NetApp SolidFire Plug-in for vRealize Orchestrator User Guide

Version 1.4

August 2017
doccomments@netapp.com

NetApp®

Contents

Introduction

The NetApp SolidFire Plug-in for vRealize Orchestrator enables you to further integrate SolidFire storage into your VMware environment by enabling you to administer your SolidFire storage system with VMware vRealize Orchestrator.

Most SolidFire API methods are modeled as actions in the plug-in, enabling you to combine the inputs and outputs of these actions into workflows to schedule and automate complex storage administration tasks.

VMware administrators can use the plug-in to reduce the need for specialized storage administration, and instead focus on making storage transparent to daily operations by administering the storage system as part of the VMware environment. You can integrate the storage system into new or existing workflows, and automate these workflows to greatly simplify your environment. The plug-in packages actions into built-in workflows that enable the VMware administrator to build more complex workflows.

Requirements

The NetApp SolidFire Plug-in for vRealize Orchestrator has the following installation and usage requirements.

Installation requirements

The plug-in supports the following software and hardware environment:

- SolidFire Element OS 8.x and 9.x
- VMware vRealize Orchestrator 6.0.1 through 6.0.5, 7.x
- VMware ESXi 6.0 and 6.5
- Host CPU: 2GHz or faster
- Host memory: 8GB or higher

Usage requirements

The plug-in has the following general usage requirements:

- Each workflow in the **NetApp SolidFire > Element** directory requires a SolidFire:SolidFireConnection object as an input parameter.
- Some actions require SolidFire types as inputs. You must construct these types before passing them to the action by either getting the object using a "Get" workflow or by using a scripting task.

Intended audience

This document addresses installation and use of the plug-in, and makes the following assumptions:

- You are familiar with storage virtualization concepts.
- You are familiar with vRealize Orchestrator concepts, such as actions, workflows, and scripts, and how to create and use them in the VMware user interface.
- You are familiar with basic concepts of SolidFire storage, such as volumes and snapshots.

- You have access to information about the network environment the SolidFire storage system resides in.

Installing the plug-in

To install the NetApp SolidFire Plug-in for vRealize Orchestrator, you need to install plug-in .vmoapp file on the server as well as import the plug-in package using the Orchestrator client.

Steps

1. Installing the plug-in .vmoapp file on page 11
2. Importing the plug-in package on page 11

Installing the plug-in .vmoapp file

You can install the plug-in .vmoapp file from the Control Center page of the Orchestrator server. Installing the file requires a restart of the Orchestrator service.

Before you begin

- Ensure the Orchestrator server is not in use by others before you begin, so that their work is not disrupted.
- You must have administrator access to the Orchestrator server.

Steps

1. Download and save the latest version of the plug-in .vmoapp file from *GitHub*.
2. Browse to the IP address or DNS name of the Orchestrator server instance.
3. When prompted, log in with the administrator credentials.
4. On the **Control Center** home page, click **Manage Plug-Ins** in the **Plug-Ins** section.
5. Under the **Install plug-in** heading, click **Browse...**.
6. Choose the plug-in file that you downloaded, and click **Open**.
7. Read the End User License Agreement in the **EULA** text box, and enable the **Accept EULA** box.
8. Click **Install**.
9. Navigate to the **Control Center** home page.
10. In the **Manage** section, click **Startup Options**.
11. Click **Restart** to restart the Orchestrator server service.

Importing the plug-in package

You need to import the plug-in package to enable functionality of workflows and other plug-in elements.

Before you begin

Back up any custom Orchestrator elements.

Steps

1. Download and save the latest version of the plug-in .package file from *GitHub*.
2. Open the Orchestrator client.
3. Select **Administer** from the drop-down list.
4. Click on the **Packages** tab.
5. Right-click in an empty area of the packages pane and select **Import package**.
6. Browse to the NetApp SolidFire .package file you saved, and click **Open**.
7. Select **Import and trust provider**.
8. In the drop-down list, choose **Import tags and overwrite existing values**.
9. Click **Import selected elements**.

   The plug-in is ready to use.

Uninstalling the plug-in

You can remove the plug-in if it is no longer needed. You do not need to remove the plug-in if you are upgrading to a newer version; in this case, the older version of the plug-in is automatically replaced.

For plug-in uninstallation instructions, see the following VMware Knowledge Base article: *https://kb.vmware.com/kb/2064575*.

Configuration

You need to configure the connection to your storage system and VMware vCenter before you can use the NetApp SolidFire Plug-in for vRealize Orchestrator.

Creating a connection

The NetApp SolidFire vRealize Orchestrator plug-in makes use of discrete connections to each of your clusters or nodes. When you create a connection, the system stores it for future use until the Orchestrator service is restarted.

Steps

1. In the **Run** interface, select the **Workflows** tab.
2. Navigate to the **NetApp SolidFire** folder and expand it.
3. Expand the **Connection** folder.
4. Right-click `CreateConnection`, and select **Start workflow...**.
5. Enter an IP address or FQDN of the cluster or node in the **Target** field.
6. Enter a cluster administrator username in the **Username** field.
7. Enter the password for the cluster administrator in the **Password** field.
8. Optional: To restrict the connection to a certain API version, enter a version in the **Version** field.
9. Optional: To connect to a specific node, enter

   `442`

   in the **Port** field. Otherwise, leave this field blank to connect to a cluster.
10. Optional: Enable the **Verify SSL** radio button to verify the security certificate installed on the storage system. This option is only valid if you connect using a FQDN.
11. Optional: Enter a value in the **Timeout MSecs** field to control how long the system tries to connect before failing.
12. Click **Submit**.

Adding a vCenter instance to vRealize Orchestrator

You can add a vCenter instance to Orchestrator to enable workflow activities from some objects in vCenter. You can do this from the Orchestrator interface using a built-in vCenter workflow.

For more information, see the VMware documentation for this procedure at *http://pubs.vmware.com/orchestrator-72/topic/com.vmware.vrealize.orchestrator-use-plugins.doc/GUID-A7E5DD99-13CF-4E31-87D7-5741380D8764.html*.

Running workflows

You can run workflows manually in vRealize Orchestrator from the **Workflows** view in the **Run** perspective. In the workflow library, workflows are organized in to folders based on what entity or part of the system they operate on.

About this task

Many workflows require a certain amount of input to run, such as an IP address, volume name, or snapshot ID. All workflows require a connection instance to run.

Steps

1. Go to the **Run** perspective and select the **Workflows** view.
2. Expand the **Library > NetApp SolidFire** folder.
3. Expand the desired group of workflows.
4. Right-click a workflow and select **Start workflow...**.
5. Enter the required information into the input fields.
6. Click **Submit**.

   The workflow runs, and you can follow the progress in the **Schema** tab. If execution fails, you can see the error messages in the **Logs** tab.

Creating a Simple Workflow

You can use the vRealize Orchestrator interface to create new workflows if the built-in workflows do not suffice. Workflows usually consist of several actions or simple workflows connected in a way that completes a complex set of tasks.

Steps

1. In the VMware vRealize Orchestrator interface, create a folder for your custom workflow.
2. Right click on the folder you created, and select **New workflow**.
3. Name the new workflow and click **Ok**.
4. In the **New workflow** dialog, select the **Schema** tab.
5. In the schema list, open the **All Actions** section.
6. Expand the folder of the group of actions you need.
7. Select the desired action, and drag it on to the graphical workflow.

   The **Setup** button appears in the graphical workflow area.
8. Click the **Setup** button.
9. Optional: In the **Promote Workflow Input/Output Parameters** window, rename any inputs or outputs as needed.
10. Click **Promote**.
11. Select another action, and drag it on to the graphical workflow after the first action.
12. Click the **Setup** button.

    vRealize Orchestrator automatically maps outputs to inputs between actions.
13. If the default input/output mapping is correct, click **Promote**.
14. Optional: Repeat the steps to select, configure, and promote more actions as needed.
15. Optional: If necessary, add scriptable task elements to the workflow to change the behavior of individual workflow elements. For more information, see the *VMware vRealize Orchestrator documentation*.

Action reference

This section describes all actions included with the plug-in.

Connection actions

Connection actions enable you to create, list, and remove connections to the storage system.

createSolidFireConnection

The createSolidFireConnection action enables you to create a connection to a SolidFire cluster. Error messages received during the execution of this action contain as much helpful information as possible to help you troubleshoot any connection issues.

Input parameters

This action has the following input parameters:

Input	Type	Description
target	string	The IP address or FQDN of the SolidFire cluster to connect to.
username	string	The username of the SolidFire cluster user account.
password	SecureString	The password for the SolidFire cluster user account.
version	string	The version of the SolidFire cluster API to use.
port	number	The port to use for the connection. Use this parameter to choose between a cluster connection and a node connection. If you omit this parameter, the default port of 443 is used. Possible values: • 443: Connect to a cluster (default). • 442: Connect to a node.
timeoutMSecs	number	The amount of time to wait for the connection to succeed, in milliseconds.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireConnection	Information about the new connection to the storage system.

getLastSolidFireConnection

The getLastSolidFireConnection action enables you to get the last-created connection to a storage system.

Input parameters

This action has no input parameters.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireConnection	Information about the last-created connection to a storage system.

listSolidFireConnections

The listSolidFireConnections action enables you to list the existing connections to SolidFire clusters and nodes.

Input parameters

This action has no input parameters.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireConnection	Information about the existing connections to the storage systems.

removeSolidFireConnection

The removeSolidFireConnection action enables you to remove an existing connection to a SolidFire storage system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The connection to remove.

Output parameter

This action has no output parameters.

Account actions

Account actions enable you to add, remove, view, and modify account and security information.

addAccount

The addAccount action enables you to add an account to a system. Once the account is added, you can create volumes under the new account. The CHAP settings you specify for the account apply to all volumes owned by the account.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
username	string	Unique username for this account. May be 1 to 64 characters in length.
initiatorSecret	string	The CHAP secret to use for the initiator. This secret must be 12-16 characters in length and should be impenetrable. The initiator CHAP secret must be unique and cannot be the same as the target CHAP secret. If not specified, a random secret is created.
targetSecret	string	The CHAP secret to use for the target (mutual CHAP authentication). This secret must be 12-16 characters in length and should be impenetrable. The target CHAP secret must be unique and cannot be the same as the initiator CHAP secret. If not specified, a random secret is created.
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireAddAccountResult	The data resulting from adding the account.

getAccountByID

The getAccountByID action enables you to return details about a specific account, given its account ID.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
accountID	number	Specifies the account for which details are gathered.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetAccountResult	The details of the account.

getAccountByName

The getAccountByName action enables you to return details about a specific account, given the account name.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
username	string	Username for the account.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetAccountResult	The details of the account.

getAccountEfficiency

The getAccountEfficiency action enables you to retrieve efficiency statistics about a volume account. This action returns efficiency information only for the account you give as a parameter.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Input	Type	Description
accountID	number	Specifies the volume account for which efficiency statistics are returned.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetEfficiencyResult	The efficiency statistics for the volume account.

listAccounts

The listAccounts action returns the entire list of accounts.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
includeStorageContainers	boolean	Storage containers are included in the response by default. To exclude storage containers, set to `false`.
startAccountID	number	The starting AccountID to return. If no account exists with this AccountID, the next account by AccountID order is used as the start of the list.
limit	number	The maximum number of account objects to return.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListAccountsResult	The efficiency statistics for the volume account.

modifyAccount

The modifyAccount action enables you to modify an existing account.

Additional information

When you lock an account, any existing connections from that account are immediately terminated. When you change an account's CHAP settings, any existing connections remain active, and the new CHAP settings are used on subsequent connections or reconnections. To clear an account's attributes, specify {} for the attributes parameter.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
accountID	number	Specifies the ID of the account to modify.
username	string	The unique username for this account. May be 1 to 64 characters in length.
status	string	Sets the status for the account. Possible values: • `active`: The account is active and connections are allowed. • `locked`: The account is locked and connections are refused.
initiatorSecret	string	The CHAP secret to use for the initiator. This secret must be 12-16 characters in length and should be impenetrable. The initiator CHAP secret must be unique and cannot be the same as the target CHAP secret. If not specified, a random secret is created.
targetSecret	string	The CHAP secret to use for the target (mutual CHAP authentication). This secret must be 12-16 characters in length and should be impenetrable. The target CHAP secret must be unique and cannot be the same as the initiator CHAP secret. If not specified, a random secret is created.
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireModifyAccountResult	Empty.

removeAccount

The removeAccount action enables you to remove an existing account.

Additional information

You must delete and purge all volumes associated with the account using the deleteVolume action before you can remove the account. If volumes on the account are still pending deletion, you cannot use removeAccount to remove the account.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
accountID	number	The account ID of the account to remove.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireRemoveAccountResult	Empty.

Asynchronous result actions

Asynchronous result actions enable you to get the results of individual asynchronous operations or list the status of all running or completed asynchronous operations on the system.

getAsyncResult

The getAsyncResult action enables you to retrieve the result of asynchronous method calls.

Additional information

Some actions require some time to run, and may not be finished when the system sends the initial response. To obtain the status or result of the action, use getAsyncResult to poll the asyncHandle value returned by the method.

getAsyncResult returns the overall status of the operation (in progress, completed, or error) in a standard fashion, but the actual data returned for the operation depends on the original action and the return data is documented with each method.

If the `keepResult` parameter is missing or false, the asyncHandle becomes inactive once the result is returned, and later attempts to query that asyncHandle return an error. You can keep the asyncHandle active for future queries by setting `keepResult` to `true`.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
asyncHandle	number	A value that was returned from the original asynchronous method call.
keepResult	boolean	If true, GetAsyncResult does not remove the asynchronous result upon returning it, enabling future queries to that asyncHandle

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireAttributes	Information about the status or result of the asynchronous operation.

listAsyncResults

You can use the listAsyncResults action to list the results of all currently running and completed asynchronous operations on the system.

Additional information

Querying asynchronous results with listAsyncResults does not cause completed asyncHandles to expire; you can use getAsyncResult to query any of the asyncHandles returned by listAsyncResults.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Input	Type	Description
asyncResultTypes	Array of string	An optional list of types of results. You can use this list to restrict the results to only these types of operations. Possible values: • `BulkVolume`: Copy operations between volumes, such as backups or restores. • `Clone`: Volume cloning operations. • `DriveRemoval`: Operations involving the system copying data from a drive in preparation to remove it from the cluster. • `RtfiPendingNode`: Operations involving the system installing compatible software on a node before adding it to the cluster.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListAsyncResultsResult	The list of asyncHandles and result information of currently running and completed asynchronous operations.

Backup target actions

Backup target actions enable you to perform various tasks related to backup targets. A backup target is a convenient way to store the details required to back up to an Amazon S3 target, such as location and credentials.

createBackupTarget

CreateBackupTarget enables you to create and store backup target information so that you do not need to re-enter it each time a backup is created.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
name	string	The name for the backup target.

Input	Type	Description
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCreateBackupTargetResult	Unique identifier assigned to the new backup target.

getBackupTarget

The getBackupTarget action enables you to return information about a specific backup target that you have created.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
backupTargetID	number	The unique identifier of the backup target.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetBackupTargetResult	Information about the backup target.

listBackupTargets

The listBackupTargets action enables you to retrieve information about all backup targets that have been created.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListBackupTargetsResult	Information about each backup target.

modifyBackupTarget

The modifyBackupTarget action enables you to change the attributes of a backup target.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
backupTargetID	number	The unique target ID of the target to modify.
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.
name	string	A new name for the backup target.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireModifyBackupTargetResult	Empty.

removeBackupTarget

The removeBackupTarget action enables you to delete backup targets.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
backupTargetID	number	The unique target ID of the target to remove.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireRemoveBackupTargetResult	Empty.

Cluster actions

Cluster actions enable you to manage the configuration and topology of the cluster and the nodes that belong to a cluster.

Some of these actions operate on nodes that are part of a cluster, or have been configured to join a cluster. For actions that operate on individual nodes, see *Node actions* on page 65. You can add

nodes to a new cluster or to an existing cluster. Nodes that are ready to be added to a cluster are in a "pending" state, which means they have been configured but not yet added to the cluster.

clearClusterFaults

The clearClusterFaults action enables you to clear information about both current and previously detected faults. Both resolved and unresolved faults can be cleared.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
faultTypes	string	Determines the types of faults cleared. Possible values: `current`: Faults that are currently detected and have not been resolved. `resolved`: (Default) Faults that were previously detected and resolved. `all`: Both current and resolved faults are cleared. The fault status can be determined by the “resolved” field of the fault object.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireClearClusterFaultsResult	Empty.

createSupportBundle

The createSupportBundle action enables you to create a support bundle file that is stored on the node's filesystem. After creation, the bundle is stored on the node as a tar.gz file.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
bundleName	string	A unique name for the support bundle. If no name is provided, then "supportbundle" and the node name are used as the file name.

Input	Type	Description
extraArgs	string	This parameter is passed to the sf_make_support_bundle script. It should be used only at the request of NetApp SolidFire Support.
timeoutSec	number	The number of seconds to allow the support bundle creation script to run before stopping. The default is 1500 seconds.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCreateSupportBundleResult	Details of the support bundle and the support bundle creation process. This type contains a URL to aid in retrieval of the newly created support bundle from the directory on the node where it was created.

deleteAllSupportBundles

The deleteAllSupportBundles action enables you to delete all support bundles generated with the createSupportBundle action.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCreateSupportBundleResult	Empty.

disableEncryptionAtRest

The disableEncryptionAtRest action enables you to remove the encryption that was previously applied to the cluster using the enableEncryptionAtRest action.

Additional information

This action is asynchronous and returns a response before encryption is disabled. You can use the getClusterInfo action to poll the system to see when the process has completed.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireDisableEncryptionAtRestResult	Empty.

enableEncryptionAtRest

The enableEncryptionAtRest action enables you to enable the Advanced Encryption Standard (AES) 256-bit encryption at rest on the cluster so that the cluster can manage the encryption key used for the drives on each node. This feature is not enabled by default.

Additional information

When you enable encryption at rest, the cluster automatically manages encryption keys internally for the drives on each node in the cluster. Nodes do not store the keys to unlock drives and the keys are never passed over the network. Two nodes participating in a cluster are required to access the key to disable encryption on a drive. The encryption management does not affect performance or efficiency on the cluster. If an encryption-enabled drive or node is removed from the cluster with the API or through vRO, encryption at rest is disabled and the data is not secure erased. You can securely erase data using the secureEraseDrives action.

You should only enable or disable encryption when the cluster is running and in a healthy state. You can enable or disable encryption at your discretion and as often as you need.

This process is asynchronous and returns a response before encryption is enabled. You can use the getClusterInfo action to poll the system to see when the process has completed.

Note: If you have a node type with a model number ending in "-NE", the enableEncryptionAtRest action will fail with a response of "Encryption not allowed. Cluster detected non-encryptable node".

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireEnableEncryptionAtRestResult	Empty.

getAPI

The getAPI action enables you to return a list of all the API methods and supported API endpoints that can be used in the system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetAPIResult	A list of all supported API methods.

getClusterCapacity

The getClusterCapacity action enables you to return high-level capacity measurements for an entire cluster.

Additional information

This action returns fields that you can use to calculate the efficiency rates shown in the Element OS Web UI. You can use the efficiency calculations in workflows to return the efficiency rates for thin provisioning, deduplication, compression, and overall efficiency.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetClusterCapacityResult	Cluster capacity details.

getClusterConfig

The getClusterConfig action enables you to return information about the cluster configuration this node uses to communicate with the cluster it is a part of.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetClusterConfigResult	Cluster configuration information the node uses to communicate with the cluster.

getClusterFullThreshold

The getClusterFullThreshold action enables you to view the stages set for cluster fullness levels. This method returns all fullness metrics for the cluster.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetClusterFullThresholdResult	Cluster fullness status and statistics.

getClusterInfo

The getClusterInfo action enables you to return configuration information about the cluster.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetClusterInfoResult	Cluster configuration information.

getClusterMasterNodeID

The getClusterMasterNodeID action enables you to retrieve the ID of the node that can perform cluster-wide administration tasks and holds the storage virtual IP address (SVIP) and management virtual IP address (MVIP).

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetClusterMasterNodeIDResult	ID of the master node.

getClusterState

The getClusterState action enables you to determine if a node is part of a cluster or not.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
force	boolean	To run this command, this parameter must be set to true.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetClusterStateResult	The name and state of the cluster.

getClusterStats

The getClusterStats action enables you to retrieve high-level activity measurements for the cluster. Values returned are cumulative from the creation of the cluster.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetClusterStatsResult	Cluster activity information.

getClusterVersionInfo

The getClusterVersionInfo action enables you to retrieve information about the Element software version running on each node in the cluster. This action also returns information about nodes that are currently in the process of upgrading software

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetClusterVersionInfoResult	Information about the Element OS version running on the the node.

getCompleteStats

NetApp engineering uses the getCompleteStats action to troubleshoot new features. The data returned from getCompleteStats is not documented, changes frequently, and is not guaranteed to be accurate. NetApp does not recommend using getCompleteStats for collecting performance data or any other management integration with a SolidFire cluster.

Additional information

If you do not provide a snapshotID with this action, a snapshot is created from the volume's active branch. If the volume from which the snapshot is created is being replicated to a remote cluster, the snapshot can also be replicated to the same target. Use the enableRemoteReplication parameter to enable snapshot replication.

Snapshot creation depends on cluster fullness. See the *NetApp SolidFire Element OS API Reference Guide* for more information.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireAttributes	System statistics.

getLimits

The getLimits action enables you to retrieve the limit values set by the API. These values might change between releases of Element OS, but do not change without an update to the system. Knowing the limit values set by the API can be useful when writing scripts for user-facing tools.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetLimitsResult	Information about the system limits.

getNtpInfo

The getNtpInfo action enables you to retrieve the current network time protocol (NTP) configuration information.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetNtpInfoResult	Information about the current NTP configuration.

getRawStats

NetApp engineering uses the getRawStats action to troubleshoot new features. The data returned from getRawStats is not documented, changes frequently, and is not guaranteed to be accurate. NetApp does not recommend using getRawStats for collecting performance data or any other management integration with a SolidFire cluster.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireAttributes	System statistics.

getSystemStatus

The getSystemStatus action enables you to query the system for status.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetSystemStatusResult	Information about the current system status.

listClusterFaults

ListClusterFaults enables you to retrieve information about any faults detected on the cluster. With this method, you can retrieve both current faults as well as faults that have been resolved. The system caches faults every 30 seconds.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
bestPractices	boolean	Include faults triggered by sub-optimal system configuration. Possible values: • `true` • `false`

Input	Type	Description
faultTypes	string	Determines the types of faults returned: • `current`: List active, unresolved faults. • `resolved`: List faults that were previously detected and resolved. • `all`: (Default) List both current and resolved faults.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListClusterFaultsResult	Information about the requested cluster faults.

listEvents

The listEvents action enables you to retrieve events detected on the cluster, sorted from oldest to newest.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
startEventID	number	Identifies the beginning of a range of events to return.
endEventID	number	Identifies the end of a range of events to return.
maxEvents	number	Specifies the maximum number of events to return.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListEventsResult	The list of events.

listSyncJobs

The listSyncJobs action enables you to return information about synchronization jobs that are running on a SolidFire cluster. This action returns slice, clone, and remote synchronization jobs.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListSyncJobsResult	The list of objects describing sync process that are currently running in the system.

modifyClusterFullThreshold

The modifyClusterFullThreshold action enables you to change the level at which the system generates an event when the storage cluster approaches a certain capacity utilization.

Additional information

You can use the threshold setting to indicate the acceptable amount of utilized block storage before the system generates a warning. For example, if you want to be alerted when the system reaches 3% below the "Error" level block storage utilization, enter a value of "3" for the stage3BlockThresholdPercent parameter. If this level is reached, the system sends an alert to the event log in the cluster management console.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
stage2AwareThreshold	number	The number of nodes of capacity remaining in the cluster before the system triggers a capacity notification.
stage3BlockThresholdPercent	number	The percentage of block storage utilization below the "Error" threshold that causes the system to trigger a cluster "Warning" alert.

Input	Type	Description
maxMetadataOverProvisionFactor	number	A value representative of the number of times metadata space can be over provisioned relative to the amount of space available. For example, if there was enough metadata space to store 100 TiB of volumes and this number was set to 5, then 500 TiB worth of volumes could be created.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireModifyClusterFullThresholdResult	Information about the current cluster fullness threshold settings.

setNtpInfo

The setNtpInfo action enables you to configure NTP on cluster nodes.

Additional information

The values you set with this interface apply to all nodes in the cluster. If an NTP broadcast server periodically broadcasts time information on your network, you can optionally configure nodes as broadcast clients.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
servers	Array of string	A list of NTP servers to add to each node's NTP configuration.
broadcastclient	boolean	Enable every node in the cluster as a broadcast client.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireSetNtpInfoResult	Empty.

Cluster admin actions

Cluster admin actions enable you to manage cluster administrator users on the system.

addClusterAdmin

The addClusterAdmin action enables you to add a new Cluster Admin account to the system. A cluster admin can manage the cluster via the API and management tools. Cluster admin accounts are completely separate and unrelated to standard tenant accounts.

Additional information

Each cluster admin can be restricted to a subset of the API. You should use multiple cluster admin accounts for different users and applications. As a best practice, give each cluster admin the minimal permissions necessary; this reduces the potential impact of credential compromise.

You must accept the End User License Agreement (EULA) by setting the "acceptEula" parameter to true to add a cluster administrator account to the system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
access	number	The account ID of the account to remove.
acceptEula	boolean	Accept the End User License Agreement. Set to `true` to add a cluster administrator account to the system. If omitted or set to `false`, the action fails.
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.
password	string	The password used to authenticate this cluster admin.
username	string	The unique username for this cluster admin. Must be between 1 and 1024 characters in length.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireAddClusterAdminResult	The cluster admin ID of the newly created cluster admin user.

getCurrentClusterAdmin

The getCurrentClusterAdmin action enables you to retrieve information about the current primary cluster administrator. The primary cluster admin user was created when the cluster was created.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetCurrentClusterAdminResult	Information about the cluster admin.

listClusterAdmins

The listClusterAdmins action enables you to retrieve a list of all cluster administrators for the cluster.

Additional information

There can be several cluster administrator accounts with different levels of permissions, but there can be only one primary cluster administrator in the system. The primary cluster admin is the administrator that was created when the cluster was created. LDAP administrators can also be created when setting up an LDAP system on the cluster.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListClusterAdminsResult	Information about all cluster and LDAP administrators that exists for a cluster.

modifyClusterAdmin

The modifyClusterAdmin action enables you to change the settings for a local or LDAP-based cluster administrator user. You cannot change settings for the primary cluster administrator account.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
access	Array of string	Controls which actions this cluster administrator can use. For more information, see the *NetApp SolidFire Element OS API Reference Guide*.

Input	Type	Description
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.
clusterAdminID	number	The ID for the local or LDAP-based cluster administrator account to modify.
password	string	The password for this cluster administrator user.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireModifyClusterAdminResult	Empty.

removeClusterAdmin

The removeClusterAdmin action enables you to remove a cluster administrator user. You cannot remove the primary cluster admin account.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
clusterAdminID	number	The ID of the cluster administrator user to remove.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireRemoveClusterAdminResult	Empty.

Drive actions

Drive actions enable you to add and manage drives that are available to a cluster. When you add a node to the cluster or install new drives in an existing node, the drives are available to be added to the cluster.

addDrives

The addDrives action enables you to add one or more available drives to the cluster, enabling the drives to host a portion of the cluster's data.

Additional information

When you add a node to the cluster or install new drives in an existing node, the new drives are marked as "available" and must be added via addDrives before they can be utilized. Use the ListDrives workflow to display drives that are "available" to be added. When you add multiple drives, it is more efficient to add them in a single addDrives operation rather than multiple individual methods with a single drive each. This reduces the amount of data balancing that must occur to stabilize the storage load on the cluster.

When you add a drive, the system automatically determines the "type" of drive it should be.

The action is asynchronous and returns immediately. However, it may take some time for the data in the cluster to be rebalanced using the newly added drives. As the new drive(s) are syncing on the system, you can use the ListSyncJobs workflow to see how the drive(s) are being rebalanced and the progress of adding the new drive. You can also use the GetAsyncResult workflow to query this workflow's returned asyncHandle.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
drives	Array of SolidFire:SolidFireNewDrive	Information about each drive to be added to the cluster: • `driveID`: (number) The ID of the drive to add. • `type`: (Optional) (string) The type of drive to add. Valid values are "slice" or "block". If omitted, the system assigns the correct type.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireAddDrivesResult	Empty.

getDriveConfig

The getDriveConfig action enables you to retrieve drive information for expected slice and block drive counts as well as the number of slices and block drives that are currently connected to the node.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetDriveConfigResult	Information about the drives that are connected to the node.

getDriveHardwareInfo

The getDriveHardwareInfo action enables you to retrieve all hardware info for the given drive. This generally includes manufacturers, vendors, versions, and other associated hardware identification information.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
driveID	number	The ID of the drive to retrieve information about. You can obtain driveIDs using the listDrives action.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetDriveHardwareInfoResult	Hardware information for the specified drive.

getDriveStats

The getDriveStats action retrieves high-level activity measurements for a single drive. Values are cumulative from the addition of the drive to the cluster. Some values are specific to block drives.

Additional information

Statistical data may not be returned for both block and metadata drives when you use this action. For more information on what data each drive type returns, see the *NetApp SolidFire Element OS API Reference Guide*.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
driveID	number	Specifies the drive for which statistics are gathered.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetDriveStatsResult	Drive activity information.

listDriveHardware

The listDriveHardware action returns all the drives connected to a node. You can use this action on individual nodes to return drive hardware information or use this action on the cluster master node MVIP to see information for all the drives on all nodes.

Additional information

The `"securitySupported": true` line of the action response does not imply that the drives are capable of encryption; only that the security status can be queried. If you have a node type with a model number ending in "-NE", commands to enable security features on these drives will fail. See the enableEncryptionAtRest action for more information.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListDriveHardwareResult	Drive hardware information for the node.

listDrives

The listDrives action enables you to retrieve the list of the drives that exist in the cluster's active nodes. This action returns drives that have been added as volume metadata or block drives. It also returns drives that have not been added and are available.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListDrivesResult	A list of drives in the cluster.

listDriveStats

The listDriveStats action enables you to retrieve high-level activity measurements for multiple drives in the cluster.

Additional information

By default, this action returns statistics for all drives in the cluster, and these measurements are cumulative from the addition of the drive to the cluster. Some values this action returns are specific to block drives, and some are specific to metadata drives.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
drives	Array of number	An optional list of driveIDs for which to return drive statistics. If you omit this parameter, measurements for all drives are returned.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListDriveStatsResult	Drive activity information.

removeDrives

The removeDrives action enables you to proactively remove drives that are part of the cluster.

Additional information

You may want to use this action when reducing cluster capacity or preparing to replace drives nearing the end of their service life. Any data on the drives is removed and migrated to other drives in the cluster before the drive is removed from the cluster. This is an asynchronous action. Depending on the total capacity of the drives being removed, it may take several minutes to migrate all of the data. Use the getAsyncResult action to check the status of the remove operation.

When removing multiple drives, use a single removeDrives action rather than multiple individual actions with a single drive each. This reduces the amount of data balancing that must occur to evenly distribute the storage load on the cluster.

You can also remove drives with a "failed" status using removeDrives. When you remove a drive with a "failed" status it is not returned to an "available" or “active” status. The drive is unavailable for use in the cluster.

Use the listDrives action to obtain the driveIDs for the drives you want to remove.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
drives	Array of number	The list of drive IDs of the drives to remove from the cluster.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireRemoveDrivesResult	The asyncHandle that you can use to obtain the operation result.

resetDrives

The resetDrives action enables you to proactively initialize drives and remove all data currently residing on a drive. The drive can then be reused in an existing node or used in an upgraded node. This action requires the force=true parameter to be included.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
drives	string	The list of device names (not driveIDs) to reset.
force	boolean	The "force" parameter must be included to successfully reset a drive.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireResetDrivesResult	The details of drives that are being reset.

secureEraseDrives

The secureEraseDrives action enables you to remove any residual data from drives that have a status of "available." You may want to use this method when replacing a drive nearing the end of its service life that contained sensitive data.

Additional information

This action uses a Security Erase Unit command to write a predetermined pattern to the drive and resets the encryption key on the drive. This asynchronous action may take up to two minutes to complete. You can use the getAsyncResult action to check on the status of the secure erase operation. You can use the listDrives action to obtain the driveIDs for the drives you want to secure erase.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
drives	Array of number	The list of driveIDs of drives to secure erase.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireSecureEraseDrivesResult	The asyncHandle value you can use to obtain the result of the operation.

testDrives

The testDrives action enables you to run a hardware validation on all drives in the node. This action detects hardware failures on the drives (if present) and reports them in the results of the validation tests. You can only use this action on nodes that are not actively participating in a cluster.

Additional information

This test takes approximately 10 minutes.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
force	boolean	Include the "force" parameter with this action to successfully test the drives on the node.
minutes	number	Specifies the number of minutes to run the test.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireTestDrivesResult	Details of the test results.

Hardware actions

Hardware actions enable you to retrieve detailed information on cluster and node hardware configurations.

getClusterHardwareInfo

The getClusterHardwareInfo action to retrieve the hardware status and information for all Fibre Channel nodes, iSCSI nodes and drives in the cluster. This generally includes manufacturers, vendors, versions, and other associated hardware identification information.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
type	string	Include only a certain type of hardware information in the response. Can be one of the following: • `drives`: List only drive information in the response. • `nodes`: List only node information in the response. • `all`: Include both drive and node information in the response. If this parameter is omitted, a type of `all` is assumed.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetClusterHardwareInfoResult	Hardware information for cluster nodes, drives, or both. Each object in the output is labeled with the ID of the related node.

getHardwareConfig

The getHardwareConfig action enables you to retrieve the hardware configuration information for a node.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetHardwareConfigResult	A list of hardware information and current settings.

getHardwareInfo

The getHardwareInfo action enables you to retrieve hardware information and status for a single node. This generally includes manufacturers, vendors, versions, drives, and other associated hardware identification information.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
force	boolean	Set the `force=true` to run on all nodes in the cluster.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetHardwareInfoResult	Hardware information for this node.

getNodeHardwareInfo

The getNodeHardwareInfo action enables you to retrieve all hardware and status information for the specified node. This generally includes manufacturers, vendors, versions, and other associated hardware identification information.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
nodeID	number	The ID of the node for which hardware information is being requested. Information about a Fibre Channel node is returned if you specify a Fibre Channel node.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetNodeHardwareInfoResult	Hardware information for the specified nodeID. Each object in this output is labeled with the nodeID of the given node.

getNvramInfo

GetNvramInfo enables you to retrieve information from each node about the NVRAM card.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
force	boolean	The `force` parameter must be included with this action to successfully run on all nodes in the cluster.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetNvramInfoResult	An object containing events and errors detected on the NVRAM card.

Initiator actions

Initiator actions enable you to add, remove, view, and modify initiators.

addInitiatorsToVolumeAccessGroup

The addInitiatorsToVolumeAccessGroup action enables you to add initiators to a specified volume access group.

Additional information

The accepted format of an initiator IQN is: iqn.yyyy-mm where y and m are digits, followed by text which must only contain digits, lower-case alphabetic characters, a period (.), colon (:) or dash (-). For example:

```
iqn.2010-01.com.solidfire:17oi.solidfire-0.1
```

The accepted format of a Fibre Channel initiator WWPN is: Aa:bB:CC:dd:11:22:33:44, or AabBCCdd11223344. For example:

```
21:00:00:0e:1e:11:f1:81
```

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
initiators	Array of number	(Required) List of initiator IDs or names (IQNs and WWPNs) to include in the volume access group. If you pass a list of initiator names, the initiators are created if they do not already exist. If you pass a list of initiator IDs, the method returns an error if any of the initiators does not already exist. Passing initiator names is deprecated; you should use initiator IDs whenever possible.
volumeAccessGroupID	number	The ID of the volume access group to which the initiator should be added.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireModifyVolumeAccessGroupResult	Information about the newly modified volume access group.

createInitiators

The createInitiators action enables you to create multiple new initiator IQNs or World Wide Port Names (WWPNs) and optionally assign them aliases and attributes. When you use createInitiators to create new initiators, you can also add them to volume access groups.

Additional information

If createInitiators fails to create one of the initiators provided in the parameter, the action returns an error and does not create any initiators (no partial completion is possible).

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Input	Type	Description
initiators	Array SolidFire:SolidFireCreateInitiator	A list containing characteristics of each new initiator: • `name`: (Required) (string)The name of the initiator (IQN or WWPN) to create. • `alias`: (Optional) (string) The friendly name to assign to this initiator. • `attributes`: (Optional) (string) Stringified JSON with key/value pairs of all attributes to assign to this initiator. • `volumeAccessGroupID`: (Optional) (number) The ID of the volume access group into to which this newly created initiator will be added.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCreateInitiatorsResult	A list of objects containing details about the newly created initiators.

deleteInitiators

The deleteInitiators action enables you to delete one or more initiators from the system (and from any associated volumes or volume access groups).

Additional information

If deleteInitiators fails to delete one of the initiators provided in the parameter, the action returns an error and does not delete any initiators (no partial completion is possible).

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
initiators	Array of number	An array of IDs of initiators to delete.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireDeleteInitiatorsResult	Empty.

listInitiators

The listInitiators action enables you to retrieve initiator IQNs or World Wide Port Names (WWPNs).

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
initiators	Array of number	A list of initiator IDs to retrieve. You can supply this parameter or the `startInitiatorID` parameter, but not both.
limit	number	The maximum number of initiator objects to return.
startInitiatorID	number	The initiator ID at which to begin the listing. You can supply this parameter or the `initiators` parameter, but not both.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListInitiatorsResult	A list of the initiator information.

modifyInitiators

The modifyInitiators action enables you to change the attributes of one or more existing initiators. You cannot change the name of an existing initiator. If you need to change the name of an initiator, delete it first with deleteInitiators and create a new one with createInitiators.

Additional information

If modifyInitiators fails to change one of the initiators provided in the parameter, the action returns an error and does not change any initiators (no partial completion is possible).

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Input	Type	Description
initiators	Array of SolidFire:SolidFireModifyInitiator	A list containing characteristics of each initiator to modify: • `initiatorID`: (Required) (string)The ID of the initiator to modify. • `alias`: (Optional) (string) A new friendly name to assign to this initiator. • `attributes`: (Optional) (string) Stringified JSON data with key-value pairs of all attributes to assign to this initiator. • `volumeAccessGroupID`: (Optional) (number) The ID of the volume access group into to which this newly created initiator will be added. If the initiator was previously in a different volume access group, it is removed from the old volume access group. If this key is present but null, the initiator is removed from its current volume access group, but not placed in any new volume access group.
limit	number	The maximum number of initiator objects to return.
startInitiatorID	number	The initiator ID at which to begin the listing. You can supply this parameter or the `initiators` parameter, but not both.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireModifyInitiatorsResult	A list of objects describing the newly modified initiators.

removeInitiatorsFromVolumeAccessGroup

The removeInitiatorsFromVolumeAccessGroup action enables you to remove initiators from a specified volume access group.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeAccessGroupID	number	The volumeAccessGroupID of the volume access group from which initiator or initiators are removed.
initiators	Array of integers or string	List of initiator IDs or names (IQNs and WWPNs) to include in the volume access group. If you pass a list of initiator names, the initiators are created if they do not already exist. If you pass a list of initiator IDs, the action returns an error if any of the initiators does not already exist. Passing initiator names is deprecated; you should use initiator IDs whenever possible.
deleteOrphanInitiators	boolean	• `true`: Delete initiator objects after they are removed from a volume access group. • `false`: Do not delete initiator objects after they are removed from a volume access group.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireModifyVolumeAccessGroupResult	Information about the newly modified volume access group.

LDAP actions

You can use the Lightweight Directory Access Protocol (LDAP) to authenticate access to SolidFire storage. The actions described in this section enable you to configure LDAP access to the storage system.

addLdapClusterAdmin

The addLdapClusterAdmin action enables you to add a new LDAP cluster administrator user. An LDAP cluster administrator can manage the cluster via the API and management tools. LDAP cluster admin accounts are completely separate and unrelated to standard tenant accounts.

Additional information

You can also use this action to add an LDAP group that has been defined in Active Directory®. The access level that is given to the group is passed to the individual users in the LDAP group.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
access	Array of string	Controls which methods this cluster admin can use. For more details on the levels of access, see the *NetApp SolidFire Element OS API Reference Guide*.
acceptEula	boolean	Accept the End User License Agreement. Set to `true` to add a cluster administrator account to the system. If omitted or set to `false`, the action fails.
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.
username	string	The distinguished user name for the new LDAP cluster admin.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireAddLdapClusterAdminResult	Empty.

disableLdapAuthentication

The disableLdapAuthentication action enables you to disable LDAP authentication and remove all LDAP configuration settings. This action does not remove any configured cluster admin accounts (user or group). However, those cluster admin accounts will no longer be able to log in.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireDisableLdapAuthenticationResult	Empty.

enableLdapAuthentication

The enableLdapAuthentication action enables you to configure an LDAP directory connection to use for LDAP authentication to acluster. Users that are members of the LDAP directory can then log in to the storage system using their LDAP credentials.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
authType	string	Identifies which user authentication method to use. Must be one of the following: • `DirectBind` • `SearchAndBind`
groupSearchBaseDN	string	The base DN of the tree to start the group search (will do a subtree search from here).
groupSearchType	string	Controls the default group search filter used, and must be one of the following: • `NoGroups`: No group support. • `ActiveDirectory`: Nested membership of all of a user's AD groups. • `MemberDN`: MemberDN style groups (single-level).
serverURIs	Array of string	A comma-separated list of LDAP server URIs (examples: `ldap://1.2.3.4` and `ldaps://1.2.3.4:123`)
userSearchBaseDN	string	The base DN of the tree to start the search (will do a subtree search from here).
searchBindDN	string	A fully qualified DN to log in with to perform an LDAP search for the user (needs read access to the LDAP directory).

Input	Type	Description
searchBindPassword	string	The password for the searchBindDN account used for searching.
userSearchFilter	string	The LDAP filter to use. The string should have the placeholder text %USERNAME% which is replaced with the username of the authenticating user. Example: `(& (objectClass=person) (sAMAccountName= %USERNAME%))` will use the `sAMAccountName` field in Active Directory® to match the username entered at cluster login.
userDNTemplate	string	A string that is used to form a fully qualified user DN. The string should have the placeholder text %USERNAME% which is replaced with the username of the authenticating user.
groupSearchCustomFilter	string	For use with the CustomFilter search type, an LDAP filter to use to return the DNs of a user's groups. The string can have placeholder text of %USERNAME% and %USERDN% to be replaced with their username and full userDN as needed.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireEnableLdapAuthenticationResult	Empty.

getLdapConfiguration

The getLdapConfiguration action enables you to get the currently active LDAP configuration on the cluster.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetLdapConfigurationResult	A list of the current LDAP configuration settings. This action does not retrieve the plain text of the search account password. If LDAP authentication is currently disabled, all the returned settings are empty with the exception of "authType", and "groupSearchType" which are set to “SearchAndBind” and “ActiveDirectory” respectively.

testLdapAuthentication

The testLdapAuthentication action enables you to validate the currently enabled LDAP authentication settings. If the configuration is correct, the action returns the group membership of the tested user.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
username	string	The username to be tested.
password	string	The password for the username to be tested.
ldapConfiguration	SolidFire:SolidFireLdapConfiguration	An ldapConfiguration object to be tested. If this parameter is provided, the action tests the provided configuration even if LDAP authentication is currently disabled.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireTestLdapAuthenticationResult	LDAP DN and group membership of the tested user.

Logging actions

Logging actions enable you to view and change information about the current login session and remote logging hosts.

getLoginSessionInfo

The getLoginSessionInfo action enables you to return the period of time a log in authentication session is valid for both login shells and the TUI.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetLoginSessionInfoResult	Information about the authentication expiration period.

getRemoteLoggingHosts

The getRemoteLoggingHosts action enables you to retrieve the current list of log servers.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetRemoteLoggingHostsResult	A lists of hosts to forward logs to.

setLoginSessionInfo

The setLoginSessionInfo action enables you to set the period of time that a session's login authentication is valid. After the log in period elapses without activity on the system, the authentication expires. New login credentials are required for continued access to the cluster once the timeout period has elapsed.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Input	Type	Description
timeout	string	The cluster authentication expiration period. Formatted in HH:mm:ss. For example: 01:30:00, 00:90:00, and 00:00:5400 can all be used to equal a 90 minute timeout period. The default value is 30 minutes.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireSetLoginSessionInfoResult	Empty.

setRemoteLoggingHosts

The setRemoteLoggingHosts action enables you to configure remote logging from the nodes in the storage cluster to a centralized log server or servers. Remote logging is performed over TCP using the default port 514.

Additional information

This action does not add to the existing logging hosts. Rather, it replaces what currently exists with new values specified by this action. You can use the GetRemoteLoggingHosts workflow to determine what the current logging hosts are and then use setRemoteLoggingHosts to set the desired list of current and new logging hosts.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
remoteHosts	Array of SolidFire:SolidFireLoggingServer	A list of hosts to send log messages to.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireSetRemoteLoggingHostsResult	Empty.

Network actions

Network actions enable you to view information about Fibre Channel and iSCSI connections to the cluster, as well as network interface information for nodes.

listFibreChannelPortInfo

The listFibreChannelPortInfo action enables you to return information about the Fibre Channel ports.

Additional information

This action is intended for use on individual nodes; a userid and password is required for access to individual Fibre Channel nodes. However, you can use this action on the cluster if you include the `force=true` parameter in the method call. When you use this action on the cluster, all Fibre Channel interfaces are listed.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
force	boolean	Set `force=true` to run on all nodes in the cluster.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListFibreChannelPortInfoResult	A list of all physical Fibre Channel ports, or a port for a single node.

listFibreChannelSessions

The listFibreChannelSessions action enables you to retrieve information about the active Fibre Channel sessions on a cluster.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListFibreChannelSessionsResult	A list of objects describing active Fibre Channel sessions on the cluster.

listISCSISessions

The listISCSISessions action enables you to return iSCSI connection information for volumes in the cluster.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListISCSISessionsResult	Information for each iSCSI session.

listNetworkInterfaces

The listNetworkInterfaces action enables you to retrieve information about each network interface on a node.

Additional information

This action is intended for use on individual nodes; a userid and password is required for access to individual nodes. However, you can use this action on the cluster if you include the `force=true` parameter in the method call. When you use this action on the cluster, all interfaces are listed.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
force	boolean	Set `force=true` to run on all nodes in the cluster.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListNetworkInterfacesResult	Information about network interfaces for a node or nodes.

listNodeFibreChannelPortInfo

The listNodeFibreChannelPortInfo action enables you to retrieve information about the Fibre Channel ports on a node.

Additional information

This action is intended for use on individual nodes; a userid and password is required for access to individual Fibre Channel nodes. However, you can use this action on the cluster if you include the `force=true` parameter in the method call. When you use this action on the cluster, all Fibre Channel interfaces are listed.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
force	boolean	Set `force=true` to run on all nodes in the cluster.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListNodeFibreChannelPortInfoResult	A list of all physical Fibre Channel ports, or a port for a single node.

Node actions

You can use node API methods to configure individual nodes. The methods in this section operate on nodes that need to be configured, are configured but not yet participating in a cluster, or are actively participating in a cluster. These methods enable you to view and modify settings for individual nodes and the cluster network used to communicate with the node.

addNodes

The addNodes action enables you to add one or more new nodes to a cluster.

Additional information

When a node that is not configured starts up for the first time, you are prompted to configure the node. Once you configure the node, it is registered as a "pending node" with the cluster.

SF-series clusters automatically image a node to the Element OS version on the cluster if the node and cluster are not at compatible versions. When you add a pending node that is running an incompatible software version, the method response includes an asyncHandle value that you can use with the GetAsyncResult method to query the status of the automatic imaging process.

The process of adding a Fibre Channel node is the same as adding SF-series iSCSI nodes to a cluster. Fibre Channel nodes are registered in the system with a NodeID. When they become accessible, they are put in a "pending node" status. The ListAllNodes API method will return the pendingNodeID for iSCSI nodes as well as any Fibre Channel nodes that are available to add to the cluster.

When you add a node to a cluster that you have configured for virtual networking, the system requires a sufficient number of virtual storage IP addresses to allocate a virtual IP to the new node. If there are no virtual IP addresses available for the new node, the AddNode operation fails. Use the ModifyVirtualNetwork method to add more storage IP addresses to your virtual network.

Once you add a node, any drives on the node are made available and you can add them using the AddDrives method to increase the storage capacity of the cluster.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
force	boolean	Include the "force" parameter with this action to successfully test the drives on the node.
minutes	number	Specifies the number of minutes to run the test.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireAddNodesResult	New IDs of the added nodes. May include asyncHandle information to use to query the imaging process if the new nodes need to be imaged.

getConfig

The getConfig action enables you to retrieve all configuration information for a node and its connection to the cluster. This action includes the same information available in both the getClusterConfig and getNetworkConfig actions.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetConfigResult	Details of the configuration information for the node.

getNetworkConfig

The getNetworkConfig action enables you to display the network configuration information for a node.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetNetworkConfigResult	Network connection types and current settings for each network interface of the node.

getNodeStats

The getNodeStats action enables you to retrieve the high-level activity measurements for a single node.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
nodeID	number	Specifies the node for which statistics are gathered.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetNodeStatsResult	Node activity information.

getOrigin

The getOrigin action enables you to retrieve the origination certificate for where the node was built.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetOriginResult	Vendor origination certification information. This value may be null if there is no origination certificaton.

getPendingOperation

The getPendingOperation action enables you to detect an operation on a node that is currently in progress. This action can also be used to report back when an operation has completed.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetPendingOperationResult	Information about the pending operation, if any.

listActiveNodes

The listActiveNodes action returns the list of currently active nodes that are in the cluster.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListActiveNodesResult	A list of active nodes in the cluster.

listAllNodes

The listAllNodes action enables you to retrieve a list of active and pending nodes in the cluster

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListAllNodesResult	A list of all nodes that are part of the cluster.

listNodeStats

The listNodeStats action enables you to view the high-level activity measurements for all nodes in a cluster.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListNodeStatsResult	Node activity information.

listPendingNodes

The listPendingNodes action enables you to retrieve a list of the currently pending nodes in the system. Pending nodes are nodes that are running and configured to join the cluster, but have not yet been added via the addNodes action.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListPendingNodesResult	A list of pending nodes in the cluster.

listPendingNodes

The listPendingNodes action enables you to retrieve a list of the currently pending nodes in the system. Pending nodes are nodes that are running and configured to join the cluster, but have not yet been added via the addNodes action.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListPendingNodesResult	A list of pending nodes in the cluster.

removeNodes

The removeNodes action enables you to remove one or more nodes that should no longer participate in the cluster.

Additional information

Before removing a node, you must remove all drives the node contains using the removeDrives action. You cannot remove a node until the removeDrives process has completed and all data has been migrated away from the node. Once you remove a node, it registers itself as a pending node. You can add the node again or shut it down (shutting the node down removes it from the pending node list).

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
nodes	Array of integers	A list of nodeIDs for the nodes to remove.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireRemoveNodesResult	Empty.

Pairing actions

Pairing actions enable you to create and manage pair relationships between clusters and volumes.

completeClusterPairing

The completeClusterPairing action is the second step in the cluster pairing process. Use this action with the encoded key received from the startClusterPairing action to complete the cluster pairing process.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
clusterPairingKey	string	A string of characters that is returned by the startClusterPairing action.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCompleteClusterPairingResult	A unique identifier for the cluster pair.

completeVolumePairing

The completeVolumePairing action enables you to complete the pairing of two volumes.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeID	number	The ID of the volume on which to complete the pairing process.
volumePairingKey	string	The key returned from the startVolumePairing action.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCompleteVolumePairingResult	Empty.

listActivePairedVolumes

The listActivePairedVolumes action enables you to list all of the active volumes paired with a volume. This action returns information about volumes with active and pending pairings.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListActivePairedVolumesResult	Volume information for the paired volumes.

listClusterPairs

The listClusterPairs action enables you to list all of the clusters a cluster is paired with. This action returns information about active and pending cluster pairings, such as statistics about the current pairing as well as the connectivity and latency (in milliseconds) of the cluster pairing.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListClusterPairsResult	Information about each paired cluster.

modifyVolumePair

The modifyVolumePair action enables you to pause or restart replication between a pair of volumes

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeID	number	Identification number of the volume to be modified.
pausedManual	boolean	Valid values that can be entered: • `true`: to pause volume replication. • `false`: to restart volume replication. If no value is specified, no change in replication is performed.

Input	Type	Description
mode	string	Volume replication mode. Possible values: • `Async`: Writes are acknowledged when they complete locally. The cluster does not wait for writes to be replicated to the target cluster. • `Sync`: The source acknowledges the write when the data is stored locally and on the remote cluster. • `SnapshotsOnly`: Only snapshots created on the source cluster are replicated. Active writes from the source volume are not replicated.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireModifyVolumePairResult	Empty.

removeClusterPair

The removeClusterPair action enables you to close the open connections between two paired clusters.

Additional information

Before you remove a cluster pair, you must first remove all volume pairing to the clusters with the removeVolumePair action.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
clusterPairID	number	The unique identifier used to pair two clusters.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireRemoveClusterPairResult	Empty.

removeVolumePair

The removeVolumePair action enables you to remove the remote pairing between two volumes. Use this action on both the source and target volumes that are paired together. When you remove the volume pairing information, data is no longer replicated to or from the volume.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeID	number	The ID of the volume on which to stop the replication process.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireRemoveVolumePairResult	Empty.

startClusterPairing

The startClusterPairing action enables you to create an encoded key from a cluster that is used to pair with another cluster. The key created from this action is used in the completeClusterPairing action to establish a cluster pairing. You can pair a cluster with a maximum of four other clusters.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireStartClusterPairingResult	Information about the new cluster pair.

startVolumePairing

The startVolumePairing action enables you to create an encoded key from a volume that is used to pair with another volume. The key that this action creates is used in the completeVolumePairing actioin to establish a volume pairing.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Input	Type	Description
mode	boolean	The mode of the volume on which to start the pairing process. The mode can only be set if the volume is the source volume. Possible values: • `Async`: (default if no mode parameter specified) Writes are acknowledged when they complete locally. The cluster does not wait for writes to be replicated to the target cluster. • `Sync`: Source acknowledges write when the data is stored locally and on the remote cluster. • `SnapshotsOnly`: Only snapshots created on the source cluster are replicated. Active writes from the source volume are not replicated.
volumeID	number	The ID of the volume on which to start the pairing process.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireStartVolumePairingResult	A string of characters that is used by the completeVolumePairing action.

Restart actions

Restart actions enable you to restart node networking and reset, restart, and shut down individual nodes.

resetNode

The resetNode action enables you to reset a node to the factory settings.

Additional information

All data, packages (software upgrades, etc), configurations, and log files are deleted from the node when you use this action. However, network settings for the node are preserved during this operation. Nodes that are participating in a cluster cannot be reset to the factory settings. The resetNode action can only be used on nodes that are in an "Available" state. It cannot be used on nodes that are "Active" in a cluster, or in a "Pending" state.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
build	string	Used to specify the URL to a remote Element software image to which the node will be reset.
force	boolean	Include this parameter to reset the node.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireResetNodeResult	Empty.

restartNetworking

The restartNetworking action enables you to restart the networking services on a node.

Additional information

This method restarts all networking services on a node, causing temporary loss of networking connectivity. Exercise caution when using this method.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
force	boolean	Include the "force" parameter with this action to successfully test the drives on the node.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireAttributes	Empty.

shutdown

The shutdown action enables you to restart or shutdown a node that has not yet been added to a cluster.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
option	string	Action to take for the node shutdown: • `restart`: Restarts the node. • `halt`: Performs full power-off of the node.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireShutdownResult	Empty.

Schedule actions

Schedule actions enable you to create and manage snapshot schedules on the storage system.

createSchedule

The createSchedule action enables you to schedule an automatic snapshot of a volume at a defined interval. You can use the created snapshot later as a backup or rollback to ensure the data on a volume or group of volumes is consistent for the point in time in which the snapshot was created.

Additional information

If you schedule a snapshot to run at a time period that is not divisible by 5 minutes, the snapshot will run at the next time period that is divisible by 5 minutes. For example, if you schedule a snapshot to run at 12:42:00 UTC, it will run at 12:45:00 UTC. You cannot schedule a snapshot to run at intervals of less than 5 minutes.

Snapshot creation depends on cluster fullness. See the *NetApp SolidFire Element OS API Reference Guide* for more information.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
schedule	SolidFire:SolidFireSchedule	An object containing information about how the snapshot should be created at each scheduled interval. See the CreateSchedule workflow for the input parameters that are gathered to create the schedule parameter.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCreateScheduleResult	Information about the newly created schedule.

getSchedule

The getSchedule action enables you to retrieve information about a scheduled snapshot.

Additional information

You can see information about a specific schedule if there are many snapshot schedules in the system. You also retrieve information about more than one schedule with this action by specifying additional scheduleIDs in the parameter.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
scheduleID	number	The unique ID of the schedule or schedules to display.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetScheduleResult	Information about the snapshot schedules.

listSchedules

The listSchedules action enables you to retrieve information about all scheduled snapshots that have been created.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListSchedulesResult	A list of the schedules currently on the cluster.

modifySchedule

The modifySchedule action enables you to change the intervals at which a scheduled snapshot occurs. This allows for adjustment to the snapshot frequency and retention.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
schedule	SolidFire:SolidFireSchedule	An object containing information about how the snapshot should be modified at each scheduled interval. See the ModifySchedule workflow for the input parameters that are gathered to create the schedule parameter.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireModifyScheduleResult	Information about the modified schedule attributes.

Sensor actions

Sensor actions enable you to view IPMI sensor information for cluster hardware.

getIpmiConfig

The getIpmiConfig action enables you to retrieve hardware sensor information from sensors that are in your node.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Input	Type	Description
chassisType	string	Used to display information for each node chassis type. Valid values: • `all`: returns sensor information for each chassis type. • `{chassis type}`: returns sensor information for a specified chassis type.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetIpmiConfigResult	Details of the sensor information.

getIpmiInfo

The getIpmiInfo action enables you to display a detailed reporting of sensors (objects) for node fans, intake and exhaust temperatures, and power supplies that are monitored by the system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetIpmiInfoResult	Detailed information from each sensor within a node.

Service actions

You can use the service actions to retrieve service information from the system.

listServices

The listServices action enables you to return the services information for nodes, drives, current software, and other services that are running on the cluster.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListServicesResult	Services that are running on drives and nodes.

restartServices

The restartServices action enables you to restart the services on a node.

Additional information

This method causes temporary node services interruption. Exercise caution when using this method.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
force	boolean	Include the "force" parameter with this action to successfully test the drives on the node.
service	string	The service name to be restarted.
action	string	The action to perform on the service (start, stop, restart).

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireAttributes	Details of the restart operation results.

Snapshot actions

Snapshot actions enable you to manage volume snapshots. You can create, modify, clone, and delete volume snapshots using these actions.

createGroupSnapshot

The createGroupSnapshot action enables you to create a point-in-time copy of a group of volumes. You can use this snapshot later as a backup or rollback to ensure the data on the group of volumes is consistent for the point in time that you created the snapshot.

Additional information

Snapshot creation depends on cluster fullness. See the *NetApp SolidFire Element OS API Reference Guide* for more information.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.
enableRemoteReplication	boolean	Values: • `true`: The snapshot will be replicated to remote storage. • `false`: Default. No replication.
name	string	The name entered for the group snapshot. If no name is entered, the date and time the group snapshot was taken is used.
retention	string	The amount of time the snapshot will be retained. Enter in HH:mm:ss.
volumes	Array of volumeIDs	The unique ID of the volume image from which to copy.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCreateGroupSnapshotResult	Information about the new group snapshot and its members.

createSnapshot

The createSnapshot action enables you to create a point-in-time copy of a volume. You can create a snapshot from any volume or from an existing snapshot.

Additional information

If you do not provide a snapshotID with this action, a snapshot is created from the volume's active branch. If the volume from which the snapshot is created is being replicated to a remote cluster, the snapshot can also be replicated to the same target. Use the `enableRemoteReplication` parameter to enable snapshot replication.

Snapshot creation depends on cluster fullness. See the *NetApp SolidFire Element OS API Reference Guide* for more information.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.
enableRemoteReplication	boolean	Possible values: • `true`: The snapshot will be replicated to remote storage. • `false`: The default. No replication.
name	string	The name entered for the group snapshot. If no name is entered, the date and time the group snapshot was taken is used.
retention	string	The amount of time the snapshot will be retained. Enter in HH:mm:ss.
snapshotID	number	The unique ID of a snapshot from which the new snapshot is made. The `snapshotID` passed must be a snapshot on the given volume.
volumeID	number	The unique ID of the volume image from which to copy.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCreateSnapshotResult	Information about and checksum data for the new volume snapshot.

deleteGroupSnapshot

The deleteGroupSnapshot action enables you to delete a group snapshot. You can use the saveMembers parameter to preserve all the snapshots that were made for the volumes in the group, but the group association will be removed.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
groupSnapshotID	number	The unique ID of the group snapshot.
saveMembers	boolean	`true`: Snapshots are kept, but group association is removed. `false`: The group and snapshots are deleted.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireDeleteGroupSnapshotResult	Empty.

deleteSnapshot

The deleteSnapshot action enables you to delete a snapshot.

Additional information

A snapshot that is currently the "active" snapshot cannot be deleted. You must roll back and make another snapshot "active" before the current snapshot can be deleted. For more details on rolling back snapshots, see the rollbackToSnapshot action.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
snapshotID	number	The ID of the snapshot to delete.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireDeleteSnapshotResult	Empty.

listGroupSnapshots

The listGroupSnapshots action enables you to retrieve information about all group snapshots that have been created.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumes	Array of volumeIDs	An array of unique volume IDs to query. If you do not specify this parameter, all group snapshots on the cluster are included.
groupSnapshotID	number	Retrieve information for an individual group snapshot ID.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListGroupSnapshotsResult	Group snapshot information.

listSnapshots

The listSnapshots action enables you to retrieve the attributes of each snapshot taken on the volume. When you use this action on the source cluster, you can retrieve information about snapshots that reside on the target cluster.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeID	number	Retrieve snapshots for a volume. If volumeID is not provided, all snapshots for all volumes are returned.
snapshotID	number	Retrieve information for an individual snapshot ID.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListSnapshotsResult	Information about each snapshot for each volume. If volumeID is not provided, all snapshots for all volumes are returned. Snapshots that are in a group are returned with a group ID.

modifyGroupSnapshot

The modifyGroupSnapshot action enables you to change the attributes of a group of snapshots. You can also use this action to enable snapshots created on the read/write (source) volume to be remotely replicated to a target SolidFire storage system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
enableRemoteReplication	boolean	Values: • `true`: The snapshot will be replicated to remote storage. • `false`: Default. No replication.
expirationTime	string	ISO 8601 date string format; use to set the time when the snapshot should be removed. If no time is entered, the current time will be used.
groupSnapshotID	string	ID of the group of snapshots.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireModifyGroupSnapshotResult	Information about the newly modified group snapshot.

modifySnapshot

The modifySnapshot action enables you to change the attributes currently assigned to a snapshot. You can use this method to enable snapshots created on the read/write (source) volume to be remotely replicated to a target SolidFire storage system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Input	Type	Description
enableRemoteReplication	boolean	Possible values: • `true`: The snapshot will be replicated to remote storage. • `false`: The default. No replication.
expirationTime	string	ISO 8601 date string format; use to set the time when the snapshot should be removed.
snapshotID	string	ID of the snapshot.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireModifySnapshotResult	Information about the newly modified snapshot.

rollbackToGroupSnapshot

The rollbackToGroupSnapshot action enables you to roll back all individual volumes in a snapshot group to each volume's individual snapshot.

Additional information

Rolling back to a group snapshot creates a temporary snapshot of each volume within the group snapshot.

Snapshot creation depends on cluster fullness. See the *NetApp SolidFire Element OS API Reference Guide* for more information.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
groupSnapshotID	number	The unique ID of the group snapshot.
name	string	Name for the group snapshot of the volume's current state that is created if `saveCurrentState = true`. If you do not give a name, then the name of the snapshots (group and individual volume) are set to a timestamp of the time that the rollback occurred.

Input	Type	Description
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.
saveCurrentState	boolean	`true`: The previous active volume image is kept. `false`: (default) The previous active volumeimage is deleted.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireRollbackToGroupSnapshotResult	Information about the newly created group snapshot.

rollbackToSnapshot

The rollbackToSnapshot action enables you to make an existing snapshot of the "active" volume image.

Additional information

This action creates a new snapshot from an existing snapshot. The new snapshot becomes "active" and the existing snapshot is preserved until it is manually deleted. The previously "active" snapshot is deleted unless you set `saveCurrentState = true`.

Snapshot creation depends on cluster fullness. See the *NetApp SolidFire Element OS API Reference Guide* for more information.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeID	number	VolumeID for the volume.
snapshotID	number	ID of a previously created snapshot on the given volume.
name	string	Name for the snapshot. If no name is given then the name of the snapshot being rolled back to is used with "- copy" appended to the end of the name.
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.

Input	Type	Description
saveCurrentState	boolean	`true`: The previous active volume image is kept. `false`: (default) The previous active volumeimage is deleted.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireRollbackToSnapshotResult	Information about the newly created snapshot.

SNMP actions

SNMP actions enable you to enable, configure, disable, and test SNMP functionality on a cluster.

getSnmpACL

The getSnmpACL action enables you to retrieve the current SNMP access permissions on the cluster nodes.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetSnmpACLResult	Information about the permissions of networks and users in the SNMP environment.

getSnmpInfo

The getSnmpInfo action enables you to retrieve the current simple network management protocol (SNMP) configuration information.

Additional information

The getSnmpInfo action is deprecated for versions later than Element OS version 8.0. The GetSnmpState and SetSnmpACL methods replace the GetSnmpInfo method.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetSnmpInfoResult	Information about the networks, access types, and users associated with for SNMP.

getSnmpState

The getSnmpState action enables you to retrieve the current state of the SNMP feature.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetSnmpStateResult	Information about the state of the SNMP feature.

getSnmpState

The getSnmpState action enables you to retrieve the current state of the SNMP feature.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetSnmpStateResult	Information about the state of the SNMP feature.

getSnmpTrapInfo

The getSnmpTrapInfo action enables you to retrieve the current SNMP trap configuration information.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetSnmpTrapInfoResult	Information about the current SNMP trap configuration information.

snmpSendTestTraps

The snmpSendTestTraps action enables you to test SNMP functionality for a cluster. This action instructs the cluster to send test SNMP traps to the currently configured SNMP manager.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireSnmpSendTestTrapsResult	Status of the test.

Storage container actions

Storage container actions enable you to create, remove, list, and change storage containers in the system.

createStorageContainer

The createStorageContainer action enables you to create a Virtual Volume (VVol) storage container.

Additional information

Storage containers are associated with a SolidFire storage system account, and are used for reporting and resource allocation. Storage containers can only be associated with virtual volumes. You need at least one storage container to use the Virtual Volumes feature.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
name	string	Name of the storage container. Follows SolidFire account naming restrictions.
accountID	number	Non-storage container account that will become a storage container.
initiatorSecret	string	The secret for CHAP authentication for the initiator.
targetSecret	string	The secret for CHAP authentication for the target.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCreateStorageContainerResult	Information about the newly created storage container.

deleteStorageContainers

The deleteStorageContainers action enables you to remove up to 2000 Virtual Volume (VVol) storage containers from the system at one time. The storage containers you remove must not contain any VVols.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
storageContainerIDs	Array of string	A list of IDs of the storage containers to delete. You can specify up to 2000 IDs in the list.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireDeleteStorageContainersResult	Empty.

getStorageContainerEfficiency

The getStorageContainerEfficiency action enables you to retrieve efficiency information about a virtual volume storage container.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
storageContainerID	number	The ID of the storage container for which to retrieve efficiency information

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetStorageContainerEfficiencyResult	Storage container efficiency information.

listStorageContainers

The listStorageContainers action enables you to retrieve information about all virtual volume storage containers known to the system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
storageContainerIDs	Array of string	A list of storage container IDs for which to retrieve information. If you omit this parameter, the method returns information about all storage containers in the system.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListStorageContainersResult	Information about all volume storage containers in the system.

modifyStorageContainer

The modifyStorageContainer action enables you to make changes to an existing virtual volume storage container.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
storageContainerID	string	The unique ID of the virtual volume storage container to modify.
initiatorSecret	string	The new secret for CHAP authentication for the initiator.
targetSecret	string	The new secret for CHAP authentication for the target.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFiremodifyStorageContainerResult	Information about the newly modified storage container.

Test actions

Test actions enable you to test network connectivity between nodes and to other clusters.

listTests

The listTests action enables you to retrieve the tests that are available to run on a node.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListTestsResult	List of tests that can be performed on the node.

listUtilities

The listUtilities workflow enables you to retrieve the operations that are available to run on a node.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListUtilitiesResult	List of utilities currently available to run on the node.

testConnectEnsemble

The testConnectEnsemble workflow enables you to add an account to a system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
ensemble	string	A comma-separated list of ensemble node cluster IP addresses for connectivity testing.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireTestConnectEnsembleResult	The results of the ensemble connectivity test.

testConnectMvip

The testConnectMvip action enables you to test the management connection to the cluster. The test pings the MVIP and executes a simple API method to verify connectivity.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Input	Type	Description
mvip	string	Optional. Use to test the management connection of a different MVIP. You do not need to use this value when testing the connection to the target cluster.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireTestConnectMvipResult	The results of the MVIP connectivity test.

TestConnectSvip

The TestConnectSvip workflow enables you to add an account to a system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
svip	string	Optional. Use to test the management connection of a different SVIP. You do not need to use this value when testing the connection to the target cluster.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireTestConnectSvipResult	The results of the SVIP connectivity test.

testPing

The testPing action enables you to test network connectivity to all nodes in the cluster on both 1G and 10G interfaces using ICMP packets. The test uses the appropriate MTU sizes for each packet based on the MTU settings in the network configuration.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Input	Type	Description
attempts	string	Optional. Use to test the management connection of a different SVIP. You do not need to use this value when testing the connection to the target cluster.
hosts	string	Specify a comma-separated list of addresses or hostnames of devices to ping.
totalTimeoutSec	number	Specifies the length of time the ping should wait for a system response before issuing the next ping attempt or ending the process.
packetSize	number	Specify the number of bytes to send in the ICMP packet sent to each IP. Number must be less than the maximum MTU specified in the network configuration.
pingTimeoutMsec	number	Specify the number of milliseconds to wait for each individual ping response.
prohibitFragmentation	boolean	Enable the DF (Do not Fragment) flag for the ICMP packets.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireTestPingResult	A list of each IP address the node was able to communicate with.

Virtual network actions

Virtual network actions enable you to add, view, change, and remove virtual networks in the system.

addVirtualNetwork

The addVirtualNetwork action enables you to add a new virtual network to a cluster configuration.

Additional information

When you add a virtual network, an interface for each node is created and each interface will require a virtual network IP address. The number of IP addresses you specify as a parameter for this action must be equal to or greater than the number of nodes in the cluster. The system bulk provisions virtual network addresses and assigns them to individual nodes automatically. You do not need to assign virtual network addresses to nodes manually.

The addVirtualNetwork action is used only to create a new virtual network. If you want to make changes to an existing virtual network, use the modifyVirtualNetwork action.

Note that virtual network parameters must be unique to each virtual network when using `namespace=false`.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
addressBlocks	Array of SolidFire:SolidFireAddressBlockParams	Attributes for this parameter are: • `start`: (string) The start of the IP address range. • `size`: (number) The number of IP addresses to include in the block.
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.
gateway	string	The name entered for the group snapshot. If no name is entered, the date and time the group snapshot was taken is used.
name	string	The amount of time the snapshot will be retained. Enter in HH:mm:ss.
namespace	boolean	The unique ID of a snapshot from which the new snapshot is made. The snapshotID passed must be a snapshot on the given volume.
netmask	string	The unique ID of the volume image from which to copy.
svip	string	Unique storage IP address for the virtual network being created.
virtualNetworkTag	number	A unique virtual network (VLAN) tag. Supported values are 1 to 4094 (the number zero (0) is not supported).

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireAddVirtualNetworkResult	The virtual network ID of the new virtual network.

listVirtualNetworks

The listVirtualNetworks action enables you to list all configured virtual networks for the cluster.

Additional information

You can use this action to verify the virtual network settings in the cluster. There are no required parameters for this action. However, to filter the results, you can pass one or more virtualNetworkID or virtualNetworkTag values.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
virtualNetworkID	number	Network ID to filter the list for a single virtual network.
virtualNetworkTag	number	Network tag to filter the list for a single virtual network.
virtualNetworkIDs	Array of virtualNetworkIDs	Network IDs to include in the list.
virtualNetworkTags	Array of virtualNetworkTags	Network tags to include in the list.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListVirtualNetworksResult	All virtual network IP addresses for the cluster.

modifyVirtualNetwork

The modifyVirtualNetwork action enables you to change the attributes of an existing virtual network. This action enables you to add or remove address blocks, change the netmask, or modify the name or description of the virtual network. You can also use it to enable or disable namespaces, as well as add or remove a gateway if namespaces are enabled on the virtual network.

Additional information

You can use this action to verify the virtual network settings in the cluster. There are no required parameters for this action. However, to filter the results, you can pass one or more virtualNetworkID or virtualNetworkTag values.

Note: This action requires either the virtualNetworkID or the virtualNetworkTag as a parameter, but not both.

Caution: Enabling or disabling the Routable Storage VLANs functionality for an existing virtual network by changing the "namespace" parameter disrupts any traffic handled by the virtual network. NetApp strongly recommends only changing the "namespace" parameter during a scheduled maintenance window.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
virtualNetworkID	number	Unique identifier of the virtual network to modify. This is the virtual network ID assigned by the cluster. This parameter is optional but either virtualNetworkID or virtualNetworkTag must be specified with this action.
virtualNetworkTag	number	The network tag that identifies the virtual network to modify. This parameter is optional but either virtualNetworkID or virtualNetworkTag must be specified with this action.
name	string	The new name for the virtual network.
addressBlocks	Array of SolidFire:SolidFireAddressBlockParams	The new addressBlock to set for this virtual network. This may contain new address blocks to add to the existing object or it may omit unused address blocks that need to be removed. Alternatively, you can extend or reduce the size of existing address blocks. You can only increase the size of the starting addressBlocks for a virtual network object; you can never decrease it. Attributes for this parameter are: • `start`: (string) The start of the IP address range. • `size`: (number) The number of IP addresses to include in the block.
gateway	string	The IP address of a gateway of the virtual network. This parameter is only valid if `namespace=true`.

Input	Type	Description
namespace	boolean	When set to `true`, enables Routable Storage VLANs functionality by recreating the virtual network and configuring a namespace to contain it. When set to `false`, disables the VRF functionality for the virtual network. Changing this value disrupts traffic running through this virtual network.
netmask	string	The new network mask for this virtual network.
svip	string	The storage virtual IP address for this virtual network. The svip for a virtual network cannot be changed. You must create a new virtual network to use a different svip address.
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireAddVirtualNetworkResult	Empty.

removeVirtualNetwork

The removeVirtualNetwork action enables you to remove a previously added virtual network.

Additional information

Note: This action requires either the virtualNetworkID or the virtualNetworkTag as a parameter, but not both.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
virtualNetworkID	number	Network ID that identifies the virtual network to remove.
virtualNetworkTag	number	Network tag that identifies the virtual network to remove.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireRemoveVirtualNetworkResult	Empty.

Virtual volume actions

Virtual volume actions enable you to retrieve information about virtual volumes in the system.

enableFeature

The enableFeature action enables you to enable cluster features that are disabled by default.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
feature	string	Valid values: `vvols`: Enable the NetApp SolidFire VVols cluster feature.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireEnableFeatureResult	Empty.

listProtocolEndpoints

The listProtocolEndpoints action enables you to retrieve information about all protocol endpoints in the cluster. Protocol endpoints govern access to their associated virtual volume storage containers.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
protocolEndpointIDs	Array of string	A list of protocol endpoint IDs for which to retrieve information. If you omit this parameter, the method returns information about all protocol endpoints.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListProtocolEndpointsResult	A list of objects containing information about each protocol endpoint inthe system.

getFeatureStatus

The getFeatureStatus action enables you to retrieve the status of a cluster feature.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
feature	string	Valid values: `vvols`: Retrieve status for the NetApp SolidFire VVols cluster feature.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetFeatureStatusResult	An array of feature objects indicating the feature name and its status.

listVolumeStatsByVirtualVolume

The listVolumeStatsByVirtualVolume enables you to list volume statistics for any volumes in the system that are associated with virtual volumes. Statistics are cumulative from the creation of the volume.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
virtualVolumeIDs	Array of string	A list of one or more virtual volume IDs for which to retrieve information. If you specify this parameter, the method returns information about only these virtual volumes.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListVolumeStatsByVirtualVolumeResult	Activity information for each volume associated with a virtual volume.

getVirtualVolumeCount

The getVirtualVolumeCount action enables you to retrieve the number of virtual volumes currently in the system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetVirtualVolumeCountResult	The number of virtual volumes currently in the system.

listVirtualVolumeBindings

The listVirtualVolumeBindings action enables you to retrieve a list of all virtual volumes in the cluster that are bound to protocol endpoints.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
virtualVolumeBindingIDs	Array of integers	A list of virtual volume binding IDs for which to retrieve information. If you omit this parameter, the action returns information about all virtual volume bindings.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListVirtualVolumeBindingsResult	A list of objects describing all virtual volumes in the cluster that are bound to protocol endpoints.

listVirtualVolumeHosts

The listVirtualVolumeHosts action enables you to retrieve a list of all virtual volume hosts known to the cluster. A virtual volume host is a VMware ESX® host that has initiated a session with the VASA API provider.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
virtualVolumeHostIDs	Array of string	A list of virtual volume host IDs for which to retrieve information. If you omit this parameter, the action returns information about all virtual volume hosts.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListVirtualVolumeHostsResult	A list of objects describing the virtual volume hosts in the cluster.

listVirtualVolumes

The listVirtualVolumes action enables you to list the virtual volumes currently in the system. You can use this action to list all virtual volumes, or only list a subset.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
details	boolean	Possible values: • `true`: Include more details about each VVol in the response. • `false`: Include the standard level of detail about each VVol inthe response.
limit	number	The maximum number of virtual volumes to list.

Input	Type	Description
recursive	boolean	Possible values: • `true`: Include information about the children of each VVol in the response. • `false`: Do not include information about the children of each VVol in the response.
startVirtualVolumeID	string	The ID of the virtual volume at which to begin the list.
virtualVolumeIDs	Array of string	A list of virtual volume IDs for which to retrieve information. If you specify this parameter, the action returns information about only these virtual volumes.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListVirtualVolumesResult	A list of objects describing the virtual volumes currently in the system.

listVirtualVolumeTasks

The listVirtualVolumeTasks action enables you to retrieve a list of virtual volume tasks in the system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
virtualVolumeTaskIDs	Array of string	A list of virtual volume task IDs for which to retrieve information. If you omit this parameter, the action returns information about all virtual volume tasks.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListVirtualVolumeTasksResult	A list of objects describing the virtual volume tasks currently in the system.

Volume actions

Volume actions enable you to add, remove, view, clone, and configure volumes in the system.

addVolumesToVolumeAccessGroup

The addVolumesToVolumeAccessGroup action enables you to add volumes to a specified volume access group.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumes	Array of number	List of volumeIDs to add to the volume access group.
volumeAccessGroupID	number	The volumeAccessGroupID of the volume access group to which volumes are added.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireAddVolumesToVolumeAccessGroupResult	Information about the newly modified volume access group.

cancelClone

The cancelClone action enables you to stop an ongoing cloneVolume or copyVolume process. When you cancel a group clone operation, the system completes and removes the operation's associated asyncHandle.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
cloneID	number	The cloneID for the ongoing clone process.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCancelCloneResult	Empty.

AddAccount

The cancelGroupClone action enables you to stop an ongoing cloneMultipleVolumes process occurring on a group of volumes. When you cancel a group clone operation, the system completes and removes the operation's associated asyncHandle.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
groupcloneID	number	The cloneID for the ongoing group clone process.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCancelGroupCloneResult	Empty.

cloneMultipleVolumes

The cloneMultipleVolumes action enables you to create a clone of a group of specified volumes.

Additional information

You can assign a consistent set of characteristics to a group of multiple volumes when they are cloned together. Before using `groupSnapshotID` to clone the volumes in a group snapshot, you must first create the group snapshot using the createGroupSnapshot action or the Web UI. Using `groupSnapshotID` is optional when cloning multiple volumes.

Snapshot creation depends on cluster fullness. See the *NetApp SolidFire Element OS API Reference Guide* for more information.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Input	Type	Description
volumes	Array of SolidFire:SolidFireCloneMultipleVolumeParams	The unique ID for each volume to include in the clone. If optional parameters are not specified, the values are inherited from the source volumes. Required parameter for `volumes` array: `volumeID` Optional parameters for `volumes` array: • `access`: Can be one of "readOnly", "readWrite", "locked", or "replicationTarget". • `attributes`: A list of name-value pairs in string format, parseable into a JSON object. • `name`: The new name for the clone. • `newAccountID`: The account ID for the new volumes. • `newSize`: The new total size of the volume, in bytes. Size is rounded up to the nearest megabyte.
access	string	The new default access method for the new volumes if not overridden by information passed in the `volumes` array.
groupSnapshotID	number	The ID of the group snapshot to use as a basis for the clone.
newAccountID	number	The new account ID for the volumes if not overridden by information passed in the `volumes` array.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCloneMultipleVolumesResult	Information about the new group clone.

cloneVolume

The cloneVolume action enables you to create a copy of a volume.

Additional information

This action is asynchronous and may take a variable amount of time to complete. The cloning process begins immediately when you make the cloneVolume request and is representative of the state of the volume when the API method is issued. You can use the getAsyncResult action to determine when the cloning process is complete and the new volume is available for connections. You can use listSyncJobs to see the progress of creating the clone.

The initial attributes and quality of service settings for the volume are inherited from the volume being cloned. You can change these settings with the modifyVolume action.

Note: Cloned volumes do not inherit volume access group membership(s) from the source volume.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeID	number	The ID of the volume to be cloned.
name	string	Name of the new cloned volume. May be 1 to 64 characters in length.
newAccountID	number	The accountID for the owner of the new volume. If not specified, the accountID of the owner of the volume being cloned is used.
newSize	number	New size of the volume, in bytes. May be greater or less than the size of the volume being cloned. If not specified, the volume size is not changed. Size is rounded up to the nearest megabyte.

Input	Type	Description
access	string	The access allowed for the new volume. Possible values: • `readOnly`: Only read operations are allowed. • `readWrite`: Reads and writes are allowed. • `locked`: No reads or writes are allowed. If not specified, the access value of the volume being cloned is used. • `replicationTarget`: Identify a volume as the target volume for a paired set of volumes. If the volume is not paired, the access status is locked. If a value is not specified, the access value does not change.
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.
snapshotID	number	ID of the snapshot that is used as the source of the clone. If no ID is provided, the current active volume is used.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCloneVolumeResult	Information about the new volume clone.

copyVolume

The copyVolume action enables you to overwrite the data contents of an existing volume with the data contents of another volume (or snapshot). Attributes of the destination volume such as IQN, QoS settings, size, account, and volume access group membership are not changed.

Additional information

The destination volume must already exist and must be the same size as the source volume. It is recommended that clients unmount the destination volume before the copy operation begins. If the destination volume is modified during the copy operation, the changes will be lost. This method is asynchronous and may take a variable amount of time to complete. You can use the getAsyncResult action to determine when the process has finished, and listSyncJobs to see the progress of the copy.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
dstVolumeID	number	The volumeID of the volume to overwrite.
volumeID	number	The volumeID of the volume to be read from.
snapshotID	number	The ID of the snapshot that is used as the source of the clone. If no ID is provided, the current active volume is used.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCopyVolumeResult	Information about the new volume copy.

createVolume

The createVolume action enables you to create a new (empty) volume on the cluster. As soon as the volume creation is complete, the volume is available for connection via iSCSI.

Additional information

Volumes created without specified QoS values use the default values. You can view default values for a volume by using the getDefaultQoS action.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
name	string	Name of the new volume (may be user- specified). Not required to be unique, but recommended. May be 1 to 64 characters in length.
accountID	number	AccountID for the owner of this volume.
totalSize	number	Total size of the volume, in bytes. Size is rounded up to the nearest megabyte.
enable512e	boolean	`true`: The volume provides 512-byte sector emulation. `false`: 512e emulation is not enabled.

Input	Type	Description
attributes	string	A list of name-value pairs in string format, parseable into a JSON object. Total attribute size must be less than 1000B, or 1KB, including JSON formatting characters.
qos	SolidFire:SolidFireQoS	Initial quality of service settings for this volume. Default values are used if none are specified. Valid settings are: • `minIOPS` • `maxIOPS` • `burstIOPS`

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCreateVolumeResult	Information about the newly created volume.

deleteVolume

The deleteVolume action enables you to mark an active volume for deletion. Once marked, the volume is purged (permanently deleted) after the cleanup interval elapses.

Additional information

After making a request to delete a volume, any active iSCSI connections to the volume are immediately terminated and no further connections are allowed while the volume is in this state. A marked volume is not returned in target discovery requests.

Any snapshots of a volume that has been marked for deletion are not affected. Snapshots are kept until the volume is purged from the system.

If a volume is marked for deletion and has a bulk volume read or bulk volume write operation in progress, the bulk volume read or write operation is stopped.

If the volume you delete is paired with a volume, replication between the paired volumes is suspended and no data is transferred to it or from it while in a deleted state. The remote volume the deleted volume was paired with enters into a PausedMisconfigured state and data is no longer sent to it or from the deleted volume. Until the deleted volume is purged, it can be restored and data transfers resume. If the deleted volume gets purged from the system, the volume it was paired with enters into a StoppedMisconfigured state and the volume pairing status is removed. The purged volume becomes permanently unavailable.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Input	Type	Description
volumeID	number	The ID of the volume to delete.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireDeleteVolumeResult	Information about the deleted volume.

deleteVolumes

The deleteVolumes action marks multiple (up to 500) active volumes for deletion.

Additional information

Once marked, the volumes are purged (permanently deleted) after the cleanup interval elapses. After making a request to delete volumes, any active iSCSI connections to the volumes are immediately terminated and no further connections are allowed while the volumes are in this state. A marked volume is not

returned in target discovery requests.

Any snapshots of a volume that has been marked for deletion are not affected. Snapshots are kept until the volume is purged from the system.

If a volume is marked for deletion and has a bulk volume read or bulk volume write operation in progress, the bulk volume read or write operation is stopped.

If the volumes you delete are paired with a volume, replication between the paired volumes is suspended and no data is transferred to them or from them while in a deleted state. The remote volumes the deleted volumes were paired with enter into a PausedMisconfigured state and data is no longer sent to them or from the deleted volumes. Until the deleted volumes are purged, they can be restored and data transfers resume. If the deleted volumes are purged from the system, the volumes they were paired with enter into a StoppedMisconfigured state and the volume pairing status is removed. The purged volumes become permanently unavailable.

Input parameters

This action has the following input parameters. At least one of the following parameters are required, and you must use only one of the parameters (they are all mutually exclusive with one another):

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
accountIDs	Array of number	A list of account IDs. All volumes from these accounts are deleted from the system.
volumeAccessGroupIDs	Array of number	A list of volume access group IDs. All of the volumes from all of the volume access groups you specify in this list are deleted from the system.
volumeIDs	Array of number	The ID of the volume to delete.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireDeleteVolumesResult	Information about the deleted volumes.

getDefaultQoS

The getDefaultQoS action enables you to retrieve the default QoS values for a newly created volume.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireVolumeQOS	The default QoS values for volumes.

getVolumeCount

The getVolumeCount action enables you to retrieve the number of volumes currently in the system..

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetVolumeCountResult	The number of volumes currently in the system.

getVolumeEfficiency

The getVolumeEfficiency action enables you to retrieve information about a volume. Only the volume you give as a parameter in this action is used to compute the capacity.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Input	Type	Description
volumeID	number	Specifies the volume for which capacity is computed.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetVolumeEfficiencyResult	Efficiency information for the volume.

getVolumeStats

The getVolumeStats action enables you to retrieve high-level activity measurements for a single volume. Values are cumulative from the creation of the volume.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeID	number	Specifies the volume for which statistics are gathered.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetVolumeStatsResult	Volume activity information.

listActiveVolumes

The listActiveVolumes action enables you to retrieve the list of active volumes currently in the system. The list of volumes is returned sorted in volumeID order and can be returned in multiple parts (pages).

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
includeVirtualVolumes	boolean	Virtual volumes are included in the response by default. To exclude virtual volumes, set to `false`.

Input	Type	Description
startVolumeID	number	Starting volumeID to return. If no volume exists with this volumeID, the next volume by volumeID order is used as the start of the list. To page through the list, pass the volumeID of the last volume in the previous response + 1.
limit	number	Maximum number of volume objects to return. 0 (zero) returns all volumes (unlimited).

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListActiveVolumesResult	The list of active volumes.

istBulkVolumeJobs

The listBulkVolumeJobs action enables you to retrieve information about each bulk volume read or write operation that is occurring in the system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListBulkVolumeJobsResult	An array of information for each bulk volume job.

listDeletedVolumes

The listDeletedVolumes action enables you to retrieve the list of volumes that have been marked for deletion and purged from the system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.

Input	Type	Description
includeVirtualVolumes	boolean	Virtual volumes are included in the response by default. To exclude virtual volumes, set to `false`.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListDeletedVolumesResult	A list of deleted volumes.

listVolumes

The listVolumes action enables you to retrieve a list of volumes that are in a cluster. You can specify the volumes you want to return in the list by using the available parameters.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
accounts	boolean	Only volumes owned by the accounts you specify here are returned. Mutually exclusive with the `volumeIDs` parameter.
includeVirtualVolumes	boolean	Virtual volumes are included in the response by default. To exclude virtual volumes, set to `false`.
isPaired	boolean	Returns volumes that are paired or not paired. Possible values: • `true`: Returns all paired volumes. • `false`: Returns all volumes not paired.
limit	number	Enables you to set the maximum number of volume results that are returned. Mutually exclusive with the `volumeIDs` parameter.
startVolumeID	number	Only volumes with an ID greater than or equal to this value are returned. Mutually exclusive with the `volumeIDs` parameter.

Input	Type	Description
volumeIDs	Array of integers	A list of volume IDs. If you supply this parameter, other parameters operate only on this set of volumes. Mutually exclusive with the `accounts`, `startVolumeID`, and `limit` parameters.
volumeName	string	Only volume object information matching the volume name is returned.
volumeStatus	string	Only volumes with the specified status are returned. Possible values: • `creating` • `snapshotting` • `active` • `deleted`

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListVolumesResult	A list of volumes.

listVolumesForAccount

The listVolumesForAccount action enables you to retrieve the list of active and (pending) deleted volumes for an account.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
includeVirtualVolumes	boolean	Virtual volumes are included in the response by default. To exclude virtual volumes, set to `false`.
accountID	number	All volumes owned by this account ID are returned.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListVolumesForAccountResult	A list of volume information.

listVolumeStats

The listVolumeStats action enables you to retrieve high-level activity measurements for a single volume, list of volumes, or all volumes (if you omit the volumeIDs parameter). Measurement values are cumulative from the creation of the volume.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
includeVirtualVolumes	boolean	Virtual Volumes are included in the response by default. To exclude virtual volumes, set `includeVirtualVolumes=false`.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListVolumeStatsResult	Volume activity information.

listVolumeStatsByAccount

The listVolumeStatsByAccount action returns high-level volume activity measurements for every account. Values are summed from all volumes owned by the account.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
includeVirtualVolumes	boolean	Virtual volumes are included in the response by default. To exclude virtual volumes, set to `false`.
accounts	Array of integers	A list of account ID for which to return volume statistics. If omitted, statistics for all accounts are returned.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListVolumeStatsByAccountResult	List of volume activity information for each account.

listVolumeStatsByVolume

The listVolumeStatsByVolume action enables you to retrieve high-level activity measurements for every volume, by volume. Values are cumulative from the creation of the volume.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
includeVirtualVolumes	boolean	Virtual volumes are included in the response by default. To exclude virtual volumes, set to `false`.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListVolumeStatsByVolumeResult	A list of volume activity information.

listVolumeStatsByVolumeAccessGroup

The listVolumeStatsByVolumeAccessGroup action enables you to get total activity measurements for all of the volumes that are a member of the specified volume access group(s).

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
includeVirtualVolumes	boolean	Virtual volumes are included in the response by default. To exclude virtual volumes, set to `false`.
volumeAccessGroups	Array of integers	An array of volumeAccessGroupIDs for which volume activity is returned. If omitted, statistics for all volume access groups are returned.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListVolumeStatsByVolumeAccessGroupResult	A list of volume activity information for all volumes in the specified volume access group.

modifyVolume

The modifyVolume action enables you to modify settings on an existing volume. You can make modifications to one volume at a time and changes take place immediately.

Additional information

If you do not specify QoS values when you modify a volume, they remain the same as before the modification. You can retrieve default QoS values for a newly created volume using the getDefaultQoS action.

When you need to increase the size of a volume that is being replicated, do so in the following order to prevent replication errors:

1. Increase the size of the "Replication Target" volume.
2. Increase the size of the source or "Read / Write" volume.

Ensure that both the target and source volumes are the same size.

Note: If you change the `access` status to `locked` or `target`, all existing iSCSI connections are terminated.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeID	number	The volume ID of the volume to be modified.
accountID	number	The account ID to which the volume is reassigned. If none is specified, the previous account name is used.

Input	Type	Description
access	string	Access allowed for the volume. Possible values: • `readOnly`: Only read operations are allowed. • `readWrite`: Reads and writes are allowed. • `locked`: No reads or writes are allowed. If not specified, the access value does not change. • `replicationTarget`: Identify a volume as the target volume for a paired set of volumes. If the volume is not paired, the access status is `locked`. If a value is not specified, the access value does not change.
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.
qos	SolidFire:SolidFireQoS	New quality of service settings for this volume. If not specified, the QoS settings are not changed.
setCreateTime	boolean	Set to `true` to change the recorded date of volume creation.
createTime	string	An ISO 8601 date string to set as the new volume creation date. Required if `setCreateTime` is set to `true`.
totalSize	number	New size of the volume in bytes. 1000000000 is equal to 1GB. Size is rounded up to the nearest megabyte. This parameter can only be used to increase the size of a volume.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireModifyVolumeResult	Information about the newly modified volume.

modifyVolumes

The modifyVolumes action enables you to configure up to 500 existing volumes at one time. Changes take place immediately. If modifyVolumes fails to modify any of the specified volumes, none of the specified volumes are changed.

Additional information

If you do not specify QoS values when you modify volumes, the QoS values for each volume remain unchanged. You can retrieve default QoS values for a newly created volume using the getDefaultQoS action.

When you need to increase the size of volumes that is being replicated, do so in the following order to prevent replication errors:

1. Increase the size of the "Replication Target" volume.
2. Increase the size of the source or "Read / Write" volume.

Ensure that both the target and source volumes are the same size.

Note: If you change the `access` status to `locked` or `target`, all existing iSCSI connections are terminated.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeIDs	Array of integers	A list of volumeIDs for the volumes to be modified.
accountID	number	The account ID to which the volumes are reassigned. If none is specified, the previous account name is used.

Input	Type	Description
access	string	Access allowed for the volumes. Possible values: • `readOnly`: Only read operations are allowed. • `readWrite`: Reads and writes are allowed. • `locked`: No reads or writes are allowed. If not specified, the access value does not change. • `replicationTarget`: Identify a volume as the target volume for a paired set of volumes. If the volume is not paired, the access status is `locked`. If a value is not specified, the access value does not change.
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.
qos	SolidFire:SolidFireQoS	New quality of service settings for the volumes. If not specified, the QoS settings are not changed.
setCreateTime	boolean	Set to `true` to change the recorded date of volume creation.
createTime	string	An ISO 8601 date string to set as the new volume creation date. Required if `setCreateTime` is set to `true`.
totalSize	number	New size of the volumes in bytes. 1000000000 is equal to 1GB. Size is rounded up to the nearest megabyte. This parameter can only be used to increase the size of a volume.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireModifyVolumesResult	Information about the newly modified volumes.

purgeDeletedVolume

The purgeDeletedVolume action enables you to immediately and permanently purge a volume that has been deleted.

Additional information

You must delete a volume using deleteVolume before it can be purged. Volumes are purged automatically after a period of time, so usage of this method is not typically required.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeID	number	The volume ID of the volume to be purged.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFirePurgeDeletedVolumeResult	Empty.

purgeDeletedVolumes

The purgeDeletedVolumes immediately and permanently purges volumes that have been deleted; you can use this method to purge up to 500 volumes at one time.

Additional information

You must delete volumes using deleteVolume or deleveVolumes before they can be purged. Volumes are purged automatically after a period of time, so usage of this method is not typically required.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeIDs	Array of integers	A list of volumeIDs of volumes to be purged from the system.
accountIDs	Array of integers	A list of accountIDs. All of the volumes from all of the specified accounts are purged from the system.
volumeAccessGroupIDs	Array of integers	A list of volumeAccessGroupIDs. All of the volumes from all of the specified volume access groups are purged from the system.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFirePurgeDeletedVolumesResult	Empty.

removeVolumesFromVolumeAccessGroup

The removeVolumesFromVolumeAccessGroup action enables you to remove volumes from a specified volume access group.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeAccessGroupID	number	The volumeAccessGroupID of the volume access group from which volumes are removed.
volumes	Array of integers	A list of volumeIDs of volumes to remove from the volume access group.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireRemoveVolumesFromVolumeAccessGroup Result	Information about the newly modified volume access group.

restoreDeletedVolume

The restoreDeletedVolume action enables you to mark a deleted volume as active again. This action makes the volume immediately available for iSCSI connection.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeID	number	VolumeID of the deleted volume to restore.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireRestoreDeletedVolumeResult	Empty.

setDefaultQoS

The setDefaultQoS action enables you to configure the default Quality of Service (QoS) values (measured in inputs and outputs per second, or IOPS) for a volume.

Additional information

For more information on QoS in a SolidFire cluster, see the *NetApp SolidFire Element OS User Guide*.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
minIOPS	number	The minimum number of sustained IOPS that are provided by the cluster to a volume.
maxIOPS	number	The maximum number of sustained IOPS that are provided by the cluster to a volume.
burstIOPS	number	The maximum number of IOPS allowed in a short burst scenario.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireSetDefaultQoSResult	The new QoS values.

startBulkVolumeRead

The startBulkVolumeRead action enables you to initialize a bulk volume read session on a specified volume.

Additional information

Only two bulk volume processes can run simultaneously on a volume. When you initialize the session, data is read from a SolidFire storage volume for the purposes of storing the data on an external backup source. The external data is accessed by a web server running on an SF-series node. Communications and server interaction information for external data access is passed by a script running on the storage system.

At the start of a bulk volume read operation, a snapshot of the volume is made and the snapshot is deleted when the read has completed. You can also read a snapshot of the volume by entering the ID of the snapshot as a parameter. When you read a previous snapshot, the system does not create a new snapshot of the volume, nor does it delete the previous snapshot when the read completes.

This action creates a new snapshot if the ID of an existing snapshot is not provided. Snapshot creation depends on cluster fullness. See the *NetApp SolidFire Element OS API Reference Guide* for more information.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeID	number	The volumeID of the volume to be read.
format	string	The format of the volume data. Can be one of the following: • `uncompressed`: Every byte of the volume is returned without any compression. • `native`: Opaque data is returned that is smaller and more efficiently stored and written on a subsequent bulk volume write.
snapshotID	number	ID of a previously created snapshot used for bulk volume reads. If no ID is entered, a snapshot of the current active volume image is made.
script	string	Executable name of a script. If no script name is given then the key and URL is necessary to access SF-series nodes. The script is run on the primary node and the key and URL is returned to the script so the local web server can be contacted.
scriptParameters	string	A list of parameters to pass to the script; parseable into JSON object format.
attributes	string	A list of name-value pair attributes for the bulk volume job in string format, parseable into a JSON object.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireStartBulkVolumeReadResult	Information about the bulk volume read process.

startBulkVolumeWrite

The startBulkVolumeWrite action enables you to initialize a bulk volume write session on a specified volume.

Additional information

Only two bulk volume processes can run simultaneously on a volume. When you initialize the write session, data is written to a SolidFire storage volume from an external backup source. The external data is accessed by a web server running on an SF-series node. Communications and server interaction information for external data access is passed by a script running on the storage system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeID	number	The volumeID of the volume to be written to.
format	string	The format of the volume data. Can be one of the following: • `uncompressed`: Every byte of the source volume is written without any compression. • `native`: Opaque data is written that is smaller and more efficiently stored.
script	string	Executable name of a script. If no script name is given then the key and URL is necessary to access SF-series nodes. The script is run on the primary node and the key and URL is returned to the script so the local web server can be contacted.
scriptParameters	string	A list of parameters to pass to the script; parseable into JSON object format.
attributes	string	A list of name-value pair attributes for the bulk volume job in string format, parseable into a JSON object.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireStartBulkVolumeWriteResult	Information about the bulk volume write process.

updateBulkVolumeStatus

You can use the updateBulkVolumeStatus action in a script to update the status of a bulk volume job that you have started with the startBulkVolumeRead or startBulkVolumeWrite actions.

Additional information

Only two bulk volume processes can run simultaneously on a volume. When you initialize the write session, data is written to a SolidFire storage volume from an external backup source. The external data is accessed by a web server running on an SF-series node. Communications and server interaction information for external data access is passed by a script running on the storage system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
key	string	The key assigned during initialization of a startBulkVolumeRead or startBulkVolumeWrite session.
status	string	The system sets the status of the given bulk volume job. Possible values: • `running`: Jobs that are still active. • `complete`: Jobs that are done. • `failed`: Jobs that have failed.
percentComplete	string	The completed progress of the bulk volume job as a percentage.
message	string	Returns the status of the bulk volume job when the job has completed.
attributes	string	A list of attributes to use to update the current attributes for the bulk volume job. Parseable into JSON object format.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireUpdateBulkVolumeStatusResult	Status of the bulk volume session.

Volume access group actions

Volume access group actions enable you to create, delete, view, and modify volume access groups and initiators in the system.

createVolumeAccessGroup

The createVolumeAccessGroup action enables you to create a new volume access group.

Additional information

When you create the volume access group, you need to give it a name, and you can optionally enter initiators and volumes. Once you create the group, you can add volumes and initiator IQNs. Any initiator IQN that you add to the volume access group is able to access any volume in the group without CHAP authentication. For volume access group system limits, see the *NetApp SolidFire Element OS API Reference Guide*.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
name	string	The name of the volume access group. Not required to be unique, but recommended. May be 1 to 64 characters in length.
initiators	Array of number	(Required) List of initiator IDs or names (IQNs and WWPNs) to include in the volume access group. If you pass a list of initiator names, the initiators are created if they do not already exist. If you pass a list of initiator IDs, the method returns an error if any of the initiators does not already exist. Passing initiator names is deprecated; you should use initiator IDs whenever possible.
volumes	Array of number	A list of volumeIDs to include in the volume access group.
attributes	string	A list of name-value pair attributes in string format, parseable into a JSON object.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireCreateVolumeAccessGroupStatusResult	Information about the newly created volume access group.

deleteVolumeAccessGroup

The deleteVolumeAccessGroup action enables you to delete a volume access group.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeAccessGroupID	number	The volumeAccessGroupID of the volume access group to be deleted.
deleteOrphanInitiators	boolean	• `true`: Delete initiator objects after they are removed from a volume access group. • `false`: Do not delete initiator objects after they are removed from a volume access group.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireDeleteVolumeAccessGroupStatusResult	Empty.

getVolumeAccessGroupEfficiency

The getVolumeAccessGroupEfficiency action enables you to retrieve efficiency information about a volume access group. Only the volume access group you provide as the parameter in this action is used to compute the capacity.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeAccessGroupID	number	Specifies the volume access group for which capacity is computed.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetEfficiencyResult	Efficiency statistics for the volume access group.

getVolumeAccessGroupLunAssignments

The getVolumeAccessGroupLunAssignments action enables you to retrieve details on LUN mappings of a specified volume access group.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeAccessGroupID	number	Specifies the volume access group for which to return information.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireGetVolumeAccessGroupLunAssignments Result	A list of all physical Fibre Channel ports, or a port for a single node.

listVolumeAccessGroups

The listVolumeAccessGroups action enables you to return information about the volume access groups that are currently in the system.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
startVolumeAccessGroupID	number	The volume access group ID at which to begin the listing. Mutually exclusive with the "volumeAccessGroups" parameter.
limit	number	The maximum number of volumeAccessGroup objects to return. Mutually exclusive with the "volumeAccessGroups" parameter.

Input	Type	Description
volumeAccessGroups	Array of integers	List of volumeAccessGroupID values to retrieve. Mutually exclusive with the "startVolumeAccessGroupID" and "limit" parameters.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireListVolumeAccessGroupsResult	Information about each volume access group.

modifyVolumeAccessGroup

You can use the modifyVolumeAccessGroup action to update initiators and add or remove volumes from a volume access group.

Additional information

If a specified initiator or volume is a duplicate of what currently exists, the volume access group is left as-is. If you don't specify a value for volumes or initiators, the current list of initiators and volumes is not changed.

You can also associate a virtual network with a volume access group. This association applies to all volumes in the volume access group.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeAccessGroupID	number	The ID of the volume access group to modify
name	string	The new name of the volume access group. Not required to be unique, but recommended. May be 1 to 64 characters in length.

Input	Type	Description
initiators	Array of integers	(Required) List of initiator IDs or names (IQNs and WWPNs) to include in the volume access group. If you pass a list of initiator names, the initiators are created if they do not already exist. If you pass a list of initiator IDs, the method returns an error if any of the initiators does not already exist. Passing initiator names is deprecated; you should use initiator IDs whenever possible.
deleteOrphanInitiators	boolean	• `true`: Delete initiator objects after they are removed from a volume access group. • `false`: Do not delete initiator objects after they are removed from a volume access group.
volumes	Array of integers	A list of volumeIDs of volumes to modify. Leave empty to not modify volumes.
attributes	string	A list of name-value pairs in string format, parseable into a JSON object.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireModifyVolumeAccessGroupResult	Information about the modified volume access group.

modifyVolumeAccessGroupLunAssignments

The modifyVolumeAccessGroupLunAssignments action enables you to define custom LUN assignments for specific volumes.

Additional information

This method changes only LUN values set on thelunAssignments parameter in the volume access group. All other LUN assignments remain unchanged.

LUN assignment values must be unique for volumes in a volume access group. You cannot define duplicate LUN values within a volume access group. However, you can use the same LUN values again in different volume access groups.

Note: Correct LUN values are 0 through 16383. The system generates an exception if you pass a LUN value outside of this range. None of the specified LUN assignments are modified if there is an exception.

Caution: If you change a LUN assignment for a volume with active I/O, the I/O can be disrupted. You may need to change the server configuration before changing volume LUN assignments.

Input parameters

This action has the following input parameters:

Input	Type	Description
connection	SolidFire:SolidFireConnection	The cluster connection to use.
volumeAccessGroupID	number	Unique volume access group ID for which the LUN assignments will be modified.
lunAssignments	Array of integers	The volumeIDs with new assigned LUN values.

Output parameter

This action has the following output parameter:

Type	Description
SolidFire:SolidFireModifyVolumeAccessGroupLunAssignmentsResult	Information about the modified volume access group LUN assignments.

Downloading Orchestrator log files

The NetApp SolidFire Plug-in for vRealize Orchestrator generates log files on the Orchestrator server. If you experience any issues with the plug-in, you can download these log files and send them to NetApp SolidFire support for assistance.

Before you begin

You must have administrator access to the Orchestrator server.

Steps

1. Browse to the IP address or DNS name of the Orchestrator server Control Center instance.
2. When prompted, log in with the administrator credentials.
3. On the **Control Center** home page, click **File System Browser** in the **Monitor and Control** section.
4. Click **app-server logs**.
5. Download the `solidfire_vro_api.log`, `solidfire_vro.log`, `server.log`, and `scripting.log` files.
6. Contact NetApp SolidFire support and be prepared to send the files to your support representative and give details about the issue you are experiencing.

Related concepts

Contacting NetApp SolidFire Active Support on page 140

Where to find additional information

You can use the resources in this section for additional information about the VMware vRealize Orchestrator Plug-in and NetApp SolidFire storage systems.

- For more information about using the vRealize Orchestrator Plug-in, see the *NetApp SolidFire VMware vRealize Orchestrator Plug-in User Guide*.
- For more information about deploying, configuring, and using your cluster, see the *NetApp SolidFire Element OS User Guide*.
- For information about the NetApp SolidFire API, see the *NetApp SolidFire Element OS API Reference Guide*.

Contacting NetApp SolidFire Active Support

You can contact NetApp SolidFire Active Support if you have any questions or comments about SolidFire documents or products in general.

Visit *NetApp SolidFire support* or email *ng-SF-Support@netapp.com* for help with NetApp SolidFire systems.

Copyright information

Trademark information

How to send comments about documentation and receive update notifications

You can help us to improve the quality of our documentation by sending us your feedback. You can receive automatic notification when production-level (GA/FCS) documentation is initially released or important changes are made to existing production-level documents.

If you have suggestions for improving this document, send us your comments by email.

doccomments@netapp.com

To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

If you want to be notified automatically when production-level documentation is released or important changes are made to existing production-level documents, follow Twitter account @NetAppDoc.

You can also contact us in the following ways:

- NetApp, Inc., 495 East Java Drive, Sunnyvale, CA 94089 U.S.
- Telephone: +1 (408) 822-6000
- Fax: +1 (408) 822-4501
- Support telephone: +1 (888) 463-8277

Index