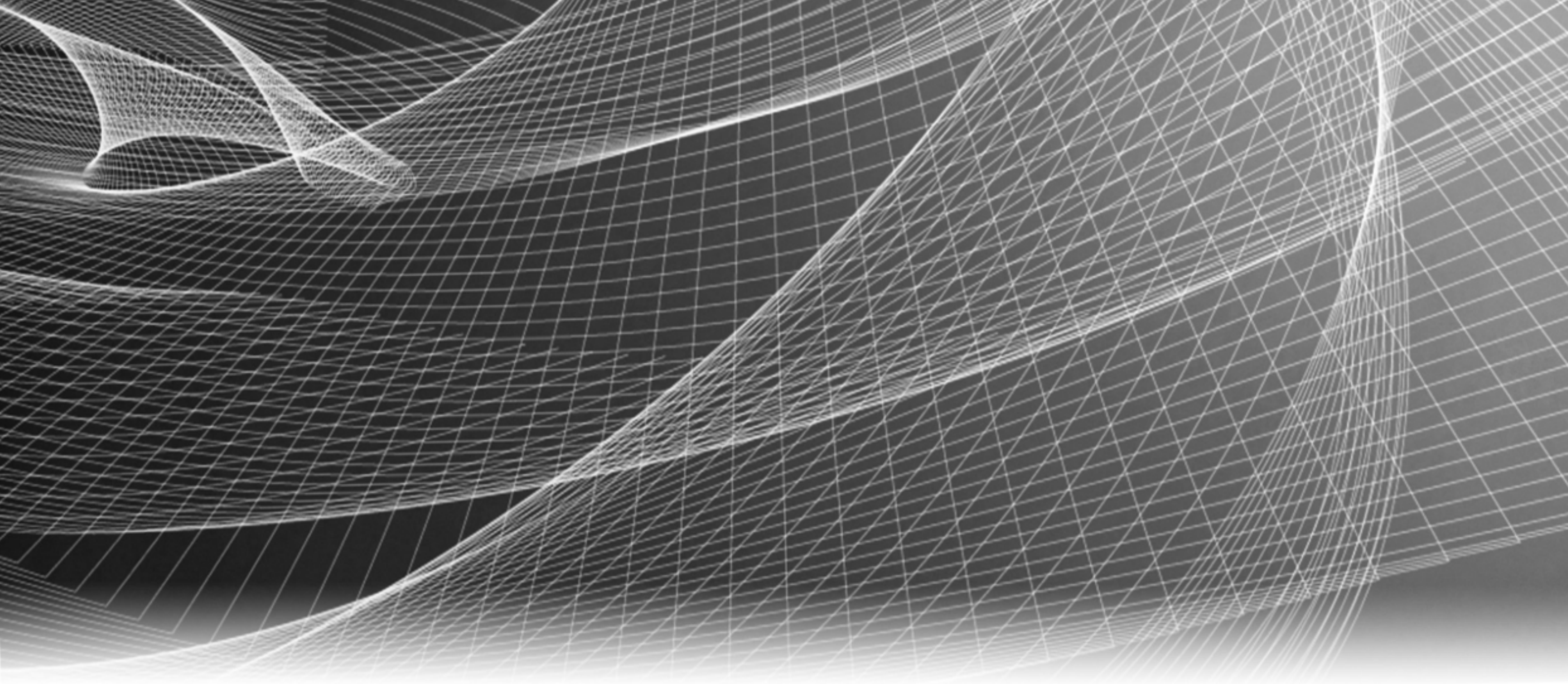

EMC® VPLEX®

GeoSynchrony

Element Manager API Guide

P/N 302-000-781
REV 01

Published March, 2014

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

For the most up-to-date regulatory document for your product line, go to the Technical Documentation and Advisories section on EMC Powerlink.

CONTENTS

Index

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 document is part of the VPLEX documentation set, and describes the VPLEX Restful API.

This guide is intended for use by customers and service providers who use the Restful API to configure and manage a storage environment.

New Commands for VPLEX

- amp register
- amp unregister
- array register
- configuration short-write
- configuration subnet clear
- configuration subnet remote-subnet add
- configuration subnet remote-subnet remove
- configuration sync-time clear
- configuration sync-time show
- device mirror-isolation disable
- device mirror-isolation enable
- device mirror-isolation show
- export initiator-port show-logins
- iscsi chap back-end add credentials
- iscsi chap back-end disable
- iscsi chap back-end enable
- iscsi chap back-end list-credentials
- iscsi chap back-end remove-credentials
- iscsi chap back-end remove-default-credential
- iscsi chap back-end set-default-credential
- iscsi chap front-end add credentials
- iscsi chap front-end disable
- iscsi chap front-end enable
- iscsi chap front-end list-credentials
- iscsi chap front-end remove-credentials
- iscsi chap front-end remove-default-credential

- iscsi chap front-end set-default-credential
- iscsi check-febe-connectivity
- iscsi isns add
- iscsi isns list
- iscsi isns remove
- iscsi sendtargets add
- iscsi sendtargets list
- iscsi sendtargets rediscover
- iscsi sendtargets remove
- iscsi targets list
- iscsi targets logout
- storage-tool compose
- user event-server add-user
- user event-server change-password
- virtual-volume provision

Modified Commands for VPLEX

- authentication directory-service configure - The **-m option** is no longer supported. Added **-g and -u options** for managing access to the management server by groups and users. New example added for configuring the Active Directory Service with custom attributes, and a new example for configuring the Open LDAP server on the management server.
- authentication directory-service map - Added note about the **-m**, **-u**, and **-g** command options. Removed ldap.conf file sentence from command description. Added new examples for user and user group mappings to VPLEX cluster.
- authentication directory service show - Changed the command output example and mapped principal description in Table 1 to specify users and/or groups are mapped to the directory server.
- authentication directory-service unmap - Added note about the **-m**, **-u**, and **-g** command options. Removed ldap.conf file sentence from command description.
- cluster-status - New output examples.
- configure metadata-backups - Added an Important note in the command description.
- configuration connect-remote-directors - New command options **--remote-ip** and **--force** and new examples.
- configuration join-clusters - New command option **--remote-ip** and new example.
- configuration subnet clear - New command to clear subnets in all subnet contexts in the new /clusters/*/connectivity context. Use the legacy **subnet clear** command to clear subnets in /clusters/*/cluster-connectivity/subnets.
- configuration sync-time - Changed to reflect that cluster 1 synchronizes to an external NTP server, and cluster 2 synchronizes to cluster 1. Replaced example for running a synchronization task on cluster 2.
- connectivity validate-wan-com - **--protocol** changed to **FC** and **UDT**
- ds dd set-log - Changed the warning message for the **cancel** option.
- export initiator-port register - New information about working with iSCSI initiators
- health-check - New output examples
- logging-volume create - Added new **vias-based** attribute to Table 17 Logging volume display fields.
- management-server set-ip - Changed IPv6 address example
- monitor create - Changed example for collecting statistics on front-end port.

- monitor stat-list - Replaced outdated command output example with new example showing categories.
- ndu start - Changed command option from **--force-with-unreachable-cws** to **--skip-cws-upgrade**. Changed syntax and description.
- remote-clusters add-addresses - Renamed to **configuration remote-clusters add-addresses**
- remote-clusters clear-addresses - Renamed to **configuration remote-clusters clear-addresses**
- rp import-certificate - Added a cautionary note about VPLEX Integrated Array Services provisioning.
- rp rpa-cluster add - Added consistency group contexts and descriptions for copies, links, and replication sets in Table 19 RecoverPoint display fields. New examples for the consistency group contexts added.
- storage-volume-unclaim - Added argument **--return-to-pool.** This argument returns the storage capacity of each VIAS-based volume to the pool on the corresponding storage array. Additional note about differing results when using this command on VIAS and non-VIAS based storage arrays.

Removed Commands for VPLEX

- ds rule island-size
- ptov describe-be-zoning
- ptov lun-mask-be-storage
- ptov query-be-storage
- ptov suggest-be-zoning-and-masking
- ptov suggest-ports-to-use
- ptov verify-be-storage
- ptov verify-be-zoning

New Context for VPLEX

- connectivity/ - Configure connectivity between back-end storage arrays, front-end hosts, local directors, port-groups and inter-cluster WANs.

Context Tree Upgrade

The cluster-connectivity context will become deprecated in a future version. It is being replaced with the Connectivity context, which is available in this release. Use the new context whenever possible.

The following changes apply:

- Any configuration that exists in cluster-connectivity/ will automatically be reflected in the connectivity/wan-com/ context and vice versa. For example, an existing wan-com configuration will be displayed in connectivity/wan-com.
- All the attributes from the cluster-connectivity/ context are present in the connectivity/wan-com/ context.
- Commands to change the remote-cluster-addresses attribute:

  Old: cluster-connectivity remote-clusters add-addresses

  New: configuration remote-clusters add-addresses

  Old: cluster-connectivity remote-clusters clear-addresses

  New: configuration remote-clusters clear-addresses

- Member-ports is now a context instead of an attribute. The fields that are part of the string representation of a member-port under the cluster-connectivity context are now table columns in a long listing (ll) of a connectivity/**/member-ports/ context.

  Old: cluster-connectivity/port-groups/port-group-0/member-ports (attribute)

  New: connectivity/wan-com/port-groups/ip-port-group-0/member-ports (context)

- Inside cluster-connectivity/port-groups/port-group-0:
  - “subnet” is the name of a subnet from cluster-connectivity/subnets
  - “option-set” is the name of a subnet from cluster-connectivity/option-sets

  If the subnet=”cluster-1-SN00” and option-set=”optionset-com-0”, then

  Old: cluster-connectivity/subnets/cluster-1-SN00/

  New: connectivity/wan-com/port-groups/ip-port-group-0/subnet/

  Old: cluster-connectivity/option-sets/optionset-com-0

  New: connectivity/wan-com/port-groups/ip-port-group-0/option-set/
- There is no longer any need or ability to set the subnet used by a port-group.
- The cluster-connectivity/option-sets context has no analogue under connectivity/. Each port-group under connectivity/wan-com/port-groups has a permanent option-set sub-context.
- The cluster-connectivity/subnets context has no analogue under connectivity/. Each port-group under connectivity/wan-com/port-groups has a permanent subnet sub-context.

  ‘subnet create’ - no analogue under connectivity. Use the permanent subnet in the port-group.

  ‘subnet destroy’ - no analogue under connectivity.

  ‘subnet modify’ - use the **set** command to change the attributes.

  ‘subnet clear’ - use the **configuration subnet clear** command.

Cluster-connectivity Context (Deprecated)

Cluster-connectivity

This context contains information for cluster connectivities.

Context

/clusters/cluster-*/cluster-connectivity

Position

Parent - cluster-*

Children - Option-sets (deprecated), Port-groups (deprecated), Subnets (deprecated)

Context-Specific Commands

remote-clusters

Data Structure

discovery-address

discovery-port

listening-port

remote-cluster-addresses

Notes

This context is replaced by:

/clusters/cluster-*/connectivity/wan-com.

Option-sets

Option-set configuration for IP wan-com.

Context

/clusters/cluster-*/cluster-connectivity/option-sets/*

Position

Parent - cluster-connectivity (deprecated)

Children - None

Context-Specific Commands

None

Data Structure

connection-open-timeout

keepalive-timeout

socket-buf-size

Notes

The currently-used option-set context is replaced by:
/clusters/cluster-*/connectivity/wan-com/port-groups/ip-port-group-*/option-set

Port-groups

A communication channel for IP wan-com composed of one port from each director of the local cluster.

Context

/clusters/cluster-*/cluster-connectivity/port-groups/port-group-0

/clusters/cluster-*/cluster-connectivity/port-groups/port-group-1

Position

Parent - cluster-connectivity

Children - None

Context-Specific Commands

None

Data Structure

enabled

member-ports

option-set

subnet

Notes

This context is replaced by Ethernet port-groups:

/clusters/cluster-*/connectivity/wan-com/port-groups/ip-port-group-0

/clusters/cluster-*/connectivity/wan-com/port-groups/ip-port-group-1

Subnets

Network configuration for IP wan-com.

Context

/clusters/cluster-*/cluster-connectivity/subnets/*

Position

Parent - cluster-connectivity

Children - None

Context-Specific Commands

In container context:

clear

create

destroy

modify

In subnet instance context:

None

Data Structure

cluster-address

gateway

mtu

prefix

proxy-external-address

remote-subnet-address

Notes

The currently-used subnet context is replaced by:

/clusters/cluster-*/connectivity/wan-com/port-groups/ip-port-group-*/subnet

Named subnet instance contexts are created in the subnets container context by the command **subnet create** and are removed by the command **subnet destroy.**

New Connectivity Context (Upgrade)

Connectivity

This context contains information about all network connectivities.

Context

/clusters/cluster-*/connectivity

Position

Parent - cluster-*

Children - back-end, front-end, local-com, wan-com

Context-Specific Commands

None

Data Structure

None

Notes

This context replaces:

/clusters/cluster-*/cluster-connectivity

Back-end

Configuration of back-end connectivity to arrays.

Context

/clusters/cluster-*/connectivity/back-end

Position

Parent - connectivity

Children - port-groups

Context-Specific Commands

None

Data Structure

None

Front-end

Configuration of front-end connectivity to hosts.

Context

/clusters/cluster-*/connectivity/front-end

Position

Parent - connectivity

Children - port-groups

Context-Specific Commands

None

Data Structure

None

Local-com

Configuration for local-com inter-director connectivity.

Context

/clusters/cluster-*/connectivity/local-com

Position

Parent - connectivity

Children - port-groups

Context-Specific Commands

None

Data Structure

None

Wan-com

Configuration for wan-com inter-cluster connectivity.

Context

/clusters/cluster-*/connectivity/wan-com

Position

Parent - connectivity

Children - port-groups

Context-Specific Commands

remote-clusters add-addresses

remote-clusters clear-addresses

Data Structure

discovery-address

discovery-port

listening-port

remote-cluster-addresses

Port-groups

Communication channels formed by one port from each director in the local cluster.

Context

/clusters/cluster-*/connectivity/*/port-groups

Position

Parent - back-end, front-end, local-com, wan-com

Children - Ethernet-port-group, Fc-port-group

Context-Specific Commands

None

Data Structure

None

Notes

Port-groups are named according to the type (Ethernet: ip/iscsi, Fibre Channel: fc) and numbering of the ports they contain. The existence of port-group instance contexts is determined by the ports on the system. IP port-groups will only exist if Ethernet ports exist and are assigned to the communication role associated with this context. Likewise, FC port-groups will only exist if there are appropriate Fibre Channel ports.

Ethernet port-group

Communication channel composed of one Ethernet port from each director in the local cluster.

Context

/clusters/cluster-*/connectivity/*/port-groups/ip-port-group-*

/clusters/cluster-*/connectivity/*/port-groups/iscsi-port-group-*

Position

Parent - port-groups

Children - member-ports, option-set, subnet

Context-Specific Commands

subnets clear

subnet remote-subnet add

subnet remote-subnet remove

Data Structure

enabled

Notes

Port-groups are named according to the role (local-com/wan-com: ip, front-end/back-end: iscsi) and numbering of the ports they contain. The existence of port-group instance contexts is determined by the ports on the system. IP/iSCSI port-groups will only exist if Ethernet ports exist and are assigned to the role associated with this context.

Fibre Channel port-group

Communication channel composed of one Fibre Channel port from each director in the local cluster.

Context

/clusters/cluster-*/connectivity/*/port-groups/fc-port-group-*

Position

Parent - port-groups

Children - member-ports

Context-Specific Commands

None

Data Structure

enabled

Notes

Port-groups are named according to the numbering of the ports they contain. The existence of port-group instance contexts is determined by the ports on the system. FC port-groups will only exist if Fibre Channel ports exist and are assigned to the role associated with this context.

Member-ports

A member port in a port-group.

Context

/clusters/cluster-*/connectivity/*/port-groups/*/member-ports/director-*

Position

Parent - Ethernet-port-group, Fc-port-group

Children - None

Context-Specific Commands

None

Data Structure

address director enabled engine port-name

Notes

Each member-port sub-context is named for the director to which the member port belongs. This naming convention avoids the name collision caused when a port has the same name on each director.

A long listing (ll) of the member-ports container context summarizes the information from each member port:

```
Director        Port   Enabled  Address
--------------  -----  -------  -------------
director-1-1-A  ETH04  enabled  192.168.11.35
director-1-1-B  ETH04  enabled  192.168.11.36
director-1-2-A  ETH04  enabled  192.168.11.37
director-1-2-B  ETH04  enabled  192.168.11.38
```

Option-sets

Option-set configuration for Ethernet port-groups.

Context

/clusters/cluster-*/connectivity/*/port-groups/ip-port-group*/option-set
/clusters/cluster-*/connectivity/*/port-groups/iscsi-port-group*/option-set

Position

Parent - Ethernet-port-group

Children - None

Context-Specific Commands

None

Data Structure

connection-open-timeout

keepalive-timeout

socket-buf-size

IP-com and iSCSI Subnets

Network configuration for Ethernet port-groups.

Context

/clusters/cluster-*/connectivity/*/port-groups/ip-port-group-*/subnet
/clusters/cluster-*/connectivity/*/port-groups/iscsi-port-group-*/subnet

Position

Parent - Ethernet-port-group

Children - None

Context-Specific Commands

All roles:

clear

Front-end and Back-end only:

remote-subnets add

remote-subnets remove

Data Structure

The attributes of a subnet depend on the role with which it is associated.

All roles:

mtu

prefix

Front-end and Back-end only:

gateway

remote-subnets

Wan-com only:

cluster-address

gateway

proxy-external-address

remote-subnet-address

Related documentation

Related documents (available on EMC Support Online) include:

- *EMC VPLEX Release Notes for GeoSynchrony Releases 5.3*
- *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 Reference 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

<table>
<tr><th></th><th></th></tr>
<tr><td></td><td>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</td></tr>
<tr><td><i>Italic</i></td><td>Used in all text (including procedures) for:
• Full titles of publications referenced in text
• Emphasis (for example a new term)
• Variables</td></tr>
<tr><td><code>Courier</code></td><td>Used for:
• System output, such as an error message or script
• URLs, complete paths, filenames, prompts, and syntax when shown outside of running text</td></tr>
<tr><td><b><code>Courier bold</code></b></td><td>Used for:
• Specific user input (such as commands)</td></tr>
<tr><td><i><code>Courier italic</code></i></td><td>Used in procedures for:
• Variables on command line
• User input variables</td></tr>
<tr><td>[ ]</td><td>Square brackets enclose optional values</td></tr>
<tr><td>|</td><td>Vertical bar indicates alternate selections - the bar means “or”</td></tr>
<tr><td>{ }</td><td>Braces indicate content that you must specify (that is, x or y or z)</td></tr>
<tr><td>...</td><td>Ellipses indicate nonessential information omitted from the example</td></tr>
</table>

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`

CHAPTER 1
VPLEX Element Manager API

This chapter describes VPLEX Element Manager API. It discusses the following topics:

Overview

VPLEX Element Manager API uses Representational State Transfer (REST) software architecture.

REST is used for distributed systems, such as the Web and enables software developers and other users to use the API to create scripts to run VPLEX command- level interfaces.

Some types of VPLEX CLI commands are not supported by the API:

- Session aware commands. For example: **cd**, **popd**, **pushd**.
- Commands that manage the security aspects of the product.

See "Supported VPLEX CLI commands" on page 35.

Command options are sent and responses are received as JSON strings. URIs are used to identify VPLEX objects and contexts.

- VPLEX base URI (IPv4): **https://***management-server-ip:port_number***/vplex/**
- VPLEX base URI (IPv6): **https:***//[mgmtserver_ipv6_addr]*

  For example:

  ```
  https://[3ffe:80c0:22c:803c:215:17ff:fed3:207]/smsflex/VPlexConsole
  .html
  ```

- Default secure port for HTTPS: **443**

VPLEX supports the following RESTful requests:

- HTTPS **GET** - Retrieves context, context attributes, and sub-contexts.
- HTTPS **PUT** - Sets values for a writable context attribute.
- HTTPS **POST** - Executes a VPLEX CLI command.

Navigation commands (**cd**, **pushd**, **popd**) and session-based commands (**capture** and **monitor**) are not supported.

Authentication

All system management operations require user authentication.

Each operation requires that the authenticating user be defined to the VPLEX SMS in a descriptively named role (**service** or **admin**, for example).

To authenticate using the VPLEX Element Manager API, a set of headers must be included with each request. Authentication headers must include the username and corresponding password. Each request is a separate session and requests do not span multiple sessions.

IMPORTANT

You cannot use a browser to authenticate. When specifying custom headers with a utility, ensure appropriate authentication is supplied with each request.

Example:

```
curl -k -H "Username:admin" -H "Password:administrator"
   https://10.12.198.67/vplex/engines
```

URI special characters

The VPLEX base URI supports special characters listed in Table 1.

Table 1 URI special characters

Character	Use
*	Any valid character, as used in CLI syntax. See the *VPLEX CLI Guide* for examples.
?	For specifying a context attribute (for listing or setting). For example: Context: https://*hostname*/vplex/*context path* Attribute: https://*hostname*/vplex/*context path*?*attrib*
+	Replaces a space between command strings. For example: CLI: cluster status URI: https://*mgmt-server-ip*/vplex/cluster+status

Note: The term *mgmt-server-ip* refers to the IP address or host name of the VPLEX management server.

VPLEX Element Manager API Response Structure

Each API call returns some sort of response. In some cases, one or all of the response fields may be empty.

VPLEX Element Manager API uses JSON as its Serialization format.

Before using the API, ensure you understand JSON terminologies and conventions as described at www.json.org.

Specifically, the VPLEX Element Manager API follows the response structure definitions listed in Table 2.

Table 2 JSON Response Structure Definitions

Convention	Basis for Definition
JSONObject	Defined at http://www.json.org
JSONArray	
JSONString	
JSONTrue	Unquoted word true defined at http://www.json.org
JSONFalse	Unquoted word false defined at http://www.json.org
JSONNumber	Unquoted number defined at http://www.json.org
JSONNull	Unquoted word null defined at http://www.json.org
JSONValue	Value defined at http://www.json.org. Values are any combination of the types defined in Table 2.

Two types of response structures are supported:

- **F0** - The original response structure, supported prior to release 5.2. F0 is the default response structure returned by the VPLEX Element Manager API.
- **F1** - The new response structure, supported from release 5.2 forward. The F1 response structure will co-exist with F0 for backward compatibility purposes. Obtain F1 response structure by specifying the HTTP **Accept** header in the request.

The response structures are selected using the Accept Header. F0 is the default if no Accept Header is specified.

See "VPLEX Element Manager API Headers" on page 25 for valid values used by the **Accept** Header.

F0 Response Structure

The structure of an F0 response consists of four levels.

```
{
  "response": {
    "context": JSONArray(Context JSONObject)|JSONNull,
    "message": JSONString | JSONNull,
    "exception": JSONString | JSONNull,
    "custom-data": JSONString | JSONNull
  }
}
```

- **Level 1: ("response")** – The root JSONObject for this structure is named "response".

- All four of the fields at level 1 (`context`, `message`, `exception`, `custom-data`) never contain non-null data at the same time. At minimum, one contains `JSONNull`. All four may simultaneously contain `JSONNull`.
- `message` contains Information or Warning messages related to the response. `message` may also contain messages printed by the CLI. `message` will always contain `JSONNull` if context is non-null.
- `exception` contains the exception generated during the operation. It may be non-null.
- `context` contains non-null data for an HTTP GET request for a valid URL.
- `custom-data` field contains non-null data for an HTTP POST request for a valid URI. Custom data is always the serialized result of a VPLEX CLI Command execution. Since this result does not have a constant structure, it cannot be parsed into JSON and so is the name of the field. This data sometimes has a tabular layout and, if printed over 80-columns (default width of a CLI Shell), looks like a table.
- HTTP Put does not generate any response data. It only generates a Success header, along with other relevant headers.

◆ Level 2: (“`context`”) - `JSONObject context` consists of the following:

```
{
  "name": JSONString,
  “type”: JSONString,
  "parent": JSONString,
  "attributes”: JSONArray(Attribute JSONObject)| JSONNull,
  “children”: JSONArray(Children JSONObject)| JSONNull
}
```

◆ Level 3: (**`JSONObject Attribute`**) - `JSONObject Attribute` consists of the following:

```
{
  "name": JSONString,
  “value”: JSONArray(JSONString) | JSONString | JSONNull
}
```

◆ Level 4: (**`JSONObject Children`**) - `JSONObject Children` consists of the following:

```
{
  "name": JSONString,
  “type”: JSONString
}
```

- The values of the `children` name fields are the child contexts of the current contexts. These values can be used to form URIs; used as such, you can recursively fetch the entire CLI context tree.

Example: F0 Response Structure GET request for valid URI

IMPORTANT

The order in which fields are serialized is not deterministic.

```
{
  "response": {
    "context": [
      {
        "name":"cluster-1",
        "parent":"/clusters",
        "type":"cluster"
        "attributes": [
          {"value":"true", "name":"allow-auto-join"},
          {"value":"0", "name":"auto-expel-count"},
          {"value":"0", "name":"auto-expel-period"},
          {"value":"0", "name":"auto-join-delay"},
          {"value":"1", "name":"cluster-id"},
          {"value":"true", "name":"connected"},
          {"value":"synchronous", "name":"default-cache-mode"},
          {"value":"director-1-1-A", "name":"default-director"},
          {"value": ["director-1-1-B", "director-1-1-A"], "name":"director-names"},
          {"value": [], "name":"health-indications" },
          {"value":"ok", "name":"health-state" },
          {"value":"1", "name":"island-id" },
          {"value":"cluster-1", "name":"name"},
          {"value":"ok", "name":"operational-status"},
          {"value": [], "name":"transition-indications" },
          {"value":[], "name":"transition-progress" }
        ],
        "children": [
          {"type":"cluster-connectivity", "name":"cluster-connectivity"},
          {"type":"consistency-groups", "name":"consistency-groups"},
          {"type":"devices", "name":"devices"},
          {"type":"exports", "name":"exports"},
          {"type":"storage-elements", "name":"storage-elements"},
          {"type":"system-volumes", "name":"system-volumes"},
          {"type":"uninterruptible-power-supplies","name":"uninterruptible-power-supplies"},
          {"type":"virtual-volumes", "name":"virtual-volumes"}
        ]
      }
    ],
    "message": null,
    "exception": null,
    "custom-data": null
  }
}
```

F1 Response Structure

F1 response structure is significantly compact compared to F0. It provides easier parsing and improves performance for both the server and client. This efficiency can be especially beneficial when working with a scaled VPLEX System.

The structure of an F1 response consists of two levels, compared to an F0 response structure's four levels.

- **Level 1: (`“response”`)** – The root `JSONObject` for this structure is named “response”. See “F0 Response Structure” on page 21 for details. Level 1 structure is identical for both F0 and F1 structures.
- **Level 2: (`JSONObject context`)** - `JSONObject context` consists of the following:

```
{
  "name": JSONString,
  “type”: JSONString,
  "parent": JSONString,
  “attribute4”:  JSONValue,
  “attribute5”:  JSONValue,
  ?
  ?
  “children”: JSONArray(JSONString)| JSONNull
}
```

- In F1 structure, the `context` object does not have an `attributes` field. What was contained within the value of `attributes` field in F0 is now included with the `context` itself in F1. At minimum, there are three non-null fields in each context (`name`, `type`, and `parent`).
- In F0 structure, the value types in the `attributes` field were specified in `JSONString`s (quoted strings). In F1 structure, they are specified in *`JSONValue`*, rendering the response more JSON-compliant. A *`JSONValue`* is one of the following:
  - `JSONArray`(*`JSONValue`*)
  - `JSONString`
  - `JSONTrue`
  - `JSONFalse`
  - `JSONNumber`
  - JSONNull
- The value for the children field is now a `JSONArray` of `JSONStrings`. Each of the values is a child context for the current context and can be used to form URIs; used as such, you can recursively fetch the entire CLI context tree.

Example: F1 Response Structure GET request for valid UR

```
{
  "response": {
    "context": [
      {
        "name":"cluster-1",
        "parent":"/clusters",
        "type":"cluster"
        "allow-auto-join": 0,
        "auto-expel-period": 0,
        "auto-join-delay": 0,
        "cluster-id": 1,
        "connected": true,
        "default-cache-mode": "synchronous",
        "default-director": "director-1-1-A",
        "director-names": ["director-1-1-B", "director-1-1-A"],
        "health-indications": [],
        "health-state": "ok",
        "island-id": 1,
        "operational-status": "ok"
        "transition-indications": [],
        "transition-progress": [],
        "children": ["cluster-connectivity", "consistency-groups", "devices",
                     "exports", "storage-elements", "system-volumes",
                     "uninterruptible-power-supplies, "virtual-volumes" ]
      }
    ],
    "message": null,
    "exception": null,
    "custom-data": null
  }
}
```

VPLEX Element Manager API Headers

This section provides a description of various HTTP Headers that the VPLEX Element Manager API supports.

See the HTTP specification at http://www.ietf.org/rfc/rfc2616.txt.

Note: HTTP header field *names* are case insensitive. Field *values* are case sensitive.

Request Headers

Request Headers are specified by VPLEX Element Manager API clients.

When composing and sending the request to the server, additional request headers may be provided by the specific programming language, script, or tool used by the client (for example, Python, Perl, Fiddler). Some of these additional headers are used by VPLEX Element Manager API and some are ignored. Table 3 does not list such third-party headers.

Table 3 Request Header Fields

Header Field Name	Description	Optional?	Example Header Field Value
Username	VPLEX username	No	service
Password	Password for *username*	No	password
Accept	Accept header specifying the type of response the client expects	Yes	See "Accept Headers" on page 25 for more information.

Accept Headers

Accept Headers are optional. If not specified, VPLEX Element Manager API returns the original JSON response format. If specified, the client is able to receive the new JSON response Format 1 (F1). The F1 format is more efficient and offers better performance when response size is large, producing approximately 80% less response compared to the original JSON Format 0 (F0).

Accept Header field values are composed of: *mainvalue*;*token1*;*token2*

`mainvalue` is **application/json** per the HTTP specification at http://www.ietf.org/rfc/rfc2616.txt.

`token1` and `token2` are **format=0|format=1** and **prettyprint=0|prettyprint=1**, respectively, separated by a semi-colon (;) and can be in any order. **format=0** is the original default JSON format. To use the prettyprint feature, specify **prettyprint=0**.

Specifying prettyprint=0 option in your header can significantly reduce the size of the F1 response. Typically, an F1 response using prettyprint=0 can be 80% less in size, compared to the F0 response format. Therefore, F1 response structure is the preferred response structure for clients.

See Table 4 for descriptions of possible Accept Header values.

Table 4 Accept Header Values

Accept Header Field Value	Description
None (Accept header not used)	Accepts F0, prettyprint on
application/json;format=1	Accepts F1.
application/json;format=0	Accepts F0.
application/json;format=0;prettyprint=0	Accepts F0, prettyprint on.
application/json;prettyprint=1	Accepts F0, prettyprint off (minified).
application/json;format=1;prettyprint=0	Accepts F1, prettyprint on.
application/json;format=1;prettyprint=1	Accepts F1, prettyprint off (minified).

Response Headers

Response Headers are headers returned by VPLEX Element Manager API in response to a client's request. Some headers will not be present in every response.

See Table 5 for descriptions of field names in Response Headers.

Table 5 Response Header Fields

Header Field Name	Description
Content-Type	MIME Type of response being returned by Server. • If Accept Header is specified and is a valid Accept Header, value will be same as Accept Header. • If DownloadFile feature is used, value is based on the type of the file. For example, a ZIP file will have value as **Content-Type: application/zip**. • Else, it is **application/json.**
Content-Length	Number of bytes in response.
Content-Disposition	Header is returned by API only when using the Downloadfile feature. Header contains name of the file along with its extension.

Table 5 Response Header Fields

Header Field Name	Description
Location	If a command takes more than sixty (60) seconds to complete, it is automatically converted to an asynchronous command and a task ID is created for the command. An HTTP 202 is returned with the Location Header. The value of the Location header is a URI which may be used to check the status of the command.
Server	Value will always be: **Apache-Coyote/1.1**.
Transfer-Encoding	Value will always be: **chunked.**
Date	Value will be the time stamp issued when the response was generated by the server.

JSON Response Structure

The following specific structure is intended to show syntax only. For examples of response structures you might actually create, see “Sample Message Responses” on page 30.

Response Syntax

```
{
 "response": {
     "message": null,
     "exception": null,
     "context": [
         {
             "name": "cluster-1",
             "children": [
                 {
                     "type": "cluster-connectivity",
                     "name": "cluster-connectivity"
                 },
                 {
                     "type": "consistency-groups",
                     "name": "consistency-groups"
                 },
                 {
                     "type": "devices",
                     "name": "devices"
                 },
                 {
                     "type": "exports",
                     "name": "exports"
                 },
                 {
                     "type": "storage-elements",
                     "name": "storage-elements"
                 },
                 {
                     "type": "system-volumes",
                     "name": "system-volumes"
                 },
                 {
                     "type": "uninterruptible-power-supplies",
                     "name": "uninterruptible-power-supplies"
                 },
                 {
                     "type": "virtual-volumes",
                "name": "virtual-volumes"
            }
        ],
        "parent": "/clusters",
        "attributes": [
            {
                "value": "true",
                "name": "allow-auto-join"
            },
            {
                "value": "0",
                "name": "auto-expel-count"
            },
            {
                "value": "0",
                "name": "auto-expel-period"
            },
            {
                "value": "0",
                "name": "auto-join-delay"
            },
            {
                "value": "1",
                "name": "cluster-id"
            },
            {
                "value": "true",
                "name": "connected"
            },
            {
                "value": "synchronous",
                "name": "default-cache-mode"
            },
            {
                "value": "director-1-1-A",
                "name": "default-director"
            },
            {
                "value": [
                    "director-1-1-B",
                    "director-1-1-A"
                ],
                "name": "director-names"
            },
            {
                "value": [

                ],
                "name": "health-indications"
            },
            {
                "value": "ok",
                "name": "health-state"
            },
            {
                "value": "1",
                "name": "island-id"
            },
            {
                "value": "cluster-1",
                "name": "name"
            },
            {
                "value": "ok",
                "name": "operational-status"
            },
                    {
                        "value": [

                        ],
                        "name": "transition-indications"
                    },
                    {
                        "value": [

                        ],
                        "name": "transition-progress"
                    }
                ],
                "type": "cluster"
            }
        ],
        "custom-data": null
    }
}
```

GET response structure

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Date: Wed, 16 Feb 2011 07:09:35 GMT

{
   "response": {
       "context": [
          "name": "...",
          "type": "...",
           "parent": "...",
           "attributes": [...],
            "children" : [...]
       }
       ],
       "message": "...",
       "exception": "...",
       "custom-data": "..."
   }
}
```

PUT response structure

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Date: Wed, 16 Feb 2011 07:11:49 GMT
```

POST response structure

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Date: Wed, 16 Feb 2011 07:24:06 GMT

{
    "response": {
        "message": null,
        "exception": null,
        "context": null,
        "custom-data": ""
    }
}
```

Sample Message Responses

Message 1

```
$ curl -K curls.cfg -d '{"args":" -l /clusters/Boston/exports/storage-views/AFTEST"}' -X POST
https://10.12.160.184/vplex/ls
{
    "response": {
       "message": "WARNING: The initiator target ports and unregistered initiators information
is from the last completed discovery on [Mon Aug 29 16:58:45 UTC 2011]. Please use the 'export
initiator-port discovery' command to manually initiate discovery.",
        "exception": null,
        "context": null,
      "custom-data": "\n/clusters/Boston/exports/storage-views/AFTEST: \nName
Value\n-------------------------
---------------------------------------------------\ncontroller-tag     -\ninitiators
[]\noperational-status        stopped\nport-name-enabled-status
[P0000000046E00709-A0-FC00,true,ok,\n
P0000000046E00709-A0-FC02,true,ok]\nports           [P0000000046E00709-A0-FC00,\n
P0000000046E00709-A0-FC02]\nvirtual-volumes           []\n\n"
    }
}
```

Message 2

```
$ curl -K curls.cfg -d '{"args":"-c Boston -v AFTEST AF_c2_r0_01_1_vol --force"}' -X POST
https://10.12.160.184/vplex/export+storage-view+removevirtualvolume
{
    "response": {
        "message": "INFO: Removed the following volumes: [AF_c2_r0_01_1_vol].",
        "exception": null,
        "context": null,
        "custom-data": "WARNING: The initiator target ports and unregistered initiators
information is\nfrom the last completed discovery on [Mon Aug 29 16:58:45 UTC 2011]. Please
use\nthe 'export initiator-port discovery' command to manually initiate discovery.\n\nRemoved
the following volumes: [AF_c2_r0_01_1_vol].\n\n"
    }
}
```

Sample Exception Responses

Exception 1

```
$ curl -K curls.cfg -d '{"args":"-c Boston -v AFTEST AF_c2_r0_01_1_vol --force"}' -X POST
https://10.12.160.184/vplex/export+storage-view+removevirtualvolume
{
    "response": {
        "message": null,
        "exception": "export storage-view removevirtualvolume:  No such option -c.\nSee 'help
export storage-view removevirtualvolume' or 'export storage-view\nremovevirtualvolume --help'
for information about required input formats.\n",
        "context": null,
        "custom-data": null
    }
}
```

Exception 2

```
$ curl -K curls.cfg -d '{"args":" -h"}' -X POST
https://10.241.164.103/vplex/export+storage-view+remfovevirtualvolume
{
    "response": {
        "message": null,
        "exception": "Need to include -f or --force option in the command arguments.",
        "context": null,
        "custom-data": null
    }
}
```

Exception 3

```
$ curl -K curls.cfg -d '{"args":" -h"}' -X POST
https://10.241.164.103/vplex/exports+storage-views+r+storage-views+removevirtualvolume
{
    "response": {
        "message": null,
       "exception": "Invalid URI in POST request. Not one among context,command or attribute.",
        "context": null,
        "custom-data": null
    }
}
```

Sample Custom Data (POST) Responses

Custom Data 1

```
$ curl -K curls.cfg -d '{"args":" -l /clusters/cluster-1/exports/storage-views"}' -X POST
https://10Custom.241.164.103/vplex/ls
{
    "response": {
        "message": null,
        "exception": null,
        "context": null,
      "custom-data": "\n/clusters/cluster-1/exports/storage-views: \nName         Operational
initiator-ports       virtual-volumes                                                      port
name, enabled, export status\n-------------  Status       ---------------------
--------------------------------------------------------------------------------
------------------------------------------\n-------------  -----------  ---------------------
--------------------------------------------------------------------------------
------------------------------------------\nVPLEX_Windows  ok           vplex-windows_3198e2,
(0,DEVICETEST_1_vol,VPD83T3:6000144000000010a00896d9579f6417,20G),
P000000003CA00896-A0-FC00,true,ok,\n                             vplex-windows_3198e3
(1,device_CLAR0145_0050_1_vol,VPD83T3:6000144000000010a00896d9579f6495,100M),
P000000003CA00896-A0-FC01,true,ok,\n
(2,device_CLAR0145_0051_1_vol,VPD83T3:6000144000000010a00896d9579f6496,100M),
P000000003CA00896-A1-FC00,true,ok,\n
(3,device_CLAR0145_0052_1_vol,VPD83T3:6000144000000010a00896d9579f6497,100M),
P000000003CA00896-A1-FC01,true,ok,\n
(4,device_CLAR0145_0053_1_vol,VPD83T3:6000144000000010a00896d9579f6498,100M),
P000000003CB00896-B0-FC00,true,ok,\n
(5,device_CLAR0145_0054_1_vol,VPD83T3:6000144000000010a00896d9579f6499,100M),
P000000003CB00896-B0-FC01,true,ok,\n
(6,testdev_vol,VPD83T3:6000144000000010a00896d9579f6454,20G),
P000000003CB00896-B1-FC00,true,ok,\n
(7,test_device_1_vol,VPD83T3:6000144000000010a00896d9579f63fc,10G)
P000000003CB00896-B1-FC01,true,ok\ntest    stopped
P000000003CA00896-A0-FC00,true,ok,\n
P000000003CA00896-A0-FC01,true,ok,\n
P000000003CA00896-A0-FC02,true,suspended,\n
P000000003CA00896-A0-FC03,true,suspended,\n
P000000003CA00896-A1-FC00,true,ok,\n
P000000003CA00896-A1-FC01,true,ok,\n
P000000003CA00896-A1-FC02,true,suspended,\n
P000000003CA00896-A1-FC03,true,suspended,\n
P000000003CB00896-B0-FC00,true,ok,\n
P000000003CB00896-B0-FC01,true,ok, ...\n
(16 total)\n\n"
    }
}
```

Custom Data 2

```
{
    "response": {
        "message": null,
        "exception": null,
        "context": null,
        "custom-data": "\n/engines/engine-2-1/directors/shinigami_68/hardware/ports/B0-FC00:
\nName     Value\n-------  -----\nenabled
true\n\n\n/engines/engine-1-1/directors/marta_35/hardware/ports/A3-FC01: \nName
Value\n-------  -----\nenabled
true\n\n\n/engines/engine-1-1/directors/marta_35/hardware/ports/A2-FC03: \nName
Value\n-------  -----\nenabled
true\n\n\n/engines/engine-1-1/directors/marta_36/hardware/ports/B1-FC02: \nName
Value\n-------  -----\nenabled
true\n\n\n/engines/engine-2-1/directors/shinigami_68/hardware/ports/B0-FC01: \nName
Value\n-------  -----\nenabled
true\n\n\n/engines/engine-1-1/directors/marta_35/hardware/ports/A3-FC02: \nName
Value\n-------  -----\nenabled
true\n\n\n/engines/engine-1-1/directors/marta_35/hardware/ports/A2-FC02: \nName
Value\n-------  -----\nenabled
true\n\n\n/engines/engine-1-1/directors/marta_36/hardware/ports/B1-FC01: \nName
Value\n-------  -----\nenabled
true\n\n\n/engines/engine-2-1/directors/shinigami_68/hardware/ports/B0-FC02: \nName
Value\n-------  -----\nenabled
true\n\n\n/engines/engine-2-1/directors/shinigami_68/hardware/ports/B0-FC03: \nName
Value\n-------  -----\nenabled
true\n\n\n/engines/engine-1-1/directors/marta_35/hardware/ports/A3-FC00: \nName
Value\n-------  -----\nenabled
true\n\n\n/engines/engine-1-1/directors/marta_36/hardware/ports/B1-FC03: \nName
Value\n-------  -----\nenabled
true\n\n\n/engines/engine-2-1/directors/shinigami_67/hardware/ports/A4-FC03: \nName
Value\n-------  -----\nenabled
true\n\n\n/engines/engine-1-1/directors/marta_36/hardware/ports/B2-FC02: \nName
Value\n-------  -----\nenabled
true\n\n\n/engines/engine-2-1/directors/shinigami_67/hardware/ports/A4-FC02: \nName
Value\n-------  -----\nenabled
true\n\n\n/engines/engine-1-1/directors/marta_36/hardware/ports/B2-FC03: \nName e
Value\n-------  -----\nenabled  true\n\n"
    }
}
```

Attribute Lists

Attributes are lists.

See http://www.json.org for complete syntax details.

Attribute List Examples

```
{
                        "value": [
                            "(Palazzo,{ summary:: ok, details:: [] })",
                            "(Venetian,{ summary:: ok, details:: [] })"
                        ],
                        "name": "operational-status"
                    },

                    {
                        "value": [
                            "Venetian_Palazzo_Distributed1",
                            "Venetian_Palazzo_Distributed2"
                        ],
                        "name": "virtual-volumes"
                    },

                        "value": [
                            "{3,Oracle_Demo_Nodes,device_Ora_data_1_vol,ok}",
                            "{6,Oracle_Demo_Nodes,device_Ora_data_4_vol,ok}",
                            "{0,Palazzo_ESX_212_View,Venetian_Palazzo_Distributed1,ok}",

                        "value": [
                            "0x5000144260061110/0x50000972c00d2998/0",
                            "0x5000144260061110/0x50000972c00d299c/0",
                                  "0x5000144260061112/0x50000972c00d2998/0",
                                  "0x5000144260061112/0x50000972c00d299c/0",

                             "value": [
                              "(Meditech_RP_Test_Vol_1,CDP_DEMO,VPLEX_Left,10.12.160.21,Production
Source)"
                             ],
                             "name": "recoverpoint-information"
```

Error responses

When the server cannot fulfill a request, it generates and returns an error. VPLEX Element Manager API returns a status code in the header of the returned error. For HTTP status codes that indicate failure, the response body includes an error message.

Failed response header

```
HTTP/1.1 #{statusCode}
Date: date
Content-Type: type
Content-Length: length
Connection: close
Server: server name
```

Failed response body

```
HTTP/1.1 516
Server: Apache-Coyote/1.1
Content-Type: application/json
Date: Wed, 03 Nov 2010 03:31:10 GMT

{
   "response": {
       "message": null,
       "exception": "Invalid argument(s) for help command.",
       "context": null,
       "custom-data": null
   }
}
```

Table 6 HTTP status codes returned for errors

HTTP status code	Error message
401	Authentication has failed. The client did not provide the correct username and password.

Table 6 HTTP status codes returned for errors

HTTP status code	Error message
404	Nonexistent URI. Context or command not found.
403	Forbidden for PUT request to change a read-only attribute.
5xx	Server internal error. The error message in the response body provides the details of the error. Specifically: 551 - 560: Tower Area 551 : No Such Component 552 : Extra Parameters Given 553 : No Healthy Directors 554 : No Such Command 555 : Already Connected To Tower 556 : Command Cancelled 541 - 550: SMS Business Layer 541-550:smsbusinesslayer 541 : Business Error 531-540:smstowercommunicationarea 531 : Sms To Tower Communication Error 532 : No Such Host 533 : No Route To Host 534 : Connection Refused 535 : No Running Firmware 536 : Firmware Busy 537 : System Error 521-530:sms-clilayer 521 : Command Error 522 : Command Shell Error 523 : Command Syntax Error 524 : Command Execution Error 511-520:WebapplicationLayer 511 : Restful System Error 512 : Command Execution Failed 513 : Invalid Input 514 : No Force Option Specified 515 : Invalid Json 516 : Command Not Supported 517 : Command Execution Timed Out

Supported VPLEX CLI commands

The following CLI commands can be executed using the VPLEX Element Manager API.

This is not a comprehensive list of supported commands on every system. To display a list of commands specifically supported by your system, see “Display supported commands” on page 44.

Note: Many of these commands require a **--force** option, particularly those that normally require a prompted response in the CLI.

Table 7 Supported VPLEX CLI commands (page 1 of 9)

VPLEX CLI command	Description
advadm dismantle	Dismantles VPLEX storage objects down to the storage-volume level, and optionally unclaims the storage volumes.
array claim	Claims and names unclaimed storage volumes for a given array
array re-discover	Re-discovers an array, and makes the array's storage volumes visible to the VPLEX.
array used-by	Displays the components that use a specified storage-array.
authentication directory-service configure	Configures a directory service on the VPLEX cluster to authenticate users against a remote directory server.
authentication directory-service map	Maps a directory server user or a user group to a VPLEX cluster.
authentication directory-service show	Displays configuration of the directory service used to authenticate users.
authentication directory-service unconfigure	Unconfigures a directory service on the VPLEX cluster.
authentication directory-service unmap	Unmaps the specified directory server user or group from the cluster.
batch-migrate cancel	Cancels an active migration and returns the source volumes to their state before the migration.
batch-migrate check-plan	Checks a batch migration plan.
batch-migrate clean	Cleans the specified batch migration and deletes the source devices.
batch-migrate commit	Commits the specified batch migration.
batch-migrate create-plan	Creates a batch migration plan file.
batch-migrate pause	Pauses the specified batch migration.
batch-migrate remove	Removes the record of the completed batch migration.
batch-migrate resume	Attempts to resume every migration in the specified batch.
batch-migrate start	Starts the specified batch migration.
batch-migrate summary	Displays a summary of the batch migration.
battery-conditioning disable	Disables battery conditioning on the specified backup battery unit(s).
battery-conditioning enable	Enables conditioning on the specified backup battery unit(s).
battery-conditioning manual- cycle cancel-request	Cancels a manually requested battery conditioning cycle on the specified backup battery unit.

Table 7 Supported VPLEX CLI commands (page 2 of 9)

VPLEX CLI command	Description
battery-conditioning manual- cycle request	Manually request a battery conditioning cycle on the specified backup battery unit.
battery-conditioning set-schedule	Set the battery conditioning schedule (day of week) for backup battery units on a cluster.
battery-conditioning summary	Displays a summary of the battery conditioning schedule for all devices, grouped by type and cluster.
cluster add	Adds a cluster to a running VPLEX.
cluster cacheflush	Flushes the cache on directors at the specified clusters to the back-end storage volumes.
cluster configdump	Dumps cluster configuration in an XML format, optionally directing it to a file.
cluster expel	Expels a cluster from its current island.
cluster forget	Tells VPLEX and VPLEX Management Console to forget the specified cluster.
cluster shutdown	Initiates the orderly shutdown of all directors at the specified cluster.
cluster status	Displays a cluster's operational status and health state.
cluster summary	Displays a summary of all clusters and the connectivity between them.
cluster unexpel	Allows a cluster to rejoin the VPLEX.
cluster-witness configure	Creates the cluster-witness context for enabling VPLEX Witness functionality and configuration commands.
cluster-witness disable	Disables Cluster Witness on both management servers and on Cluster Witness Server.
cluster-witness enable	Enables Cluster Witness and Cluster Witness Server on both clusters in a VPLEX Metro or Geo configuration.
collect-diagnostics	Collects logs, cores, and configuration information from the management server and directors.
configuration configure-auth-service	Configures the authentication service selected by the user for authenticating the users of VPLEX.
configuration connect-local-directors	Connects to the directors in the local cluster.
configuration connect-remote-directors	Connects the local cluster to directors in the remote cluster.
configuration enable-front-end-ports	After the meta-volume is created, continues the EZ-Setup wizard.
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.
configuration get-product-type	Displays the VPLEX product type (Local, Metro, or Geo).

Table 7 Supported VPLEX CLI commands (page 3 of 9)

VPLEX CLI command	Description
configuration join-clusters	Completes setup of VPLEX Metro or Geo using the EZ-Setup wizard.
configuration metadata backup	Configures and schedules the daily backup of VPLEX metadata.
configuration register-product	Registers the VPLEX product with EMC.
configuration show-meta-volume-candidates	Display the volumes which meet the criteria for a VPLEX meta-volume.
configuration sync-time	Synchronizes the time of the local management server with a remote management server.
connect	Connects to a director.
connectivity director	Displays connections from the specified director through data (non-management) ports.
connectivity show	Displays the communication endpoints that can see each other.
connectivity validate-be	Checks that the back-end connectivity is correctly configured.
connectivity validate-wan-com	Verifies the expected IP WAN COM connectivity.
connectivity window set	Sets values that control the operation of communications.
connectivity window show	Displays values that control the generic operation of communications for one or more directors.
connectivity window stat	Displays the current bandwidth statistics for all directors.
consistency-group add-virtual-volumes	Adds one or more virtual volumes to a consistency group.
consistency-group choose-winner	Selects a winning cluster during an inter-cluster link failure.
consistency-group create	Creates and names an empty consistency group.
consistency-group destroy	Destroys the specified empty consistency group(s).
consistency-group list-eligible-volumes	Displays the virtual volumes that are eligible to be added to a specified consistency group.
consistency-group remove-virtual-volumes	Removes one or more virtual volumes from the consistency group.
consistency-group resolve-conflicting-detach	Select a winning cluster on a consistency group on which there has been a conflicting detach.
consistency-group resume-after-data-loss-failure	Resumes I/O on an asynchronous consistency group when there are data loss failures.
consistency-group resume-after-rollback	Resume I/O to the volumes on the winning cluster in a consistency group after: • The losing cluster(s) have been detached, and • Data has been rolled back to the last point at which all clusters had a consistent view.

Table 7 Supported VPLEX CLI commands (page 4 of 9)

VPLEX CLI command	Description
consistency-group resume-at-loser	If I/O is suspended due to a data change, resumes I/O at the specified cluster and consistency group.
consistency-group set-detach-rule active-cluster-wins	Sets the detach-rule for one or more asynchronous consistency groups to “active-cluster-wins”
consistency-group set-detach-rule no-automatic-winner	Sets the detach-rule for one or more asynchronous consistency groups to “no-automatic-winner”.
consistency-group set-detach-rule winner	Sets the detach-rule for one or more synchronous consistency groups to “winner”.
consistency-group summary	Displays all the consistency groups with a state other than 'OK”.
device attach-mirror	Attaches a mirror as a RAID 1 child to another (parent) device, and starts a rebuild to synchronize the mirror.
device collapse	Collapses a one-legged device until a device with two or more children is reached.
device detach-mirror	Detaches an up-to-date child from the parent RAID 1 or removes the child from a parent RAID 1.
device resume-link-down	Resumes I/O for devices on the winning island during a link outage.
device resume-link-up	Resumes I/O on suspended top level devices, virtual volumes, or all virtual volumes in the system.
director appcon	Runs the application console on Linux systems.
director appdump	Downloads an application dump from one or more boards.
director appstatus	Displays the status of the application on one or more boards.
director commission	Starts the director’s participation in the cluster.
director decommission	Decommissions a director. The director stops participating in cluster activities.
director fc-port-stats	Displays/resets Fibre Channel port statistics for a specific director.
director forget	Removes a director from the VPLEX.
director ping	Displays the round-trip latency from a given director to the target machine, excluding any VPLEX overhead.
director tracepath	Displays the route taken by packets from a specified director to the target machine.
director uptime	Prints the uptime information for all connected directors.
disconnect	Disconnects one or more connected directors.
dm migration cancel	Cancels an existing data migration.
dm migration clean	Cleans a committed data migration.
dm migration commit	Commits a completed data migration allowing for its later removal.
dm migration pause	Pauses the specified in-progress or queued data migrations.

Table 7 Supported VPLEX CLI commands (page 5 of 9)

VPLEX CLI command	Description
dm migration remove	Removes the record of canceled or committed data migrations.
dm migration resume	Resumes a previously paused data migration.
dm migration start	Starts the specified migration.
drill-down	Displays the components of a view, virtual volume or device, down to the storage-volume context.
ds dd create	Creates a new distributed-device.
ds dd declare-winner	Declares a winning cluster for a distributed-device that is in conflict after a link outage.
ds dd destroy	Destroys the specified distributed-device(s).
ds dd remove-all-rules	Removes all rules from all distributed devices.
ds dd set-log	Allocates/unallocates segments of a logging volume to a distributed device or a component of a distributed device.
ds rule destroy	Destroys an existing rule.
ds rule island-containing	Adds a island-containing rule to an existing rule-set.
ds rule-set copy	Copy an existing rule-set.
ds rule-set create	Creates a new rule-set with the given name and encompassing clusters.
ds rule-set destroy	Destroys an existing rule-set.
ds rule-set what-if	Tests if/when I/O is resumed at individual clusters, according to the current rule-set.
ds summary	Display summary information about distributed-devices.
event-test	Verifies that the SMS can receive events from a director.
export initiator-port discovery	Discovers initiator ports on the front-end fabric.
export initiator-port register	Registers an initiator-port and associates one WWN pair with it
export initiator-port register-host	Creates a view, and registers each port WWN /name pair as an initiator port in that view.
export initiator-port unregister	Unregisters the specified initiator-port(s).
export port summary	Displays a summary of unhealthy exported ports.
export storage-view addinitiatorport	Adds the specified initiator port(s) to a storage view.
export storage-view addport	Adds the specified port(s) to the storage view.
export storage-view addvirtualvolume	Adds a virtual volume to a storage view.
export storage-view checkconfig	Checks the configuration of the views.

Table 7 Supported VPLEX CLI commands (page 6 of 9)

VPLEX CLI command	Description
export storage-view create	Creates a view with the given ports.
export storage-view destroy	Destroys the specified storage view.
export storage-view find	Displays export views for a specified volume, LUN, initiator, or cluster. Displays next available LUN number for all storage views.
export storage-view find-unmapped-volumes	Display unexported virtual volumes in the specified cluster.
export storage-view map	Displays only storage volumes with an I/O status other than 'Alive'.
export storage-view removeinitiatorport	Removes the specified initiator-port(s) from the view.
export storage-view removeport	Removes the specified port(s) from a storage view.
export storage-view removevirtualvolume	Removes the specified virtual volume from the view.
export storage-view show-powerpath-interfaces	Displays the mapping between PowerPath® interfaces and the VPLEX system ports.
export storage-view summary	Lists each view and the number of virtual volumes and initiators that it contains.
export target-port renamewwns	Renames a target port's WWN pair.
extent create	Creates one or more storage-volume extents.
extent destroy	Destroys one or more storage-volume extents.
extent summary	Displays a list of a cluster's unhealthy extents.
getsysinfo	Returns information about the current system.
health-check	Displays overall hardware/software health.
help	Displays help on one or more commands.
local-device create	Creates a new local-device.
local-device destroy	Destroys existing local-devices.
local-device summary	Displays unhealthy local devices and a summary of all local devices.
logical-unit forget	Forgets the specified logical units (LUNs).
logging-volume add-mirror	Adds a logging volume mirror.
logging-volume create	Creates a new logging volume in a cluster.
logging-volume destroy	Destroys an existing logging volume.
ls	Displays information about the current object or context.
management-server set-ip	Assigns IP address, net-mask, and gateway IP address to the management port connected to customer networks.
manifest version	Displays the version of the currently loaded manifest file.

Table 7 Supported VPLEX CLI commands (page 7 of 9)

VPLEX CLI command	Description
meta-volume attach-mirror	Attaches a storage-volume as a mirror to a meta-volume.
meta-volume backup	Creates a new meta-volume and writes the current in-memory system data to the new meta-volume without activating it.
meta-volume create	Creates a new meta-volume in a cluster when there is no existing active meta-volume.
meta-volume destroy	Destroys a meta-volume, and frees its storage volumes for other uses
meta-volume detach-mirror	Detaches a storage-volume/mirror from a meta-volume.
meta-volume move	Writes the current in-memory system data to the specified target meta-volume, then activates it.
meta-volume verify-on-disk-consistency	Analyzes a meta-volume's committed (on-disk) header slots for consistency across all mirrors/components.
ndu pre-config-upgrade	Disruptively upgrades a VPLEX Geo that has not been fully installed and configured.
ndu rolling-upgrade c4lx-to-sles	Performs a rolling upgrade of each director’s operating system from C4LX to SLES11.
ndu rolling-upgrade ssd-fw	Starts a rolling upgrade of SSD firmware on the directors.
notifications call-home import-event-modifications	Imports and applies a modified call-home notification file.
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.
notifications call-home test	Passes a test event through the call-home.
notifications snmp-trap create	Creates an SNMP trap sink for call-home events.
notifications snmp-trap destroy	Destroys one or more SNMP traps.
rebuild set-transfer-size	Changes the transfer-size of the given devices.
rebuild show-transfer-size	Shows the transfer-size of specified RAID 1 devices.
rebuild status	Displays all global and cluster-local rebuilds along with their completion status.
remote-clusters add-addresses	Adds one or more address:subnet mask configurations for the specified remote-cluster entry for this cluster.
remote-clusters clear-addresses	Clear one, several or all address:subnet maskpairs for the specified remote-cluster entry for this cluster.
report aggregate-monitors	Aggregate the reports generated by the report create-monitors or monitor commands.
report capacity-arrays	Generates a capacity report.

Table 7 Supported VPLEX CLI commands (page 8 of 9)

VPLEX CLI command	Description
report capacity-clusters	Generates a capacity report for every cluster.
report capacity-hosts	Generates a host capacity report.
report create-monitors	Creates three performance monitors for each director in the VPLEX: storage-volume performance, port performance, and virtual volume performance. Each monitor has one file sink.
report poll-monitors	Polls the report monitors created by the report create-monitors command.
rp rpa-cluster remove	Removes information about a RecoverPoint Appliance from VPLEX.
rp summary	Displays a summary of replication information for each RecoverPoint Appliance.
rp validate-configuration	Validates the RecoverPoint splitter configuration.
schedule list	Lists all scheduled jobs.
security delete-ca-certificate	Deletes the specified CA certificate and its key.
security delete-host-certificate	Deletes the specified host certificate.
security remove-login-banner	Removes the login banner from the management server.
security renew-all-certificates	Renews CA and host security certificates.
security set-login-banner	Applies a text file as the login banner on the management server.
sessions	Displays active VPLEX Management Console sessions.
set	Changes the value of a writable attribute(s) in the given context.
show-use-hierarchy	Display the usage hierarchy for a storage element down to the storage-array.
sms dump	Collects the logs files on the management server.
snmp-agent configure	Configures the SNMP agent service on the local cluster.
snmp-agent start	Starts the SNMP agent service.
snmp-agent status	Displays the SNMP agent service on the local cluster.
snmp-agent stop	Stops the SNMP agent service.
snmp-agent unconfigure	Destroys the SNMP agent.
storage-volume auto-unbanish-interval	Displays or changes auto-unbanish interval on a single director.
storage-volume claim	Claims the specified storage volumes.
storage-volume find-array	Searches storage arrays for the specified storage-volume(s).
storage-volume forget	Tells the cluster that a storage-volume or a set of storage volumes are physically removed.

Table 7 Supported VPLEX CLI commands (page 9 of 9)

VPLEX CLI command	Description
storage-volume list-banished	Displays banished storage-volumes on a director.
storage-volume resurrect	Resurrect the specified storage-volume(s).
storage-volume summary	Displays a list of a cluster's storage volumes.
storage-volume unclaim	Unclaims the specified previously claimed storage volumes.
storage-volume unbanish	Unbanishes a storage-volume on one or more directors.
storage-volume used-by	Displays the components that use the specified storage volumes.
subnet clear	Clears one or more attributes of an existing subnet configuration.
subnet create	Creates a new subnet.
subnet destroy	Destroys a subnet configuration.
subnet modify	Modifies an existing subnet configuration.
user reset	Allows an Administrator user to reset the password for any username.
validate-system-configuration	Performs a basic system configuration check.
vault go	Initiates a manual vault of all dirty pages in the cache for every director in the cluster
vault overrideUnvaultQuorum	Allows the cluster to proceed with the recovery of the vault(s) without all the required director(s).
vault status	Displays the current cache vault/unvault status of the cluster.
version	Display version information for connected directors.
virtual-volume cache-invalidate	Invalidates the cache of the specified virtual volume.
virtual-volume create	Creates a virtual volume with synchronous cache mode on a host device.
virtual-volume destroy	Destroys existing virtual volumes.
virtual-volume expand	Non-disruptively increases the capacity of an existing virtual volume.
virtual-volume summary	Displays a summary of all virtual volumes.

Display supported commands

Use the POST command to return an up-to-date list of all VPLEX CLI commands that can be executed using VPLEX Element Manager API.

The POST command also supports the **Help** command for a specific command as described in “Getting help for a supported command” on page 45.

Header request

Example:

```
POST https://mgmt-server-ip/vplex/help
HTTP/1.1
Username: service
Password: password
Content-Length: 0
```

Response

```
The following example is a partial list of the output.
{
    "response": {
        "message": null,
        "exception": null,
        "context": null,
        "custom-data": "[advadm dismantle, array claim, array
   re-discover, authentication directory-service configure,
   authentication directory-service map, authentication
   directory-service show, authentication directory-service
   unconfigure, authentication directory-service unmap, batch-migrate
   cancel, batch-migrate check-plan, batch-migrate clean,
   batch-migrate commit, batch-migrate create-plan, batch-migrate
   pause, batch-migrate remove, batch-migrate resume, batch-migrate
   start, batch-migrate summary, cache clear all, cache clear
   olderthan, cache dump, cache refresh olderthan, cache stats, cache
   style, cache use existing, cache use newerthan, cache use off,
   cluster add, cluster cacheflush, cluster configdump, cluster expel,
   cluster forget, cluster shutdown, cluster status, cluster
:
:
:
:
"
    }
}
```

Process Help Output

The following example uses **awk, curl,** and the VPLEX CLI **help** command (via REST POST request).

```
curl -k -H "Username:service" -H "Password:service" -s  -g -d
   '{"args":""}' -X POST https://<ipaddress>/vplex/help
|awk -F, '/custom-data/ {for (x=1;x <= NF; x++) {printf "%s\n",$x}}'
   |sed -e 's/\].*$//' -e 's/^.*\[/ /'
```

Getting help for a supported command

You can issue the **Help** command for any supported command by using the **args** parameter in the request as described below.

Note: The **Help** command issued from VPLEX Element Manager API provides a subset of the information displayed when issued from the VPLEX CLI.

Header request

```
POST https://mgmt-server-ip/vplex/help
HTTP/1.1
Username: service
Password: password
Content-Length: 30

{ "args" : "cluster status" }
```

Response

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Transfer-Encoding: chunked
Date: Thu, 17 Feb 2011 05:42:48 GMT

{
    "response": {
        "message": null,
        "exception": null,
        "context": null,
        "custom-data": "synopsis: status [<options>]\r\n\r\nDisplays a
   cluster's operational-status and health-state.\r\n\r\noptions (* =
   required):\r\n  -h, --help\r\n          Displays the usage for this
   command.\r\n  --verbose\r\n          Provide more output during
   command execution.  This may not have any effect for some
   commands.\r\n  -c, --clusters= <clusters>\r\n          clusters
   whose operational-status to display.\r\n\r\nAlong with the
   operational-status, an indication of why it could be non-nominal and
   a progress indicator are displayed.\r\nHealth-state has a similar
   indicator.\r\n\r\n"
}
```

Cluster-contexts RESTful command

The **cluster-contexts** command returns a comma separated list of all the clusters contexts for each VPLEX SMS cluster. The result of this command is available in the custom data of the JSON response.

Note: The use of the cluster-contexts command is supported only through the RESTful API.

Command Syntax

The request type is GET.

```
GET https://IP_address_of_SMS/vplex/cluster-contexts
```

Header

```
HTTP/1.1
Username:service
Password:password
Content-Length: 0
```

Response

The following is an example of the output.

```
"response": {
       "context": null,
       "message": "cluster-contexts Command - review specified context
   paths for clusters",
       "exception": null,
       "custom-data":
   "[[/clusters/cluster-1/cluster-connectivity,/clusters/cluster-1
/consistency-groups,/clusters/cluster-1/devices,/clusters/cluster-1/
exports,/clusters/cluster-1/exports/initiator-ports,/clusters/cluster-
   1/exports/ports,/clusters/cluster-1/exports/storage-views,/clusters
   /cluster-1/storage-elements,/clusters/cluster-1/storage-elements/
extents,/clusters/cluster-1/storage-elements/storage-arrays,/clusters/
   cluster-1/storage-elements/storage-volumes,/clusters/cluster-1/
system-volumes,/clusters/cluster-1/virtual-volumes,/clusters/
cluster-1/uninterruptible-power-supplies],[/clusters/shanghai/
cluster-connectivity,/clusters/shanghai/consistency-groups,/clusters/s
   hanghai/devices,/clusters/shanghai/exports,/clusters/shanghai/
exports/initiator-ports,/clusters/shanghai/exports/ports,/clusters/
shanghai/exports/storage-views,/clusters/shanghai/storage-elements,/cl
   usters/shanghai/storage-elements/extents,/clusters/shanghai/
storage-elements/storage-arrays,/clusters/shanghai/storage-elements/
storage-volumes,/clusters/shanghai/system-volumes,/clusters/shanghai/v
   irtual-volumes,/clusters/shanghai/uninterruptible-power-supplies]]"
   }
```

CHAPTER 2
API commands

This chapter describes VPLEX Element Manager API commands:

GET

Performs an ls of a context and shows the context and attributes, and sub-contexts.

URI rules

VPLEX URI (root)

```
https://mgmt-server-ip/vplex/
```

Returns the attributes and child contexts available at the root context.

VPLEX context

```
URI root/context
```

Example:

```
https://mgmt-server-ip/vplex/clusters/cluster-1/
```

Returns the attributes and child contexts available at the specified cluster context.

Attribute

```
URI root/context?attribute
```

Example:

```
https://mgmt-server-ip/vplex/clusters/cluster-1?cluster-id
```

If an attribute does not exist, returns HTTP error 404.

Header request

Example:

```
GET https://mgmt-server-ip/vplex/clusters/cluster-name
HTTP/1.1
Username: service
Password: password
```

Response

Example:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Date: Wed, 16 Feb 2011 07:24:06 GMT
{

    "response": {
        "message": null,
        "exception": null,
        "context": {
          "name": "cluster-1",
          "children": [
                {
                    "type": "cluster-connectivity",
                    "name": "cluster-connectivity"
                },
                {
                    "type": "devices",
                    "name": "devices"
                },
                {
                    "type": "exports",
                    "name": "exports"
                },
                {
                    "type": "storage-elements",
                    "name": "storage-elements"
                },
                {
                    "type": "system-volumes",
                    "name": "system-volumes"
                },
                {
                    "type": "uninterruptible-power-supplies",
                    "name": "uninterruptible-power-supplies"
                },
                {
                    "type": "virtual-volumes",
                    "name": "virtual-volumes"
                },
                {
                    "type": "volume-sets",
                    "name": "volume-sets"
                }
            ],
            "parent": "/clusters",
            "attributes": [
                {
                    "value": "true",
                    "name": "allow-auto-join"
                },
                {
                    "value": "0",
                    "name": "auto-expel-count"
                },
                {
                    "value": "0",
                    "name": "auto-expel-period"
                },
                {
                    "value": "0",
                    "name": "auto-join-delay"
                },
                {
                    "value": "1",
                    "name": "cluster-id"
                },
                {
                    "value": "true",
                    "name": "connected"
                },
                {
                    "value": "synchronous",
                    "name": "default-cache-mode"
                },
                {
                    "value": "director-1-1-B",
                    "name": "default-director"
                },
                {
                    "value": "[director-1-1-B, director-1-1-A]",
                    "name": "director-names"
                },
                {
                    "value": "[engine-1-1 : director-1-1-A :
   stand-by-power-supply-A : The SPS is in a faulted state. ]"
                    "name": "health-indications"
                },
                {
                    "value": "major-failure",
                    "name": "health-state"
                },
                {
                    "value": "1",
                    "name": "island-id"
                },
                {
                    "value": "cluster-1",
                    "name": "name"
                },
                {
                    "value": "ok",
                    "name": "operational-status"
                },
                {
                    "value": "[]",
                    "name": "transition-indications"
                },
                {
                    "value": "[]",
                    "name": "transition-progress"
                }
            ],
            "type": "cluster"
        },
        "custom-data": null
    }
}
```

PUT

Sets a value for a writeable attribute. Performs the same function as the VPLEX CLI **set** command with the exception that it does not accept options (for example: **set --force**).

Note: To use options, use **set** command using POST method.

URI rules

Base URI (root)

```
https://mgmt-server-ip/vplex/
```

VPLEX context

```
URI root/context?attribute=value
```

Example:

```
https://mgmt-server-ip/vplex/clusters/cluster-1?
   cluster-attribute=value
```

Attribute

```
https://mgmt-server-ip/vplex/clusters/cluster-1?
   allow-auto-join=new-value
```

Updates the attribute with the new value. Note that the value can be a number or text string; no spaces. If the attribute is read-only, returns HTTP error 403.

Header request

Example:

```
PUT https://mgmt-server-ip/vplex/clusters/cluster-
   id?allow-auto-join=true
HTTP/1.1
Username: service
Password: password
```

Response

Example:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Transfer-Encoding: chunked
Date: Wed, 16 Feb 2011 07:15:42 GMT
```

POST

Executes a VPLEX CLI command.

URI rules

Base URI (root)

```
https://mgmt-server-ip/vplex/xyz
```

Where “*xyz*” is a VPLEX CLI command. Note that commands with spaces use “+” instead of spaces.

⚠WARNING

If the CLI command includes --force as an optional argument, the --force argument MUST be included when the command is used in VPLEX Element Manager API.

Most instances of the **--force** argument allow commands that would otherwise require confirmation to be run without confirmation. Since VPLEX Element Manager API is non-interactive, make sure to include **--force** if it is an available argument for a CLI command.

If a command takes longer than 60 seconds to complete, the server returns a HTTP error 202. The return includes a Location header. The Location header provides a URI that can be used to determine the status of the command. The location information is part of the response header and not part of the response structure itself.

The command then completes asynchronously.

See “commandresult” on page 55 and “downloadfile” on page 55 for more information.

VPLEX context

Not applicable.

Attribute

```
https://mgmt-server-ip/vplex/extent+create
```

Note: Command arguments are placed in the request body of the POST method.

Header request

Example:

```
POST https://mgmt-server-ip/vplex/extent+create
HTTP/1.1
Username: service
Password: password

{ "args" : "-d test -s 1G"}
```

Example for asynchronous commands:

```
POST https://mgmt-server-ip/vplex/sms+dump
HTTP/1.1
Username: service
Password: password
Content-Length: 34

{"args" : "-d /var/log/VPlex/cli/wwwroot/outgoing"}
```

Request body

The command arguments is in JSON format with the following pattern:

{*"args":user_supplied command_arguments*}

where “args” is a key word understood by the VPLEX Element manager server.

Depending on your programming language, scripting language, or tool you may have to escape the Request Body appropriately. For example, when using Python the string for the Request body is:

"{\"*args\" : \"user_supplied command arguments\"*}"

Note: The Request body of the POST method can be empty if the command takes no arguments.

Response

Example:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Transfer-Encoding: chunked
Date: Wed, 16 Feb 2011 07:24:06 GMT

{
    "response": {
        "message": null,
        "exception": null,
        "context": null,
        "custom-data": ""
    }
}
```

Example for asynchronous commands:

```
HTTP/1.1 202 Accepted
Server: Apache-Coyote/1.1
Location: https://mgmt-server-ip/vplex/commandresult/1326192320937
Content-Type: text/json
Date: Tue, 10 Jan 2012 10:45:20 GMT
Content-Length: 271

{
    "response": {
        "message": "Command execution taking longer than 60 seconds.
   Command will be executed Asynchronously. Check command status at URL
   specified in Location",
        "exception": null,
        "context": null,
        "custom-data": null
    }
}
```

Note: Use the `Location` URL to get asynchronous response text at a later time.

IMPORTANT

After running an asynchronous command such as listed above, you then need to issue a GET on the location header URI to obtain the asynchronous output. If the Asynchronous command is slow in response, the client may retry using the URI at the Location header several times.

commandresult

For VPLEX commands that take more than 60 seconds to execute, returns the status of the command.

URI rules

BaseURI(root)

```
https://mgmt-server-ip/vplex/commandresult/xyz
```

Where “xyz” is a command result ID obtained by a previous asynchronous command invocation.

Note: Asynchronous command results are only cached for 24 hours. If the above request is issued after 24 hours have passed, the command returns HTTP error 404.

Header request

Example:

```
https://mgmt-server-ip/vplex/commandresult/1323445837220
GET HTTP/1.1
Username: service
Password: password
```

Response

Example:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Transfer-Encoding: chunked
Date: Wed, 16 Feb 2011 07:24:06 GMT
```

If the command is still in progress, the response is as follows:

```
HTTP/1.1 517
Server: Apache-Coyote/1.1
Content-Type: application/json
Transfer-Encoding: chunked
Date: Wed, 04 Jan 2012 12:01:03 GMT

{
    "response": {
        "message": "Command execution continues to be under progress",
        "exception": null,
        "context": null,
        "custom-data": null
    }
}
```

downloadfile

Downloads files from the management server over HTTP.

Some VPLEX CLI commands that produce output as a file can be instructed to place the file in a folder designated by the user.

This command downloads only those output files that can be directed to the folder:

/var/log/VPlex/cli/wwwroot/outgoing

VPLEX CLI commands whose output can be directed to /var/log/VPlex/cli/wwwroot/outgoing include:

- cluster configdump
- collect-diagnostics
- connectivity director
- getsysinfo
- report aggregate-monitors
- report capacity-arrays
- report capacity-clusters
- report capacity-hosts
- report create-monitors
- sms dump
- syrcollect

For example, to direct the output of the **sms dump** command to the folder:

```
POST https://mgmt-server-ip/vplex/sms+dump
HTTP/1.1
Username: service
Password: password
Content-Length: 34
```

{"args" : "-d /var/log/VPlex/cli/wwwroot/outgoing"}

Note: For commands that produce a file as an output, client scripts should specify only paths to which they have write access or the command will fail. Additionally, if client scripts attempt to download the file (using the downloaded file URI) after the command has completed, you must use the path /var/log/VPlex/cli/wwwroot/outgoing.

URI rules

BASE URI:

```
https://mgmt-server-ip/vplex/downloadfile/file-name
```

Example:

```
https://mgmt-server-ip/vplex/downloadfile/smsDump.zip
```

The specified file is expected to be in: /var/log/VPlex/cli/wwwroot/outgoing

If the file does not exist, this command returns HTTP error 404.

Up to 50 simultaneous download operations are supported. If the 50 simultaneous download limit is reached, the VPLEX Element Manager API will return an HTTP error code 503 with error message:

```
Server Too Busy, Retry after 2 mins.
```

Header request

```
Example
https://mgmt-server-ip/vplex/downloadfile/smsDump.zip
HTTP/1.1
Username: service
Password: password
```

Response

Example:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Disposition: attachment; filename=smsDump.zip
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Content-Type: application/zip
Content-Length: 46494483
Date: Wed, 04 Jan 2012 14:28:22 GMT
<the file contents.>
```

CHAPTER 3
Developer Guidelines

This chapter includes information about sample scripts on the VPLEX Management Server.

Locate the scripts on the server at **/opt/emc/vplex/clientscripts.zip**.

Guidelines

Deep recursive HTTP GET calls

Avoid deep recursive HTTP GET calls. Performing deep recursive HTTP GET calls (for example, https://ip:port/vplex/**) results in longer response time. This is because the **cluster contexts** command receives all the objects under the VPLEX context. The VPLEX Element Manager API clients must always perform single context level queries. To retrieve the virtual-volume context structure of a VPLEX cluster:

1. Use the new **cluster contexts** command to get the available contexts.

   ```
   GET  https://IP_address_of_SMS/vplex/cluster-contexts
   ```

2. Find the context for virtual-volumes from the output of the **cluster-contexts** command, and build a query.

   ```
   GET
   https://IP_address_of_SMS/vplex/clusters/<clusterName>/virtual-volu
   mes
   ```

For example:

```
GET
https://IP_address_of_SMS/vplex/clusters/NewYork/virtual-volumes
```

Note: You can execute this query for each cluster.

Frequent HTTP GET calls

Avoid frequent HTTP GET calls. Every HTTP GET call reconstructs the managed object attributes, placing a load on the management server, which can slow down responses to other users of the management server. Construct requests in a manner to gather only the necessary data. "Sample Scripts" on page 60 provides more information. It is recommended to retrieve large amounts of data only as frequent as every 20 minutes. When an application polls to fetch configuration information periodically, it does not have to poll the entire data all the time. The **health-check** command can be used to obtain the overall health and the configuration summary of the system. If there is a change in configuration, poll the entire data. The polling interval to check the health of the system must be around 20 minutes, and applications can take around 60 minutes to get the entire system information. Longer polling intervals help determine changes in the object properties.

Load balancing management servers

In VPLEX Metro or Geo systems, there are two management servers that can be used for managing either of the VPLEX clusters. In order to achieve load balancing between the RESTful applications and the daily management activities, use one management server for RESTful API applications, and the other for normal VPLEX management.

File download

The number of simultaneous file download operations that are allowed with the current RESTful API is 50. However, to achieve optimum performance on download operations without impacting the response time for other users of the management server, the number of download operations must be restricted to 5.

New output format

Starting in Release 5.2, VPLEX RESTful API supports a new output format. The new format allows disabling pretty print, which is 80% more efficient than the older format. However, the old format is still available for backward compatibility. The new format requires less memory, CPU, and network bandwidth. This format also makes client side scripting more efficient because there is less data to deal with. The new JSON format is also more compatible with the JSON Standard.

To enable this format, add the following HTTP header to your request:

```
Accept: application/json;format=1;prettyprint=1
```

Pool listing

Do not use the RESTful interface to poll the pools on a managed array. This is a very expensive operation. It is recommended that you limit the number of times you query pools.

Sample Scripts

This section includes information on the sample perl and python scripts.

Perl scripts

When you extract **clientscripts.zip**, the following directory structure is created for Perl scripts in the /opt/emc/vplex directory:

- clientscripts/
- clientscripts/perl5/

The /perl5/ folder contains 4 files:

- **HttpApi.pm** - A module that handles all the HTTP interactions.
- **VPLEXApi.pm** - The main module that demonstrates how to consume RESTful API.
- **main.pl** - A loader script that does command line parsing and hands over the calls to VPlexApi.
- **Util.pm** - A module that provides utility functions mostly related to config management.

Prerequisites

The Perl scripts require:

- VPLEX version 5.0 or higher
- Perl JSON module
- Perl CURL Module/Perl LWP Module.

  Note: Perl 5 for MSWin32-x86-multi-thread requires either Crypt::SSLeay or IO::Socket::SSL.

- The storage volume passed as argument has at least 5 MB of free space.

Python scripts

When you extract **clientscripts.zip,** the following directory structure is created for Python scripts:

clientscripts/

clientscripts/python2.6/

The /pyhton26/ dfolder contains the following files:

- **httpapi.py** - A file containing the code to do the basic RESTful API operations of GET and POST.
- **vplexapi.py** - The main module that demonstrates how to consume RESTful API.
- **main.py** - A loader script that does command line parsing and hands over the calls to VPlexApi.
- **util.py** - A module that provides utility functions mostly related to config management.
- **vplexapi.py** - A file containing the API methods that perform the RESTful API calls.
- **monitoring.py** - This module demonstrates how to monitor the health of the VPLEX system.
- **MappingInfo.py** - This module demonstrates how to build a virtual volume to storage volume mapping for both local and distributed volumes.
- **settings.cfg** - The configuration for setting up the VPLEX instance for RESTful API communication.

Prerequisites

The Python scripts require:

- VPLEX version 5.3 or higher
- Python versions 2.6 and below: simplejson module
- Python version 2.6 and above: the inbuilt json module
- PyCurl module (Python Curl Module)
- The storage volume passed as argument has at least 5 MB of free space.

INDEX