IP WAN COM port.
ip-com-port.send-bytes type: counter, units: bytes/second, arguments: port-name	IP WAN COM Port bytes sent	Number of bytes sent through UDT on this IP WAN COM port.
ip-com-port.send-drops type: counter, units: counts/second, arguments: port-name	IP WAN COM Port sent packets dropped	Number of sent packets dropped on this IP WAN COM port.
ip-com-port.send-pckts type: counter, units: counts/second, arguments: port-name	IP WAN COM Port packets sent	Number of packets sent through UDT on this IP WAN COM port.

Table 30 Fibre Channel WAN COM (fc-com-port) statistics

Statistic	Type	Description
fc-com-port.recv-bytes type: counter, units: bytes/second, arguments: fibrechannel-port	FC-COM bytes received	Number of bytes received on this FibreChannel port.
fc-com-port.recv-pckts type: counter, units: counts/second, arguments: fibrechannel-port	FC-COM packets received	Number of packets received on this FibreChannel port.
fc-com-port.send-bytes type: counter, units: bytes/second, arguments: fibrechannel-port	FC-COM bytes send	Number of bytes sent on this FibreChannel port.
fc-com-port.send-pckts type: counter, units: counts/second, arguments: fibrechannel-port	FC-COM packets send	Number of packets sent on this FibreChannel port.

Table 31 COM cluster I/O statistics

Statistic	Description
com-cluster-io.avg-lat type: reading, units: microseconds, arguments: cluster-id	Average latency in microseconds of all I/O from the local cluster to the other cluster in the last query period. Takes a cluster number as an argument
com-cluster-io.max-lat type:reading, units: microseconds, arguments: cluster-id	Maximum latency in microseconds of all I/O from the local cluster to the other cluster. Takes a cluster number as an argument.
com-cluster-io.min-lat reading, units: microseconds, arguments: cluster-id	Minimum latency in microseconds of all I/O from the local cluster to the other cluster. Takes a cluster number as an argument.
com-cluster-io.send-ops type:reading, units: none, arguments: cluster-id	Number of I/O send operations to the cluster.

Performance statistics retrievable by SNMP

The SNMP agent must be configured and started on the local cluster in order to retrieve MIB statistics.

Use the **snmp-agent configure** command to configure and start the SNMP agent.

Table 32 SNMP director level statistics (1 of 2)

Statistic MIB Name	Table	Description
vplexDirectorCpuIdle	vplexDirectorProcTable	% CPU Idle since the last time this statistic was retrieved.
vplexDirectorHeapUsed	vplexDirectorMemTable	Director Heap utilization percentage.
vplexDirectorFEOpsRead	vplexDirectorFETable	Total number of read operations through director's all FE ports.

Table 32 SNMP director level statistics (2 of 2)

vplexDirectorFEOpsWrite	vplexDirectorFETable	Total number of Write operations through director's all FE ports.
vplexDirectorFEOpsQueued	vplexDirectorFETable	Number of outstanding I/Os queued on director's FE ports.
vplexDirectorFEOpsActive	vplexDirectorFETable	Total number of outstanding I/Os active on director's FE ports.
vplexDirectorFEOpsAvgReadLatency	vplexDirectorFETable	Average latency in microseconds for all Read commands received on director's FE ports. This is the average latency of the last 100 reads performed at the director.
vplexDirectorFEOpsAvgWriteLatency	vplexDirectorFETable	Average latency in microseconds for all Write commands received on director's FE ports. This is average of the last 100 write operations at the director.
vplexDirectorBEOpsAvgReadLatency	vplexDirectorBETable	Average Latency in microseconds for all Read commands received on director's BE ports. This is average of the last 100 read operations at the director.
vplexDirectorBEOpsAvgWriteLatency	vplexDirectorFETable	Average latency in microseconds for all Write commands received on director's BE ports. This is average of the last 100 write operations.
vplexDirectorFEBytesRead	vplexDirectorFETable	Total number of bytes read on the director's FE ports.
vplexDirectorFEBytesWrite	vplexDirectorFETable	Total number of bytes Written on the director's FE ports.
vplexDirectorBEOpsRead	vplexDirectorBETable	Total number of read operations through director's all BE ports.
vplexDirectorBEOpsWrite	vplexDirectorBETable	Total number of Write operations through director's all BE ports.
vplexDirectorBEBytesRead	vplexDirectorBETable	Total number of bytes read on the director's BE ports.
vplexDirectorBEBytesWrite	vplexDirectorBETable	Total number of bytes written on the director's BE ports.

Table 33 SNMP port level statistics

Statistic MIB Name	Table	Description
vplexDirectorFEPortBytesRead	vplexDirectorFEPortTable	Number of bytes read on this FE port.
vplexDirectorFEPortBytesWrite	vplexDirectorFEPortTable	Number of bytes written on this FE port.
vplexDirectorBEPortBytesRead	vplexDirectorBEPortTable	Number of bytes read through this BE port.
vplexDirectorBEPortBytesWrite	vplexDirectorBEPortTable	Number of bytes written through this BE port.

Glossary

This glossary contains terms related to VPLEX federated storage systems. Many of these terms are used in these manual.

A

AccessAnywhere The breakthrough technology that enables VPLEX clusters to provide access to information between clusters that are separated by distance.

active/active A cluster with no primary or standby servers, because all servers can run applications and interchangeably act as backup for one another.

Active Directory A directory service included in most Windows Server operating systems. AD authenticates and authorizes users and computers in a network of Windows domain type.

active mirror A copy of data that is part of a local or remote mirroring service.

active/passive A powered component that is ready to operate upon the failure of a primary component.

array A collection of disk drives where user data and parity data may be stored. Devices can consist of some or all of the drives within an array.

asynchronous Describes objects or events that are not coordinated in time. A process operates independently of other processes, being initiated and left for another task before being acknowledged.

For example, a host writes data to the blades and then begins other work while the data is transferred to a local disk and across the WAN asynchronously. See also "synchronous."

B

bandwidth The range of transmission frequencies a network can accommodate, expressed as the difference between the highest and lowest frequencies of a transmission cycle. High bandwidth allows fast or high-volume transmissions.

backend port VPLEX director port connected to storage arrays (acts as an initiator).

bias When a cluster has the bias for a given DR1, it continues to service I/O to volumes on that cluster if connectivity to the remote cluster is lost (due to cluster partition or cluster failure). The bias for a specific volume is determined by the detach rules for the volume, the detach rules for the consistency group (if the volume is a member of a consistency group) and VPLEX Witness (if VPLEX Witness is deployed).

bit A unit of information that has a binary digit value of either 0 or 1.

block The smallest amount of data that can be transferred following SCSI standards, which is traditionally 512 bytes. Virtual volumes are presented to users as a contiguous lists of blocks.

block size The actual size of a block on a device.

byte Memory space used to store eight bits of data.

C

cache Temporary storage for recent writes and recently accessed data. Disk data is read through the cache so that subsequent read references are found in the cache.

cache coherency Managing the cache so data is not lost, corrupted, or overwritten. With multiple processors, data blocks may have several copies, one in the main memory and one in each of the cache memories. Cache coherency propagates the blocks of multiple users throughout the system in a timely fashion, ensuring the data blocks do not have inconsistent versions in the different processors caches.

cluster Two or more VPLEX directors forming a single fault-tolerant cluster, deployed as one to four engines.

cluster ID The identifier for each cluster in a multi-cluster deployment. The ID is assigned during installation.

cluster deployment ID A numerical cluster identifier, unique within a VPLEX cluster. By default, VPLEX clusters have a cluster deployment ID of 1. For multi-cluster deployments, all but one cluster must be reconfigured to have different cluster deployment IDs.

cluster IP seed The VPLEX IP seed is used to generate the IP addresses used by the internal components of the VPLEX. For more information about components and their IP addresses, refer to EMC VPLEX Installation and Setup Guide. Cluster ID is used by the virtualization software (inter director messaging, cluster identification).

clustering Using two or more computers to function together as a single entity. Benefits include fault tolerance and load balancing, which increases reliability and up time.

COM The intra-cluster communication (Fibre Channel). The communication used for cache coherency and replication traffic.

command line interface (CLI) Method of operating system or application software by typing commands to perform specific tasks.

consistency group A VPLEX structure that groups together virtual volumes and applies the same detach and failover rules to all member volumes. Consistency groups ensure the common application of a set of properties to the entire group. Create consistency groups for sets of volumes that require the same I/O behavior in the event of a link failure. There are two types of consistency groups:

- **Synchronous Consistency Groups** - Use write-through (synchronous) cache mode to write data to the underlying storage before an acknowledgement is sent to the host. Dependent on latency between clusters and the application's tolerance of the latency
- **Asynchronous Consistency Groups** - Use write-back (asynchronous) cache mode to write protect data by mirroring it to the memory another director in the cluster. Data is destaged asynchronously to the back-end storage arrays. Writes are acknowledged once the data has been committed to disk in write order

continuity of operations (COOP) The goal of establishing policies and procedures to be used during an emergency, including the ability to process, store, and transmit data before and after.

controller A device that controls the transfer of data to and from a computer and a peripheral device.

D

data sharing The ability to share access to the same data with multiple servers regardless of time and location.

detach rule Predefined rules that determine which cluster continues I/O when connectivity between clusters is lost. A cluster loses connectivity to its peer cluster due to cluster partition or cluster failure.

Detach rules are applied at two levels; to individual volumes, and to consistency groups. If a volume is a member of a consistency group, the group detach rule overrides the rule set for the individual volumes. Note that all detach rules may be overridden by VPLEX Witness, if VPLEX Witness is deployed.

device A combination of one or more extents to which you add specific RAID properties. Local devices use storage from only one cluster. In VPLEX Metro and Geo configurations, distributed devices use storage from both clusters.

director A CPU module that runs GeoSynchrony, the core VPLEX software. There are two directors (A and B) in each engine, and each has dedicated resources and is capable of functioning independently.

dirty data The write-specific data stored in the cache memory that has yet to be written to disk.

disaster recovery (DR) The ability to restart system operations after an error, preventing data loss.

discovered array An array that is connected to the SAN and discovered by VPLEX.

disk cache A section of RAM that provides cache between the disk and the CPU. RAMs access time is significantly faster than disk access time; therefore, a disk-caching program enables the computer to operate faster by placing recently accessed data in the disk cache.

distributed device A RAID 1 device whose mirrors are in different VPLEX clusters.

distributed file system (DFS) Supports the sharing of files and resources in the form of persistent storage over a network.

distributed RAID1 device (DR1) Distributed devices have physical volumes at both clusters in a VPLEX Metro or VPLEX Geo configuration for simultaneous active/active read/write access using AccessAnywhere™

E

engine Consists of two directors, management modules, and redundant power. Unit of scale for VPLEX configurations. Single = 1 engine, dual = 2 engines, Quad = 4 engines per cluster.

Ethernet A Local Area Network (LAN) protocol. Ethernet uses a bus topology, meaning all devices are connected to a central cable, and supports data transfer rates of between 10 megabits per second and 10 gigabits per second. For example, 100 Base-T supports data transfer rates of 100 Mb/s.

event A log message that results from a significant action initiated by a user or the system.

extent All or a portion (range of blocks) of a storage volume.

F

failover Automatically switching to a redundant or standby device, system, or data path upon the failure or abnormal termination of the currently active device, system, or data path.

fault domain A set of components that share a single point of failure. For VPLEX, the concept that every component of a Highly Available system is separated, so that if a fault occurs in one domain, it will not result in failure in other domains to which it is connected.

fault tolerance Ability of a system to keep working in the event of hardware or software failure, usually achieved by duplicating key system components.

Fibre Channel (FC) A protocol for transmitting data between computer devices. Longer distance requires the use of optical fiber; however, FC also works using coaxial cable and ordinary telephone twisted pair media. Fibre channel offers point-to-point, switched, and loop interfaces. Used within a SAN to carry SCSI traffic.

Fibre Channel over IP (FCIP) Combines Fibre Channel and Internet protocol features to connect SANs in geographically distributed systems.

field replaceable unit (FRU) A unit or component of a system that can be replaced on site as opposed to returning the system to the manufacturer for repair.

firmware Software that is loaded on and runs from the flash ROM on the VPLEX directors.

front end port VPLEX director port connected to host initiators (acts as a target).

G

geographically distributed system A system physically distributed across two or more geographically separated sites. The degree of distribution can vary widely, from different locations on a campus or in a city to different continents.

gigabit (Gb or Gbit) 1,073,741,824 (2^30) bits. Often rounded to 10^9.

gigabit Ethernet The version of Ethernet that supports data transfer rates of 1 Gigabit per second.

gigabyte (GB) 1,073,741,824 (2^30) bytes. Often rounded to 10^9.

global file system (GFS) A shared-storage cluster or distributed file system.

H

hold provisioning An attribute of a registered array that allows you to set the array as unavailable for further provisioning of new storage.

host bus adapter (HBA) An I/O adapter that manages the transfer of information between the host computers bus and memory system. The adapter performs many low-level interface functions automatically or with minimal processor involvement to minimize the impact on the host processors performance.

I

input/output (I/O) Any operation, program, or device that transfers data to or from a computer.

internet Fibre Channel protocol (iFCP) Connects Fibre Channel storage devices to SANs or the Internet in geographically distributed systems using TCP.

intranet A network operating like the World Wide Web but with access restricted to a limited group of authorized users.

internet small computer system interface (iSCSI) A protocol that allows commands to travel through IP networks, which carries data from storage units to servers anywhere in a computer network.

I/O (input/output) The transfer of data to or from a computer.

K

kilobit (Kb) 1,024 (2^10) bits. Often rounded to 10^3.

kilobyte (K or KB) 1,024 (2^10) bytes. Often rounded to 10^3.

L

latency Amount of time it requires to fulfill an I/O request.

LDAP Lightweight Directory Access Protocol, an application protocol that accesses and maintains distributed directory information services over an IP network.

load balancing Distributing the processing and communications activity evenly across a system or network so no single device is overwhelmed. Load balancing is especially important when the number of I/O requests issued is unpredictable.

local area network (LAN) A group of computers and associated devices that share a common communications line and typically share the resources of a single processor or server within a small geographic area.

local device A combination of one or more extents to which you add specific RAID properties. Local devices use storage from only one cluster.

logical unit number (LUN) Virtual storage to which a given server with a physical connection to the underlying storage device may be granted or denied access. LUNs are used to identify SCSI devices, such as external hard drives, connected to a computer. Each device is assigned a LUN number which serves as the device's unique address.

M

megabit (Mb) 1,048,576 (2^20) bits. Often rounded to 10^6.

megabyte (MB) 1,048,576 (2^20) bytes. Often rounded to 10^6.

metadata Information about data, such as data quality, content, and condition.

metavolume A storage volume used by the system that contains the metadata for all the virtual volumes managed by the system. There is one metadata storage volume per cluster.

Metro-Plex Two VPLEX Metro clusters connected within metro (synchronous) distances, approximately 60 miles or 100 kilometers.

mirroring The writing of data to two or more disks simultaneously. If one of the disk drives fails, the system can instantly switch to one of the other disks without losing data or service. RAID 1 provides mirroring.

mirroring services Mirroring features provided through a storage service profile.

miss An operation where the cache is searched but does not contain the data, so the data instead must be accessed from disk.

move data across profiles This is a new menu option under "Mobility Central".

N

namespace A set of names recognized by a file system in which all names are unique.

network System of computers, terminals, and databases connected by communication lines.

network architecture Design of a network, including hardware, software, method of connection, and the protocol used.

network-attached storage (NAS) Storage elements connected directly to a network.

network partition When one site loses contact or communication with another site.

O

Open LDAP Open source implementation of the Lightweight Directory Access Protocol (LDAP).

P

parity checking Checking for errors in binary data. Depending on whether the byte has an even or odd number of bits, an extra 0 or 1 bit, called a parity bit, is added to each byte in a transmission. The sender and receiver agree on odd parity, even parity, or no parity. If they agree on even parity, a parity bit is added that makes each byte even. If they agree on odd parity, a parity bit is added that makes each byte odd. If the data is transmitted incorrectly, the change in parity will reveal the error.

partition A subdivision of a physical or virtual disk, which is a logical entity only visible to the end user, not any of the devices.

R

RAID The use of two or more storage volumes to provide better performance, error recovery, and fault tolerance.

RAID 0 A performance-orientated striped or dispersed data mapping technique. Uniformly sized blocks of storage are assigned in regular sequence to all of the arrays disks. Provides high I/O performance at low inherent cost. No additional disks are required. The advantages of RAID 0 are a very simple design and an ease of implementation.

RAID 1 Also called mirroring, this has been used longer than any other form of RAID. It remains popular because of simplicity and a high level of data availability. A mirrored array consists of two or more disks. Each disk in a mirrored array holds an identical image of the user data. RAID 1 has no striping. Read performance is improved since either disk can be read at the same time. Write performance is lower than single disk storage. Writes must be performed on all disks, or mirrors, in the RAID 1. RAID 1 provides very good data reliability for read-intensive applications.

RAID leg A copy of data, called a mirror, that is located at a user's current location.

rebuild The process of reconstructing data onto a spare or replacement drive after a drive failure. Data is reconstructed from the data on the surviving disks, assuming mirroring has been employed.

RecoverPoint Appliance (RPA) Hardware that manages all aspects of data protection for a storage group, including capturing changes, maintaining the images in the journal volumes, and performing image recovery.

RecoverPoint cluster All connected RecoverPoint Appliances on both sides of the replication.

RecoverPoint site All RecoverPoint entities on one side of the replication.

Recovery Point Objective See "RPO" on page 282.

Recovery Time Objective See "RTO" on page 282.

redundancy The duplication of hardware and software components. In a redundant system, if a component fails then a redundant component takes over, allowing operations to continue without interruption.

registered array An array that is registered with VPLEX. Registration is required to make the array available for services-based provisioning. Registration includes connecting to and creating awareness of the array's intelligent features. Only VMAX and VNX arrays can be registered.

reliability The ability of a system to recover lost data.

remote direct memory access (RDMA) Allows computers within a network to exchange data using their main memories and without using the processor, cache, or operating system of either computer.

Replication set When RecoverPoint is deployed, a production source volume and one or more replica volume(s) to which it replicates.

restore source This operation restores the source consistency group from data on the copy target.

RPO Recovery Point Objective. The time interval between the point of failure of a storage system and the expected point in the past, to which the storage system is capable of recovering customer data. Informally, RPO is a maximum amount of data loss that can be tolerated by the application after a failure. The value of the RPO is highly dependent upon the recovery technique used. For example, RPO for backups is typically days; for asynchronous replication minutes; and for mirroring or synchronous replication seconds or instantaneous.

RTO Recovery Time Objective. Not to be confused with RPO, RTO is the time duration within which a storage solution is expected to recover from failure and begin servicing application requests. Informally, RTO is the longest tolerable application outage due to a failure of a storage system. RTO is a function of the storage technology. It may measure in hours for backup systems, minutes for a remote replication, and seconds (or less) for a mirroring.

S

scalability Ability to easily change a system in size or configuration to suit changing conditions, to grow with your needs.

services-based provisioning

simple network management protocol (SNMP) Monitors systems and devices in a network.

site ID The identifier for each cluster in a multi-cluster plex. By default, in a non-geographically distributed system the ID is 0. In a geographically distributed system, one clusters ID is 1, the next is 2, and so on, each number identifying a physically separate cluster. These identifiers are assigned during installation.

SLES SUSE Linux Enterprise Server is a Linux distribution supplied by SUSE and targeted at the business market.

small computer system interface (SCSI) A set of evolving ANSI standard electronic interfaces that allow personal computers to communicate faster and more flexibly than previous interfaces with peripheral hardware such as disk drives, tape drives, CD-ROM drives, printers, and scanners.

splitter EMC RecoverPoint write-splitting technology built into GeoSynchrony starting in 5.1.

storage area network (SAN) A high-speed special purpose network or subnetwork that interconnects different kinds of data storage devices with associated data servers on behalf of a larger network of users.

storage view A combination of registered initiators (hosts), front-end ports, and virtual volumes, used to control a hosts access to storage.

storage volume A Logical Unit Number (LUN) or unit of storage presented by the back end array.

stripe depth The number of blocks of data stored contiguously on each storage volume in a RAID 0 device.

striping A technique for spreading data over multiple disk drives. Disk striping can speed up operations that retrieve data from disk storage. Data is divided into units and distributed across the available disks. RAID 0 provides disk striping.

synchronous Describes objects or events that are coordinated in time. A process is initiated and must be completed before another task is allowed to begin.

For example, in banking two withdrawals from a checking account that are started at the same time must not overlap; therefore, they are processed synchronously. See also "asynchronous."

T

throughput 1. The number of bits, characters, or blocks passing through a data communication system or portion of that system.

2. The maximum capacity of a communications channel or system.

3. A measure of the amount of work performed by a system over a period of time. For example, the number of I/Os per day.

tool command language (TCL) A scripting language often used for rapid prototypes and scripted applications.

transfer size The size of the region in cache used to service data migration. The area is globally locked, read at the source, and written at the target. Transfer-size can be as small as 40 K, as large as 128 M, and must be a multiple of 4 K. The default value is 128 K.

A larger transfer-size results in higher performance for the migration, but may negatively impact front-end I/O. This is especially true for VPLEX Metro migrations. Set a large transfer-size for migrations when the priority is data protection or migration performance.

A smaller transfer-size results in lower performance for the migration, but creates less impact on front-end I/O and response times for hosts. Set a smaller transfer-size for migrations when the priority is front-end storage response time.

transmission control protocol/Internet protocol (TCP/IP) The basic communication language or protocol used for traffic on a private network and the Internet.

U

uninterruptible power supply (UPS) A power supply that includes a battery to maintain power in the event of a power failure.

universal unique identifier (UUID) A 64-bit number used to uniquely identify each VPLEX director. This number is based on the hardware serial number assigned to each director.

V

virtualization A layer of abstraction implemented in software that servers use to divide available physical storage into storage volumes or virtual volumes.

virtual volume Unit of storage presented by the VPLEX front end ports to hosts. A virtual volume looks like a contiguous volume, but can be distributed over two or more storage volumes.

W

wide area network (WAN) A geographically dispersed telecommunications network. This term distinguishes a broader telecommunication structure from a local area network (LAN).

world wide name (WWN) A specific Fibre Channel Name Identifier that is unique worldwide and represented by a 64-bit unsigned binary value.

write-back mode A caching technique where the completion of a write request is communicated as soon as the data is in cache, with writes to disk occurring at different times. Write-back is faster than write-through, but risks a system failure before the data is safely written to disk.

write-through mode A caching technique in which the completion of a write request is communicated only after data is written to disk. This is almost equivalent to non-cached systems, but with data protection.

Index

N

O

P

R

S