Hitachi Virtual Storage Platform Fx00 and Gx00

SVOS RF 8.2

Hitachi Storage Advisor Embedded Guide

This manual explains how to use Storage Advisor Embedded and its web-based user interface to manage and operate storage systems. The contents of this manual apply to Hitachi Virtual Storage Platform F350, F370, F700, F900 all-flash arrays and Hitachi Virtual Storage Platform G350, G370, G700, G900 storage systems.

The options to manage servers and attach volumes to a server are only applicable in certain configurations.
Contact your Hitachi representative for more details and to enable these options for your configuration.

MK-97HM85022-02
July 2018

All other trademarks, service marks, and company names in this document or website are properties of their respective owners.

EXPORT CONTROLS - Licensee will comply fully with all applicable export laws and regulations of the United States and other countries, and Licensee shall not export, or allow the export or re-export of, the Software, API or Materials in violation of any such laws or regulations. By downloading or using the Software, API, or Materials, Licensee agrees to the foregoing and represents and warrants that Licensee is not located in, under the control of, or a national or resident of any embargoed or restricted country.

Contents

Chapter 1: Managing storage system operations

Get an overview of Storage Advisor Embedded and its features for storage system management.

Storage Advisor Embedded overview

Storage Advisor Embedded provides simplified management of a storage system.

Storage Advisor Embedded allows you to quickly build an environment with a storage system and its resources, such as pools, where you can perform simple operations using a web-based user interface without having to consider complicated storage system configurations.

The web-based user interface of Storage Advisor Embedded consists of the elements shown in the following figure.

Navigation Bar

You can manage and operate storage systems in Storage Advisor Embedded by selecting and operating the resources displayed in the navigation bar.

Health Status

This area displays statuses such as Error or Warning when any of the following events are detected:

- The pool usage rate exceeds a threshold value
- An error occurs in a backup or a restore operation
- SIMs (messages that are generated when the controller of the storage system detects an error or a service request) are sent

Dashboard

The dashboard appears when you log in. In the dashboard, you can see the following items, and check the operating status and the entire capacity of the storage system:

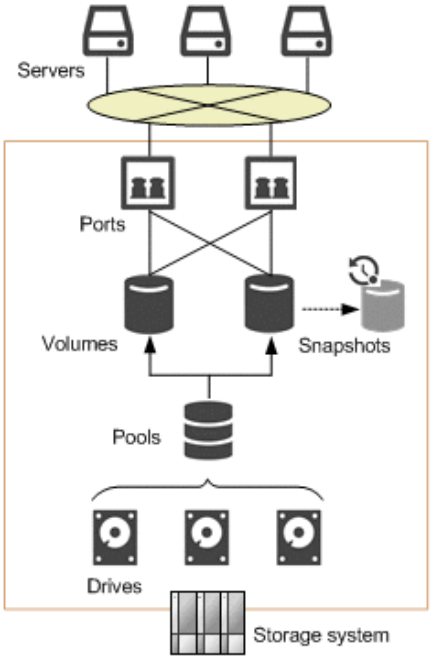
- Information about the storage system, including the storage system name, model name, and serial number. The resource status is also displayed in this area, and you can check the Health Status message by clicking the status and launching the maintenance utility to view alerts. Click Information > System Information to view detailed information about the storage system.
- Capacity Usage, including the capacity of the storage systems, and the amount of used capacity.
- The number of registered servers. Access the Servers page to perform operations related to servers.
- The number of volumes. Access the Volumes page to perform operations related to volumes.
- Data Reduction, including the amount of data reduction achieved by using the capacity saving feature (deduplication and compression) for the storage system. The value displayed as the capacity of data after reduction includes (in addition to user data) the size of metadata and garbage data generated by the storage system. For this reason, the value might be temporarily greater than the capacity of data before reduction.
- Storage System Performance, including the status of IOPS, the response time (time required to respond to a request from the server), and the data transfer speed. By checking the performance values regularly, you can notice at an early stage any large changes from the usual performance values. In particular, a large increase in the response time signals a problem in the storage system.

Note:

Performance reports may not show the current information if performance data is not obtained from the storage system. This issue should resolve itself after some time when the storage system is less busy and able to obtain performance data.

Storage system resource configuration

The following figure shows the storage system resources that can be configured and managed using Storage Advisor Embedded.



Drives

Drives are hardware composed of storage media, and devices for reading and writing to the storage media. According to the specified settings, you can use drives as data drives for reading and writing data, or as spare drives for copying data to continue storage system operations if an error occurs in the data drive. Drive capacity is displayed as physical capacity in one of the following units: KB (kilobyte), MB (megabyte), GB (gigabyte), or TB (terabyte).

Pools

Pools for thin provisioning are virtual areas in a storage system that are associated with multiple drives. Thin provisioning provides virtual volumes to a server and uses the actual capacity from a pool when a server makes a write request.

Volumes

Volumes are virtual storage areas attached to the server.

Snapshots

Snapshots are images of volumes at a specific point. When a snapshot is created, differential data from the original volume is stored in the pool. You can use snapshots to create backups, or to create volumes to clone a snapshot so that the data can also be used on other servers.

Ports

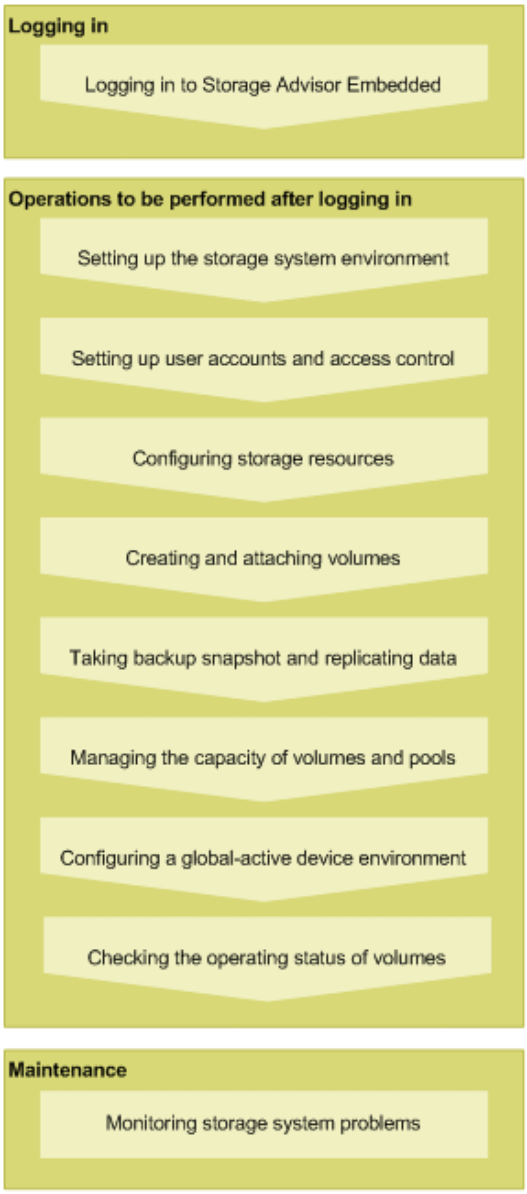
Ports are interfaces for connecting storage systems with servers.

Servers

Servers are units for managing business servers that use a storage system. For cluster configurations, all of the servers (nodes) that make up a cluster are managed as one server.

Storage system management workflow

Storage Advisor Embedded allows you to easily configure your storage system resources, such as volumes and pools, and perform daily operations, such as backing up data and monitoring. You can perform operations by using a sophisticated web-based user interface or by using REST API functions, incorporated into business applications or executed automatically as scripts. The following figure shows the workflow for configuring and managing storage systems using Storage Advisor Embedded.



Management client OS and browser requirements

The following describes the requirements for the management client to log in to Storage Advisor Embedded.

OS and architecture	Browser and version
Windows Server 2016 (64-bit)	Google Chrome (version 63 or later) Internet Explorer 11
Windows 10 (32-bit or 64-bit)	
Windows 8.1 (32-bit or 64-bit)	

OS and architecture	Browser and version
Red Hat Enterprise Linux 7.4 (64-bit)	Google Chrome (version 63 or later)

Logging in to Storage Advisor Embedded

You must log in to Storage Advisor Embedded.

Before you begin

Identify the following items:

- Protocol

  For non-SSL communication, use `http`. For SSL communication, use `https`.
- IP address of the storage system

  Use the IP address of CTL1 or CTL2, as specified in the maintenance utility.

Procedure

1. Open a web browser and specify the following URL:

   ```
   protocol://IP-address-of-storage-system/
   ```

   Note:

   If the web browser repeatedly fails to open, close all active browser windows, and then clear the web browser's cache.

   If the web browser fails to open even after the cache has been cleared, start the maintenance utility by specifying the following URL, and then check for alerts:

   ```
   protocol://IP-address-of-storage-system/MaintenanceUtility/
   ```

   To check for alerts, log in as a user registered in the Maintenance user group (a built-in user group).
2. Enter a user name and password, and then log in.

   Use the following account when logging in for the first time:

   User name: maintenance

   Password: raid-maintenance

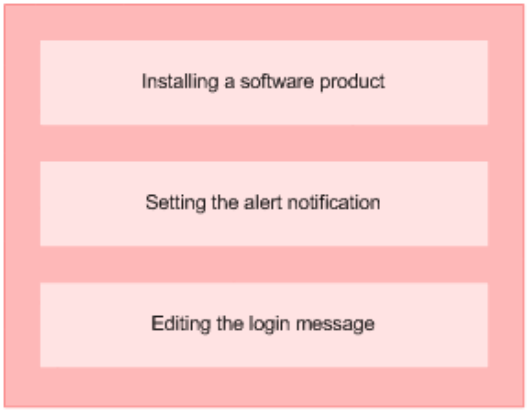
   The account will be locked for 60 seconds after three consecutive unsuccessful login attempts.

Chapter 2: Setting up the storage system environment

You can install various software products that enable you to use storage system functionality. You can also configure a storage system to link to a mail server so that alert notifications are sent in the event of a failure, and edit the message displayed in the login window.

Setting up the storage system environment

To use the various storage system functions, register your license key for the software product. You can also set the method by which alert notifications are sent in the event of a failure, and edit the message displayed in the login window. The following figure shows this workflow.



Installing a software product

Register the license key for optional software so you can use the additional features.

Before you begin

Have available the license key code or the license key file for the software product to be installed.

Procedure

1. In the navigation bar, click (**Settings**), and then select **Licenses**.
2. In the maintenance utility, click **Install**.

3. Specify the license key code or the license key file, and register the license key.
4. In the list of license keys, confirm that the status of the software product has changed to **Installed**.
5. In the maintenance utility, click **Log Out**.

Enabling or disabling a license key

You can enable or disable the license key (Term key) of a software product that has not yet expired.

Procedure

1. In the navigation bar, click (**Settings**), and then select **Licenses**.
2. In the maintenance utility, select the software product whose status you want to change. Then, click **Enable** to enable the license or **Disable** to disable the license.
3. In the license key list, verify that the status of the software product has changed to **Installed (Enabled)** or **Installed (Disabled)**.
4. In the maintenance utility, click **Log Out**.

Uninstalling a software product

You can uninstall software products that are no longer used.

Procedure

1. In the navigation bar, click (**Settings**), and then select **Licenses**.
2. In the maintenance utility, select the software product to be uninstalled, and then click **Remove**.
3. In the list of license keys, confirm that the status of the software product has changed to **Not Installed**.
4. In the maintenance utility, click **Log Out**.

Setting the alert notification

You can send SIMs (Service Information Messages) to the administrator by using emails, syslog outputs, and SNMP traps.

SIMs are messages that are generated when the controller of the storage system detects an error or requests a service. Messages may be output from channels, ports, or microprocessors of the storage system. They indicate maintenance required for the storage system and identify actions to take when a failure occurs.

Before you begin

- Identify the SIM associated with the alert notification.
- For email notifications:
  - Make sure port 25 of the SMTP mail server is free.
  - Identify the mail server information and the email addresses of the sender and receivers.
- For syslog output:
  - Make sure the port used for forwarding messages to the syslog server is free.
  - Identify the syslog server information and the settings for communicating with the syslog server.
- For SNMP notifications:
  - Make sure the SNMP manager to which notifications are to be sent has been configured.
  - Identify the information for sending and receiving SNMP traps.

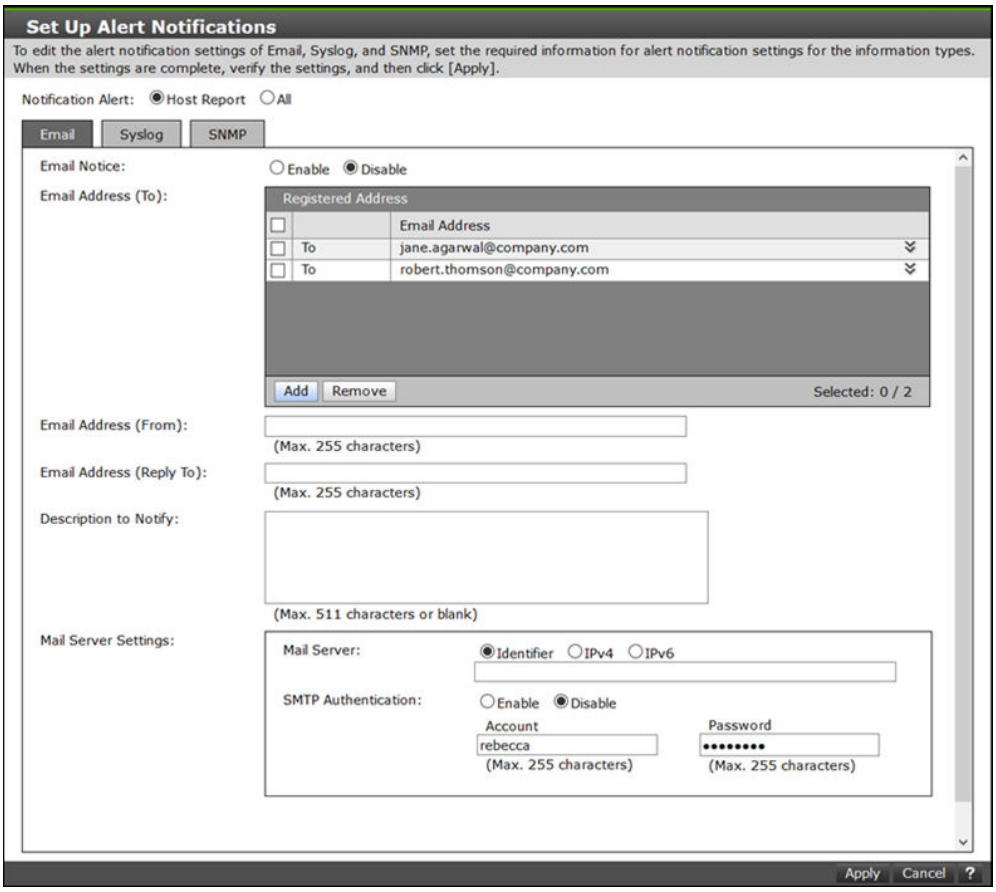
Note:

For details about the SIM to which the alert notification applies, see the SIM Reference Guide for your storage system.

Procedure

1. In the navigation bar, click (**Settings**), and then select **Alert Notifications**.

2. In the maintenance utility, click **Set Up**.



3. Click the tab for the desired notification method (Email, Syslog, or SNMP), specify the required information and selections, and then click **Apply**.

   You can specify multiple notification methods.

   For details about the SNMP trap configuration and supported MIB specifications, see the *SNMP Agent User Guide*.

4. To verify that the notification settings are correct, send a test notification by using the notification method that was set.
5. In the maintenance utility, click **Log Out**.

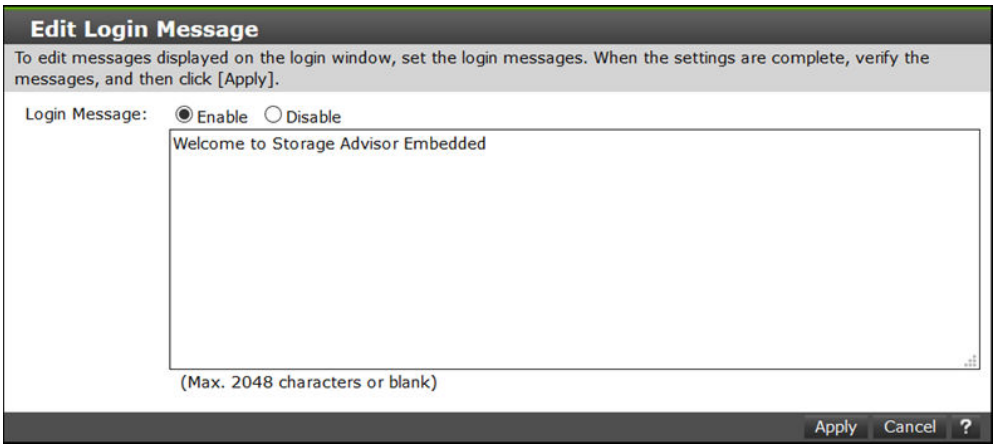
Modifying the login message

You can modify the message that displays in the login window of Storage Advisor Embedded. This is the same message that displays in the login window of the maintenance utility.

Procedure

1. In the navigation bar, click ⚙ (**Settings**), and then select **Maintenance Utility**.

2. From **Menu**, click **System Management** > **Edit Login Message**.



3. Modify the login message.
4. In the maintenance utility, click **Log Out**.

Chapter 3: Setting up user accounts and permissions

Use the maintenance utility to create and manage user accounts for Storage Advisor Embedded.

User accounts and permissions

You can set up a user account for using Storage Advisor Embedded and managing a storage system.

A user's operating permissions are set based on the role assigned to the user group to which the user belongs.

To use Storage Advisor Embedded to manage a storage system, you need to register the user in the Administrator user group (a built-in user group) and the Maintenance user group (a built-in user group).

Even if a user account was created by using another management tool, such as the storage system REST API, the account can be used in Storage Advisor Embedded if registered in a user group to which all of the following roles have been assigned:

- Storage Administrator (Initial Configuration)
- Storage Administrator (Provisioning)
- Storage Administrator (Local Copy)
- User Maintenance

Creating user accounts

You can create a user and assign that user to one or more of the available user groups, depending on what the user needs to accomplish in Storage Advisor Embedded.

Before you begin

Obtain the following information:

- User name
- Password

Procedure

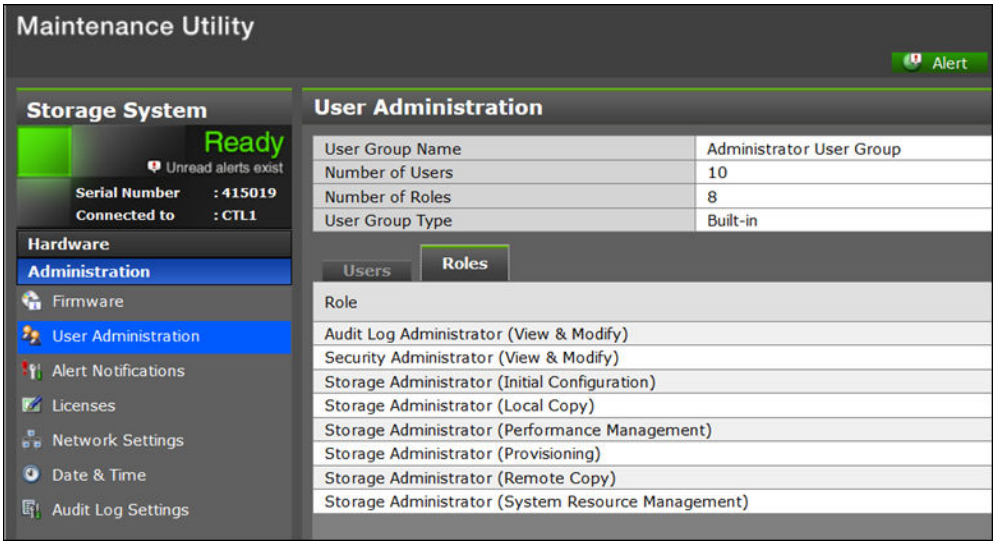
1. In the navigation bar, click (**Settings**), and then select **User Administration**.

2. In the maintenance utility, click **Create User**.
3. Specify the required information to create a user account.

   Select **Administrator User Group** and **Maintenance User Group** as the user groups.

   Note:

   To check the roles set for a user group, click the name of the user group and select the **Roles** tab.



4. In the maintenance utility, click **Log Out**.
5. Log in to Storage Advisor Embedded by using the newly created user account.

Modifying user accounts

You can modify the passwords of user accounts, or the user groups that user accounts belong to.

Procedure

1. In the navigation bar, click (**Settings**), and then select **User Administration**.
2. In the maintenance utility, click the user group name.
3. Select the user account that you want to modify, and then click **Edit**.
4. Specify the required information.

   Note:

   To check the roles set for a user group, click the name of the user group and select the **Roles** tab.

5. In the maintenance utility, click **Log Out**.

6. Confirm that you can log in to Storage Advisor Embedded by using the revised user account.

Removing user accounts

You can remove user accounts.

Procedure

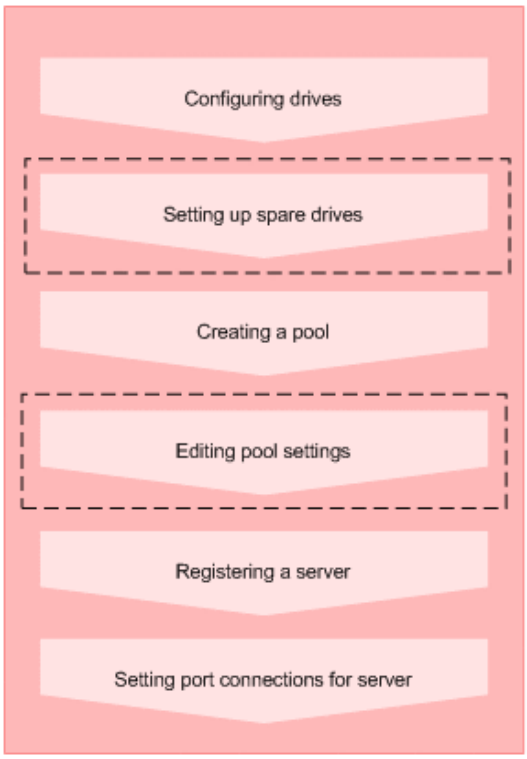
1. In the navigation bar, click (**Settings**), and then select **User Administration**.
2. In the maintenance utility, click the user group name.
3. Select the user that you want to remove, and then click **Remove**.
4. In the maintenance utility, click **Log Out**.

Chapter 4: Configuring storage resources

Configure drives and pools, and register a server so you can attach volumes.

Workflow for configuring storage resources

Prepare the storage capacity of a storage system to be used as volumes. To attach the volumes, register servers to be managed by Storage Advisor Embedded and configure connection information for the port on the storage system, as shown in the following figure.



Legend:

: Operations performed as necessary

Note:

After registering servers, you can add and delete WWNs or iSCSI initiator names.

When using the REST API:

The following REST APIs can be executed to automate the operations:

- Setting up a spare drive
- Creating a pool
- Changing pool settings
- Registering a server
- Adding path information to a server
- Creating a volume
- Changing the settings of an iSCSI target
- Getting information about the WWN (iSCSI name) of an HBA
- Deleting information about the WWN (iSCSI name) of an HBA from a server

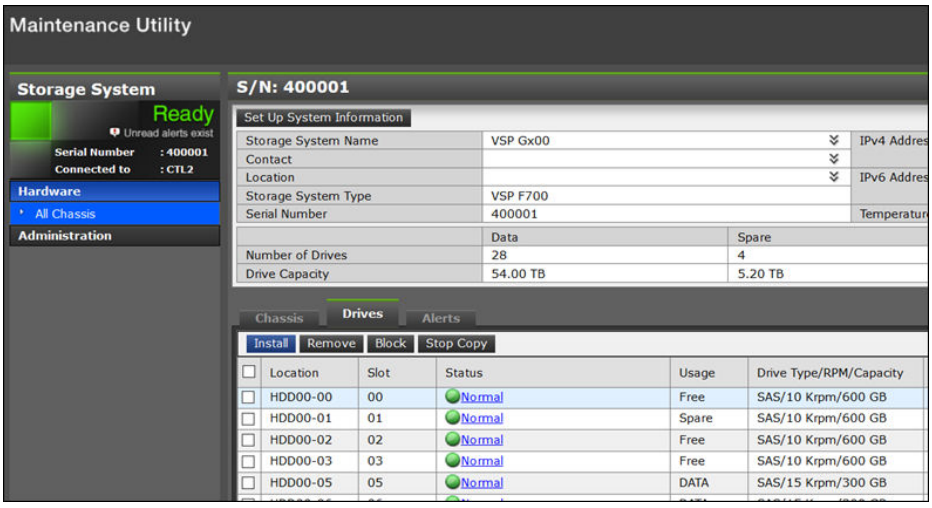
Configuring drives

You can configure the storage system to recognize drives that are inserted into slots and configured for use.

Drives are hardware composed of storage media and devices for reading and writing to the storage media. According to the specified settings, you can use drives as data drives for reading and writing data, or as spare drives if an error occurs in a data drive.

Procedure

1. In the navigation bar, click (**Settings**), and then select **Drive Settings**.



2. In the maintenance utility, click **Install** to detect the added drives.
3. Click **Install** again for the storage system to recognize the drives.
4. Confirm that the new drives are added to the list of available drives.
5. In the maintenance utility, click **Log Out**.

Setting up spare drives

You can set up spare drives to prepare for the possibility of a drive failure. If a drive failure occurs, a spare drive allows you to continue storage management tasks by using data that was copied to another drive.

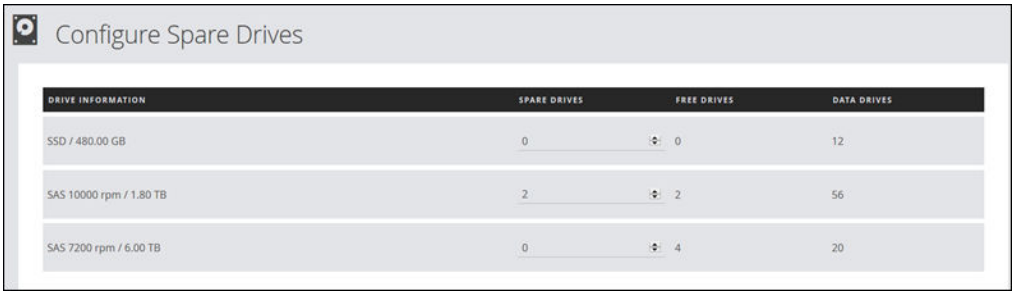
Free drives that are not specified as data drives (drives that are not used for the reading and writing of data) can be selected and set up as spare drives.

Before you begin

Drives that are identified as spare are displayed in the list of available drives.

Procedure

1. In the navigation bar, click (**Settings**), and then select **Configure Spare Drives**.



2. Specify the number of spare drives that you want to set up as **Spare Drives**, and then click **Submit**.

Creating a pool

You can create a pool by specifying drives.

Pools for thin provisioning are virtual areas in a storage system that are associated with multiple drives. Thin provisioning provides virtual volumes to a server and uses the actual capacity from a pool when a server makes a write request.

If you create a pool in Storage Advisor Embedded, a recommended configuration is displayed from among the various different configurations supported by the storage system. This configuration consists of the combination of drives that allows drive space to be used most effectively.

Using the Dynamic Tiering license, you can create tiered pools by specifying drives that have drive types, speeds, or capacities. The tiers are arranged by performance, and data is automatically stored in optimum tiers.

Caution:

In a pool used to store differential-data snapshots, the usage may exceed the depletion threshold set on the pool. If that happens, all of the snapshots may become unusable. You can change the depletion threshold by editing the pool.

Before you begin

- Make sure that a drive that is configured in the storage system is available.
- Identify the pool name.

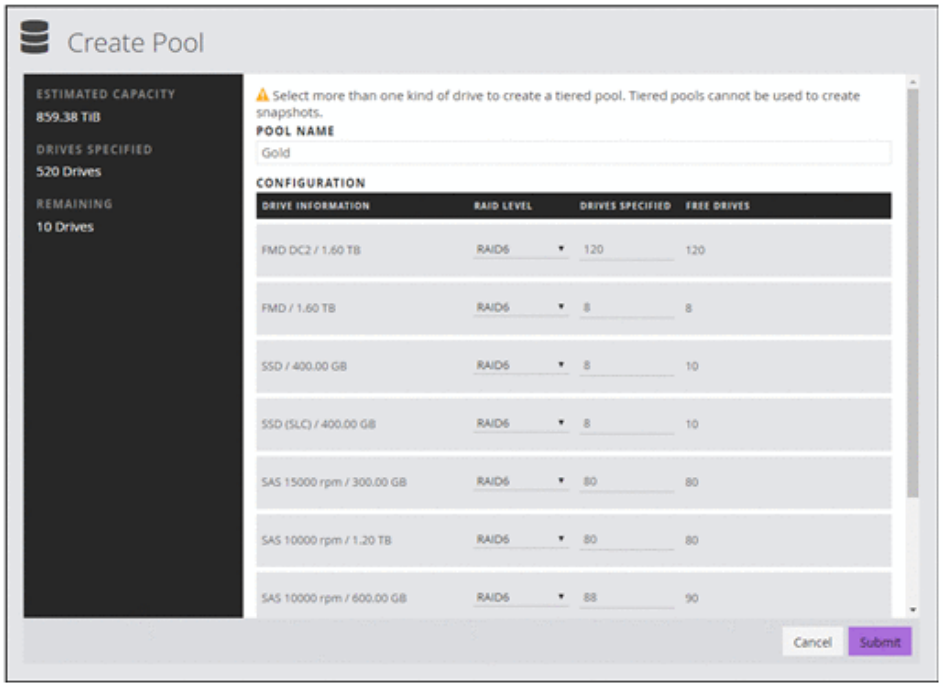
  To specify the configuration of the pool being created, also obtain the following information:

  - Drive information (drive type, drive speed, and capacity)
  - RAID level
  - Number of drives to be used

Procedure

1. In the navigation bar, click **Pools**.
2. Click the plus sign (**+**).
3. Enter the **Pool Name**. To accept the recommended configuration, click **Submit** to create a pool.

   To specify the pool configuration, change the number of drives. To specify a different RAID configuration, select the RAID level and specify the number of usable drives. Next, click **Check** to check the available combinations, and then click **Submit**.



Modifying pool settings

You can modify the pool name and the threshold value set for the pool usage rate.

In a pool used to store differential-data snapshots, the usage may exceed the depletion threshold set on the pool. If that happens, all snapshots may become unusable. You can change the depletion threshold by modifying the pool.

Procedure

1. In the navigation bar, click **Pools**.
2. Click (**Edit Pool**) for the pool you are editing.
3. Modify the settings.
4. Click the pool name to open the details page, and then confirm that the settings are correct.

Registering a server

Register servers in Storage Advisor Embedded so that you can assign storage resources to them. For cluster configurations, you need to register all of the servers (nodes) that make up the cluster as one server. In that case, for the WWNs or iSCSI initiator names that are used for connecting to storage system ports, add all of the WWNs or iSCSI initiator names of each server.

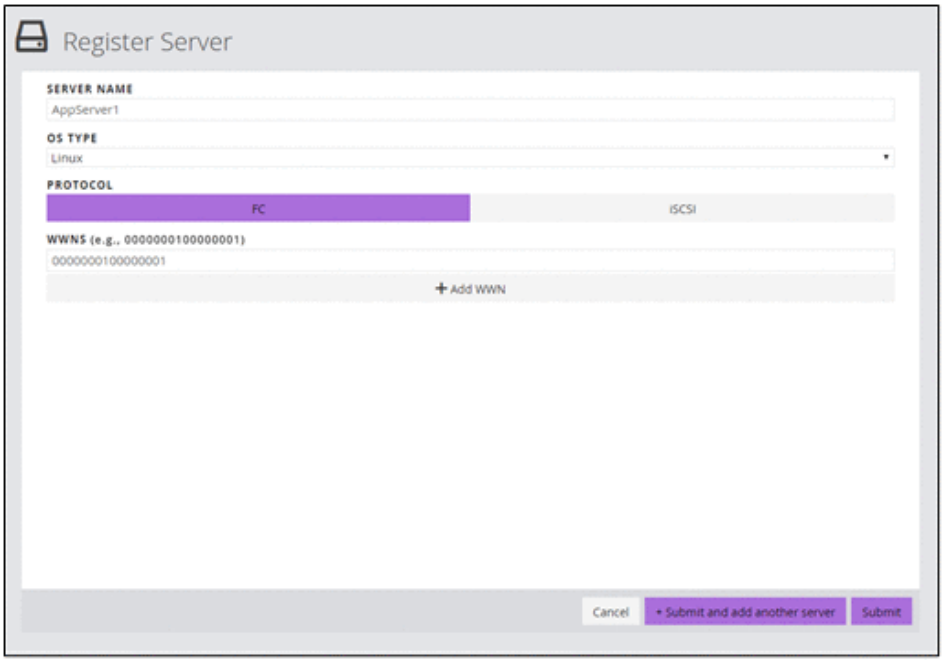
Before you begin

- The server must be connected to the storage system via a Fibre Channel (FC) connection or an iSCSI connection.
- If you are using an FC connection, obtain the following items:
  - Server name
  - OS type
  - WWN
- If you are using an iSCSI connection, obtain the following items:
  - Server name
  - OS type
  - iSCSI initiator name

Procedure

1. In the dashboard or in the navigation bar, click **Servers**.
2. Click the plus sign (**+**).
3. Enter **Server Name**. Enter a WWN or a iSCSI initiator name to be used when connecting with a storage system port. To specify multiple WWNs or iSCSI initiator names, add rows by using **+ Add WWN** or **+ Add iSCSI Initiator Name**. In addition, specify the required information to register the server.

   To register an additional server, click **+ Submit and add another server**.



Setting port connections for a server

You can set information for connecting the server port and storage system port.

Before you begin

- The server must be registered in Storage Advisor Embedded.
- Identify the following items:
  - Server name
  - WWN or iSCSI initiator name
  - Storage system port ID

Procedure

1. In the dashboard or in the navigation bar, click **Servers**.
2. Select one or more servers and click (**Configure Port Connections**).
3. Click the WWN or iSCSI initiator name of the server and the storage system port ID to set a port connection.

Note:

After you configure paths to a server to which volumes are already allocated, the following warning message might be temporarily displayed in the server details window. If this message does not disappear after a while, refresh the window.

```
Port connections for volume allocation are partially
configured. To expose volumes to the server, configure port
connections.
```

Managing servers

You can edit server settings and delete servers.

Modifying server settings

You can modify the server settings, such as server name, OS type, and other information, registered in Storage Advisor Embedded.

Before you begin

- If you are using a Fibre Channel connection, obtain the following items:
  - Server name
  - OS type
  - WWN
- If you are using an iSCSI connection, obtain the following items:
  - Server name
  - OS type
  - iSCSI initiator name

Procedure

1. In the dashboard or in the navigation bar, click **Servers**.
2. Click (**Edit Server**) of the port that you want to edit.
3. Modify the settings.
4. Click the server name to open the details page, and then confirm that the settings are correct.

Adding a WWN or iSCSI initiator name to a server

You can add a WWN or iSCSI initiator name when you add HBAs or configure a cluster by adding servers. For cluster configurations, you need to register all of the servers (nodes)

that make up the cluster as one server. In that case, for the WWNs or iSCSI initiator names that are used for connecting to storage system ports, add all of the WWNs or iSCSI initiator names of each server.

Procedure

1. In the dashboard or in the navigation bar, click **Servers**.
2. Click (**Edit Server**) for the server for which you are adding a WWNor iSCSI initiator name .
3. Add a WWN or an iSCSI initiator name.
4. Click the server name to open the details page, and then verify that the WWN or iSCSI initiator name was added.

Deleting a WWN or iSCSI initiator name from a server

You can delete a WWN or iSCSI initiator name.

Procedure

1. In the dashboard or in the navigation bar, click **Servers**.
2. Click (**Edit Server**) for the server for which you are deleting the WWNor iSCSI initiator name .
3. Delete the WWN or iSCSI initiator name.
4. Click the server name to open the details page, and then verify that the WWN or iSCSI initiator name was deleted.

Modifying iSCSI target names

If the server and the storage system are connected using iSCSI, you can modify the iSCSI target name (iSCSI name of the iSCSI target).

Procedure

1. In the dashboard or in the navigation bar, click **Servers**.
2. Click the name of the server whose settings you want to modify.
3. On the details page, click (**Edit iSCSI Target Names**).
4. Check and modify the settings.
5. Click the server name to open the details page, and then verify that the settings are correctly specified.

Deleting a server

When you no longer need to manage a server, you can delete it from Storage Advisor Embedded.

Before you begin

- Confirm the name of the server that you want to delete.
- Ensure that no volumes are attached to the server that you want to delete.

Procedure

1. On the dashboard or in the navigation bar, click **Servers**.
2. Select one or more servers and click (**Delete Servers**).

Modifying the storage system port settings

You can modify the storage port settings.

Procedure

1. In the navigation bar, click **Ports**.
2. Click (**Edit Port**) of the port that is to be edited.
3. Modify the following settings:
   - Port ID
   - Port security
   - Port speed
   - FC information or iSCSI information

   Note:

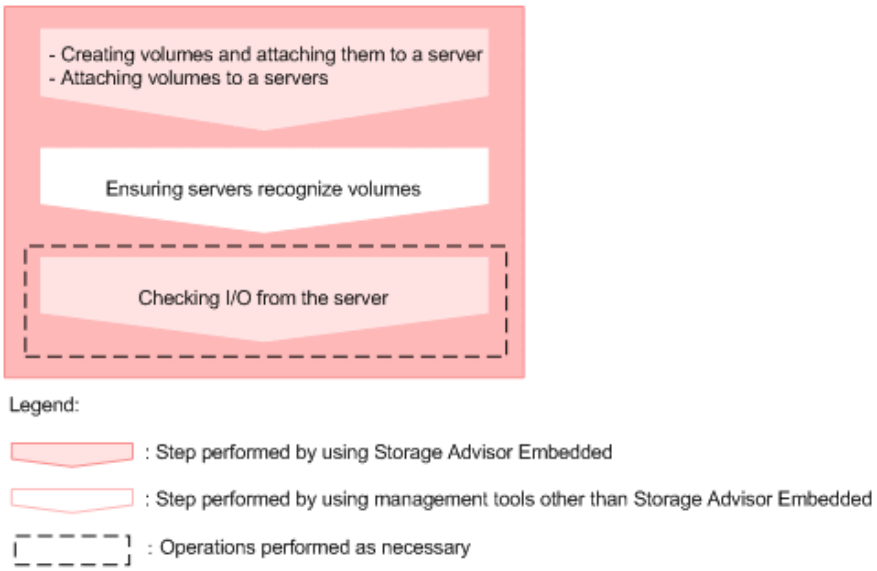
   For iSCSI (optical) ports, **Port Speed** is fixed at 10 Gbps. If a value other than 10 Gbps is specified, that value will be ignored.
4. Click the port ID to open the details page, and then verify that the settings are correctly specified.

Chapter 5: Creating and attaching volumes

You can attach a volume to a server managed by Storage Advisor Embedded.

Workflow for creating and attaching volumes

Attach volumes to the server managed by Storage Advisor Embedded. You can either create volumes and attach them to a server in a single workflow, or create volumes separately and then attach them to a server.



When using the REST API:

The following REST APIs can be executed to automate the operations:

- Creating a volume
- Connecting a volume to a server

Creating volumes and attaching them to a server

Storage Advisor Embedded allows you to create volumes and attach them to servers in a single workflow or one at a time. By enabling the capacity saving feature (deduplication and compression) when creating volumes, you can reduce the amount of data to be stored, depending on the data type.

Before you begin

- Ensure that a pool is available.

  If you want to enable the capacity saving feature for the volumes to be created, you cannot use a tiered pool.
- Obtain the following items:
  - Server name
  - Pool name
  - Capacity
  - Number of volumes
  - Volume name

Note:

If a host group (or iSCSI target) that was assigned to a virtual storage machine by using another management tool has been added to a server, you cannot attach volumes to the server.

Procedure

1. In the dashboard or in the navigation bar, click **Servers**.
2. Select one or more servers and click , and then select **Create and Attach Volumes**.
3. Create volumes by specifying the required information, and then click **Submit** to attach those volumes to the server.

For the name of a volume, specify a name that is a combination of **Volume Name** and a number beginning with **Suffix Start Number**.

For example, if you specify "volume" for **Volume Name**, "7" for **Suffix Start Number**, and "2" for **Number of Digits**, the volume name will be "volume07".

Creating volumes

You can create volumes from the free capacity of a pool. By enabling the capacity saving feature (deduplication and compression), you can reduce the amount of data to be stored, depending on the characteristics of the data.

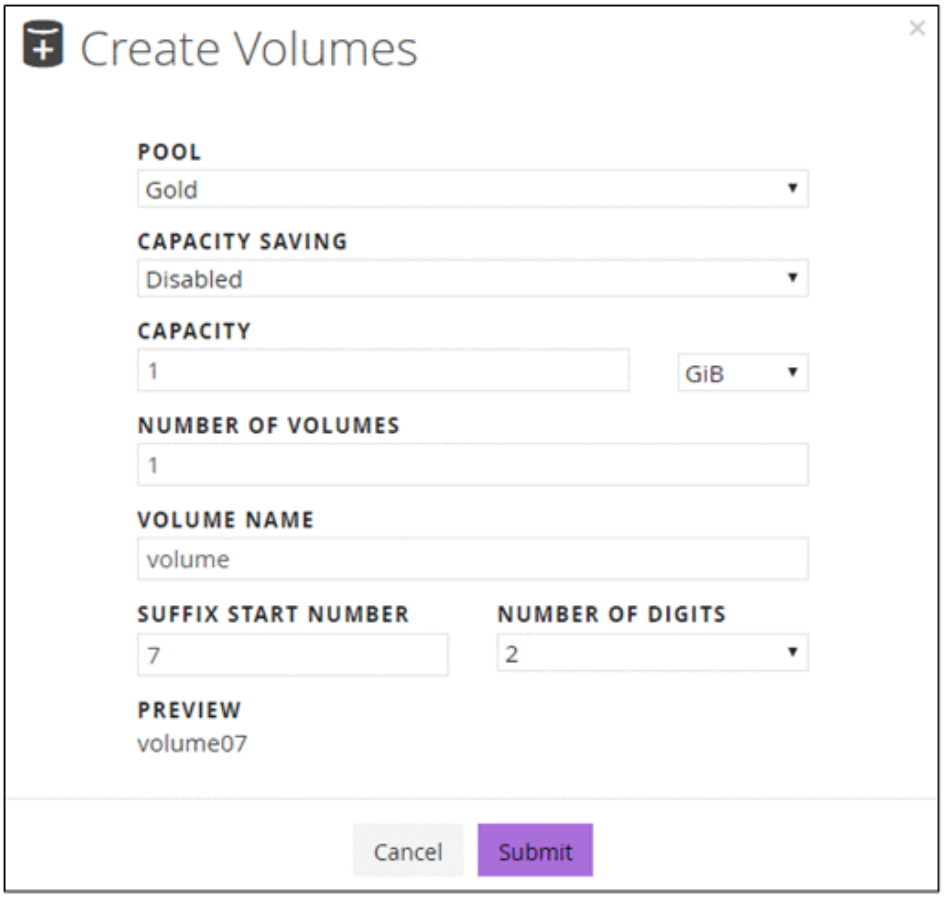
Before you begin

- Ensure that a pool is available.

  If you want to enable the capacity saving feature for the volumes to be created, you cannot use a pool that has a hierarchical structure.
- Obtain the following items:
  - Pool name
  - Capacity
  - Number of volumes
  - Volume name

Procedure

1. In the dashboard or in the navigation bar, click **Volumes**.
2. Click the plus sign (**+**).
3. Specify the required information and create the volumes.



For the name of a volume, specify a name that is a combination of **Volume Name** and a number beginning with **Suffix Start Number**.

For example, if you specify "volume" for **Volume Name**, "7" for **Suffix Start Number**, and "2" for **Number of Digits**, the volume name will be "volume07".

Attaching volumes to servers by selecting servers

You can attach an existing volume to a server by selecting a volume from the server page.

Procedure

1. On the dashboard or in the navigation bar, click **Servers**.
2. Select one or more servers and click , and then select **Attach Volumes**.

   If a host group (or iSCSI target) that was assigned to a virtual storage machine by using another management tool has been added to a server, you cannot attach volumes to the server.
3. Select the volume to be attached to the server.

Volumes for which **Attached Unmanaged** is displayed for the volume type are volumes that were attached by using management software other than Storage Advisor Embedded.

Attaching volumes to servers by selecting volumes

You can attach an existing volume to a server by selecting a server from the volume page.

Procedure

1. In the dashboard or in the navigation bar, click **Volumes**.
2. Select one or more volumes and click (**Attach Volumes**).

   Volumes for which **Attached Unmanaged** is displayed for the volume type are volumes that were attached by using management software other than Storage Advisor Embedded.
3. Select one or more servers to which the volumes will be attached.

   If a host group (or iSCSI target) that was assigned to a virtual storage machine by using another management tool has been added to a server, you cannot attach volumes to the server.

Checking I/O from the server

You can check the status of I/O from the server to determine whether a volume is correctly attached.

Procedure

1. In the dashboard or in the navigation bar, click **Servers**.
2. Click the server name to open the server details page.
3. Click the volume name to open the volume details page.
4. Check the IOPS graph on the **Performance Monitor** to confirm that I/O information is being displayed.

Modifying a volume name

You can modify the volume name.

Procedure

1. In the dashboard or in the navigation bar, click **Volumes**.
2. Click (**Edit Volumes**) of the volume you want to modify.
3. Modify the name and click **Submit**.

Modifying the setting of the capacity saving feature for volumes

You can modify the setting of the capacity saving feature (deduplication and compression) for volumes.

The capacity saving feature is significantly effective in reducing the amount of data stored on other volumes used for similar operations. Enable the capacity saving feature to reduce the amount of data stored on other volumes. Disable the feature for volumes where the function is not effectively reducing the amount of data being stored. If you change the capacity saving feature settings while it is enabled, rehydrating data might take some time.

Before you begin

- Identify the server name.
- Identify the volume name.
- If the planned changes can impact the free space on the pool, check the current amount of free space.

Procedure

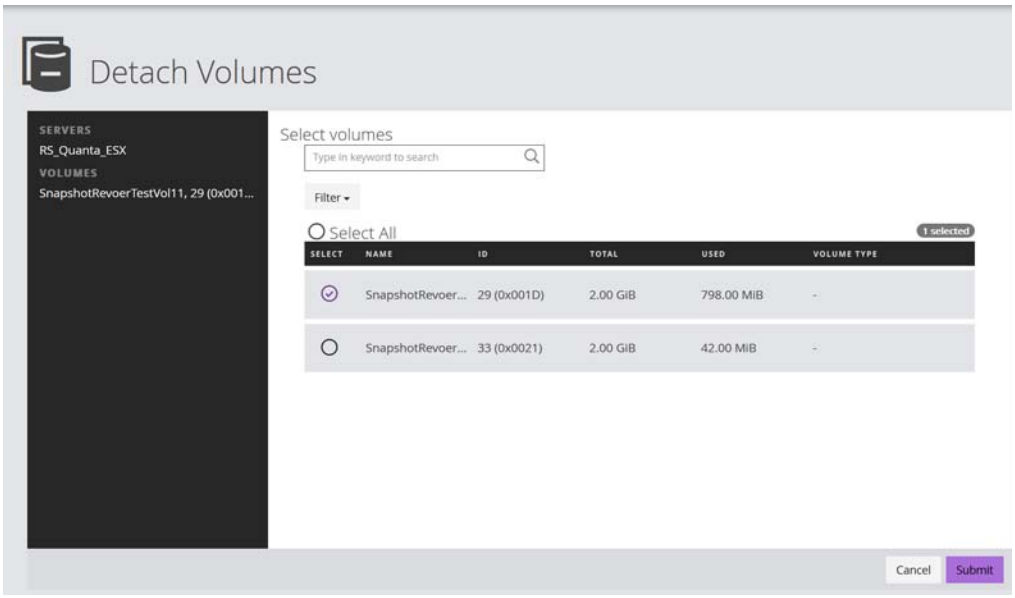
1. In the dashboard or in the navigation bar, click **Servers**.
2. Click the server name to open the server details page.
3. Select one or more volumes and click (**Edit Volumes**).
4. Modify the setting of the capacity saving feature.
5. Click the volume name to open the details page, and then confirm that the settings are correctly specified.

Detaching volumes from a server

If a server no longer needs to use a specific volume, you can detach that volume from the server. You can select multiple volumes from multiple servers, but only the volumes that are attached to all selected servers.

Procedure

1. In the dashboard or in the navigation bar, click **Servers**.
2. Select one or more servers and click (**Detach Volumes**).



Note that only those volumes that are attached to all of the selected servers can be detached.

3. Select one or more volumes and detach them.

Deleting volumes

You can delete volumes that are no longer needed. Data in the volumes is also deleted. You cannot delete volumes that are attached to servers.

Before you begin

- Ensure that there is no snapshot created for the volume.
- Neither Snapshot nor Attached Unmanaged can be displayed for the volume type.

Note:

Volumes for which Attached Unmanaged is displayed for the volume type are volumes that were attached by using management software other than Storage Advisor Embedded.

Procedure

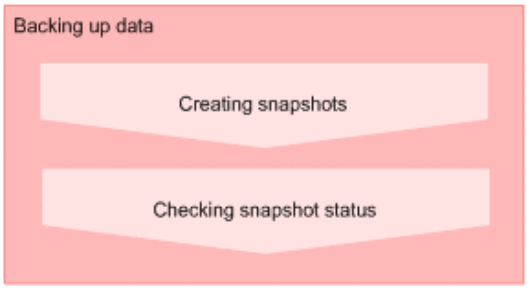
1. In the dashboard or in the navigation bar, click **Volumes**.
2. Select one or more volumes and click (**Delete Volumes**) .

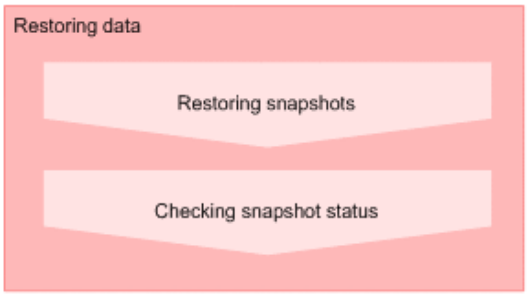
Chapter 6: Taking backup snapshots and replicating data

You can create backup data and use data for other purposes, such as moving the data to another server.

Workflow for creating backup data using snapshots

To create backup data, you can add a snapshot of a volume. Snapshots are images of volumes at a specific point in time. Differential data from the original volume is stored in the pool when a snapshot is created. This section describes the workflows for creating a snapshot and restoring data. For details about snapshot-related functions, see the Hitachi Thin Image User Guide.





Schedule creating snapshots by using REST API functions. You can also manually create a snapshot by using the web-based user interface functions.

When using the REST API:

The following REST APIs can be executed to automate the operations:

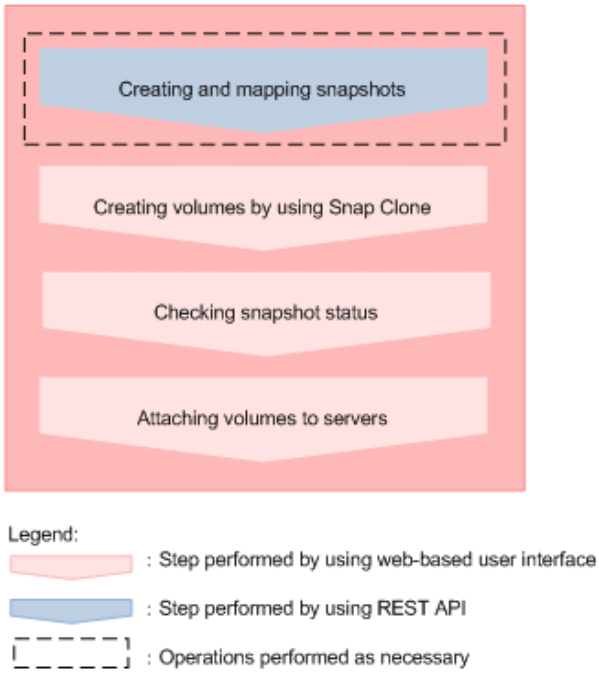
- Creating a snapshot
- Getting snapshot information
- Getting information about a specific snapshot
- Restoring a snapshot

Note:

If you use the web-based user interface, the date and time when a snapshot was created are displayed based on the time zone set for the system of the management client. If you use the REST API, the data and time are returned based on Coordinated Universal Time (UTC).

Workflow for replicating data by using Snap Clone

You can nondisruptively copy data from volumes to create copies for software testing or development and data protection operations. For details about Snap Clone functions, see the *Hitachi Thin Image User Guide*.



When using the REST API:

The following REST APIs can be executed to automate the operations:

- Creating a snapshot
- Adding path information to a server
- Deleting path information from a server

Note:

If you use the web-based user interface, the date and time when a snapshot was created are displayed based on the time zone set for the system of the management client. If you use the REST API, the data and time are returned based on Coordinated Universal Time (UTC).

Creating snapshots on a regular basis using the REST API

You can create snapshots by using the REST API to back up the data in operation.

You can create snapshots on a regular basis by creating scripts and using those scripts in your operation.

Before you begin

Identify the following items:

- Volume ID
- Snapshot group name
- Pool ID

  Specify the pool in which to store differential data.

Note:

- Be sure to select a pool that has a sufficient amount of free space. The snapshots, including those that already exist, might become unusable if the pool's usage rate exceeds the depletion threshold value.
- Differential data cannot be stored in a pool that has a hierarchical structure.

Procedure

1. Use the REST API to create snapshots. Specify `Snapshot` as the snapshot type.

   Request line

   ```
   POST base-URL/simple/v1/objects/snapshots
   ```

   For details, see the description of how to create snapshots by using the API.

Result

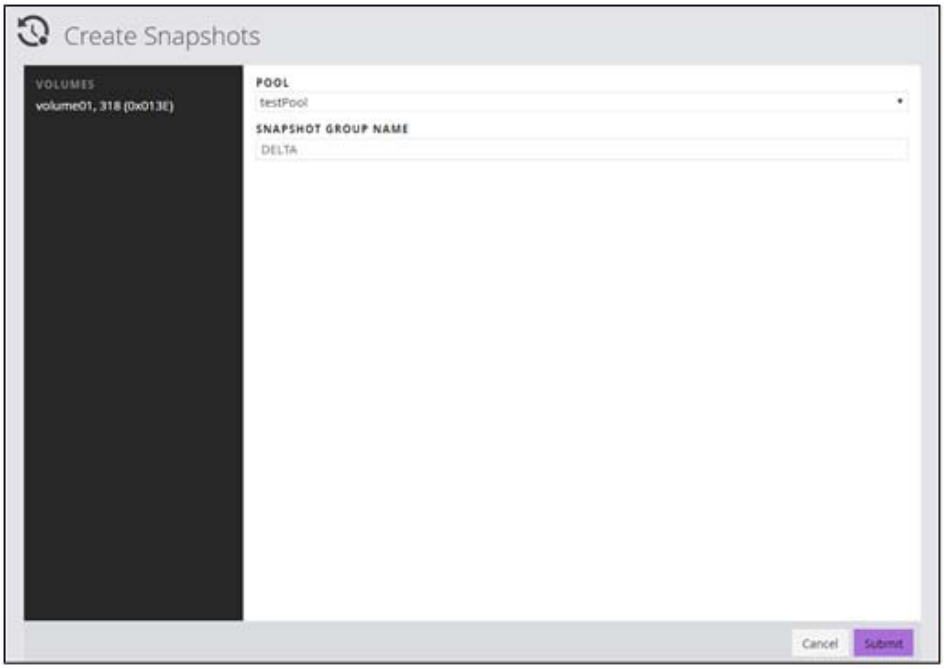
The created snapshots are displayed in the inventory of snapshots.

Creating snapshots

You can create snapshots manually. Snapshots are images of volumes at a specific point. Differential data from the original volume is stored in the pool when a snapshot is created.

Procedure

1. In the dashboard or in the navigation bar, click **Servers**.
2. Click the server name to open the server details page.
3. Select one or more volumes and click (**Create Snapshots**).



4. Specify the required information and then create the snapshots.

   Be sure to select a pool that has a sufficient amount of free space. The snapshots, including those that already exist, might become unusable if the pool's usage rate exceeds the depletion threshold value.

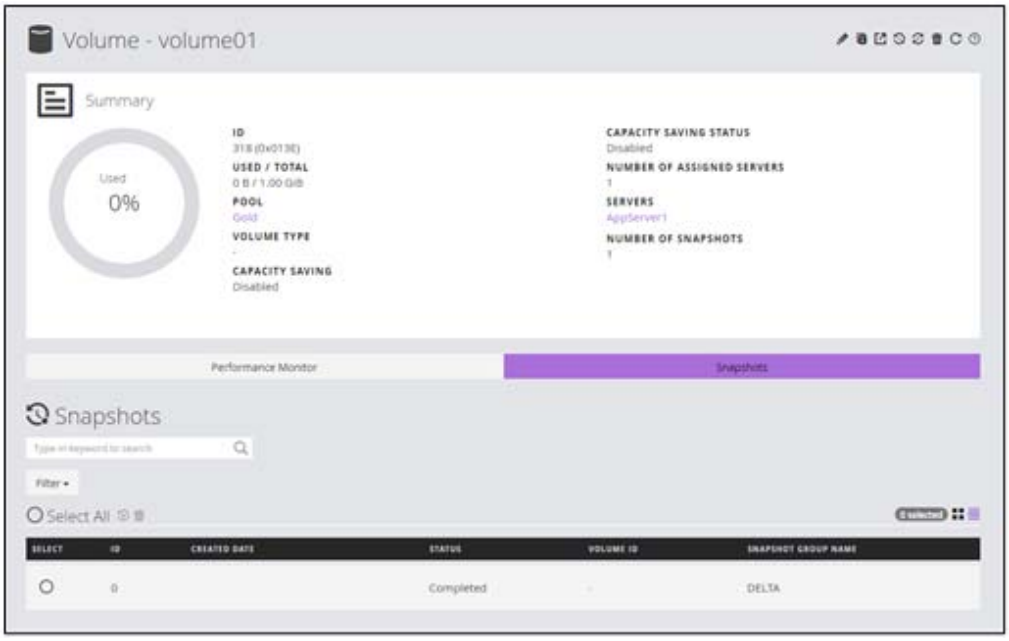
   Differential data cannot be stored in a pool that has a hierarchical structure.

Checking snapshot and Snap Clone status

You can check the status of snapshots and of Snap Clone. If the Health Status indicates that an error has occurred, you can immediately check whether the error occurred in the snapshot or the Snap Clone.

Procedure

1. In the dashboard or in the navigation bar, click **Servers**.
2. Click the server name to open the server details page.
3. Click the volume name to open the volume details page.
4. On the **Snapshots** tab, check the status of the snapshot or Snap Clone.



The statuses displayed when operations are performed on a snapshot are as follows:

Status	Description
Creating	Creation of the snapshot settings is in progress.
In Sync	Synchronization or restoration of the snapshot is complete.
Completed	Creation of the snapshot is complete.
Restoring	Restoration of the snapshot is in progress.
Deleting	Deletion of the settings and differential data for the snapshot is in progress.
Error	A failure has occurred.

The statuses displayed when operations are performed on Snap Clone are as follows:

Status	Description
Preparing	Creation of the Snap Clone settings is in progress.
Clone Ready	Creation of the settings for Snap Clone is complete.
Cloning	Replication by using Snap Clone is in progress.

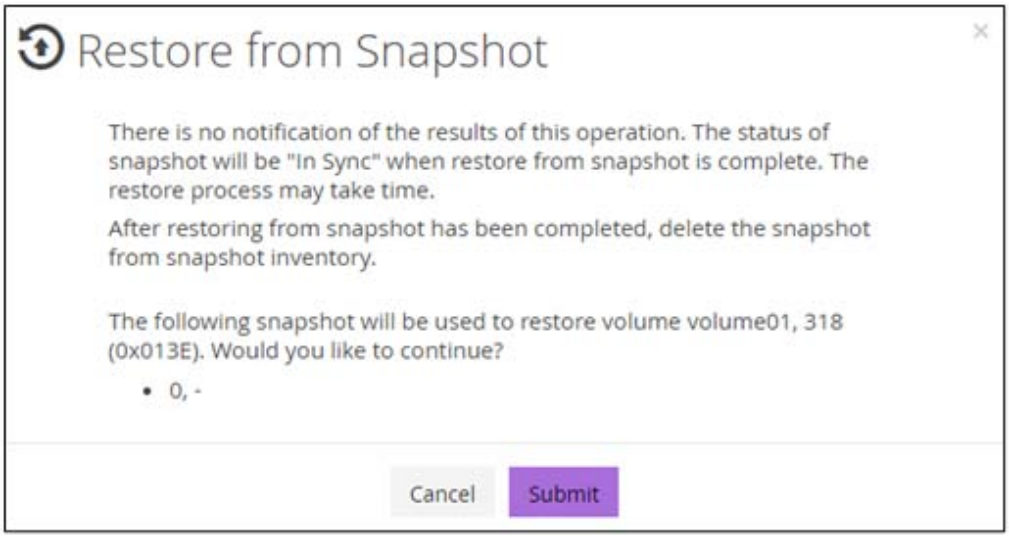
Status	Description
Removing	Removal of the settings for Snap Clone is in progress.
Error	A failure has occurred.

Restoring snapshots

In the event of volume failure, you can use the snapshots created in advance, to restore the data of a particular volume to a specific point in time.

Procedure

1. In the dashboard or in the navigation bar, click **Servers**.
2. Click the server name to open the server details page.
3. Click the volume name to open the volume details page.
4. On the **Snapshots**, display the inventory of snapshots, select the snapshot with date and time to be restored, and then click (**Restore from Snapshot**).



Result

The data is restored back to its state at the date and time when the snapshot was created.

You can check whether a volume is being restored by referring to the inventory of snapshots. When processing to restore a volume is complete, the status In Sync is displayed.

Creating and mapping snapshots using the REST API

You can create and map snapshots by using the REST API to prepare volumes to be used on other servers.

After you create and map a snapshot, you can use the snapshot as a volume to be created by using Snap Clone.

Before you begin

Identify the following items:

- Volume ID
- Snapshot group name
- Pool ID

  Specify the pool in which to store differential data.

Note:

Differential data cannot be stored in a pool that has a hierarchical structure.

Procedure

1. Use the REST API to create and map a snapshot. Specify `Mapped Snapshot` as the snapshot type.

   Request line

   ```
   POST base-URL/simple/v1/objects/snapshots
   ```

   For details, see the description of how to create snapshots by using the API.

Result

The newly created snapshots and their associated volume IDs are displayed in the snapshot inventory. The created snapshots are also displayed in the inventory of volumes and can be used as volumes to be created by using Snap Clone.

Volume names in the following format are automatically assigned to the created volumes: "Snapshot of ID: *ID-of-volume-created-from-snapshot*".

Creating volumes by using Snap Clone

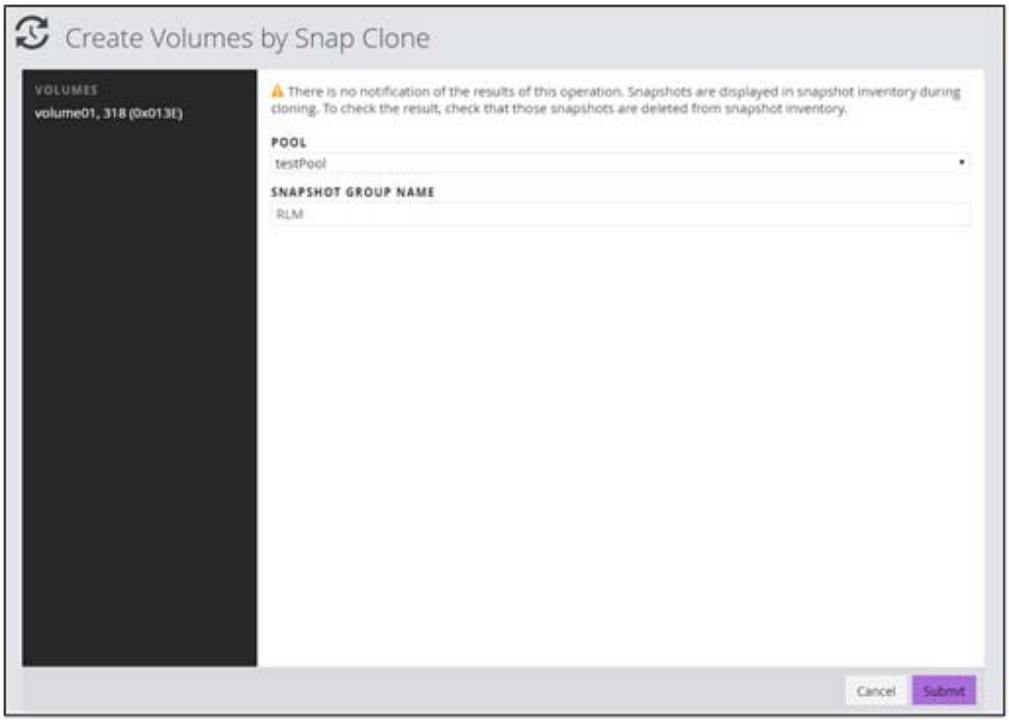
Using Snap Clone, you can create a replica or clone of volumes for software testing or development and data protection operations.

Before you begin

- Be sure to select a pool that has a sufficient amount of free space. The snapshots, including those that already exist, might become unusable if the pool's usage rate exceeds the depletion threshold value.
- Do not use Snap Clone to duplicate volumes in a pool that has a hierarchical structure.

Procedure

1. In the dashboard or in the navigation bar, click **Servers**.
2. Click the server name to open the details page.
3. Select one or more volumes and click (**Create Volumes by Snap Clone**).



4. Specify the required information and then execute Snap Clone.

Result

When the Snap Clone processing finishes, the volume is no longer displayed in the inventory of snapshots. The volumes created by Snap Clone are displayed in the inventory of volumes and can be used as normal volumes.

Volume names in the following format are automatically assigned to the created volumes: "Clone of ID: *ID-of-volume-created-from-snapshot*".

Deleting snapshots

You can delete snapshots that are no longer necessary.

Procedure

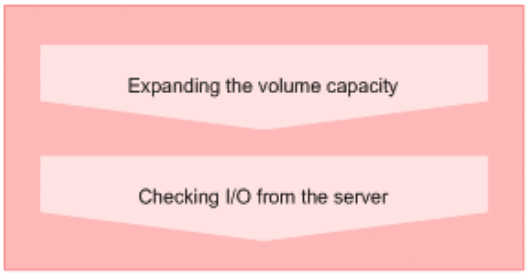
1. In the dashboard or in the navigation bar, click **Servers**.
2. Click the server name to open the server details page.
3. Click the volume name to open the volume details page.
4. On the **Snapshots**, select one or more snapshots and click (**Delete Snapshots**).

Chapter 7: Managing the capacity of volumes and pools

You can increase the capacity of a volume or a pool that has a high usage rate, and you can create multiple pools to improve fault tolerance or ensure I/O performance.

Workflow for expanding the capacity of volumes

If the capacity of a volume is insufficient, expand the capacity, as shown in the following workflow.



When using the REST API:

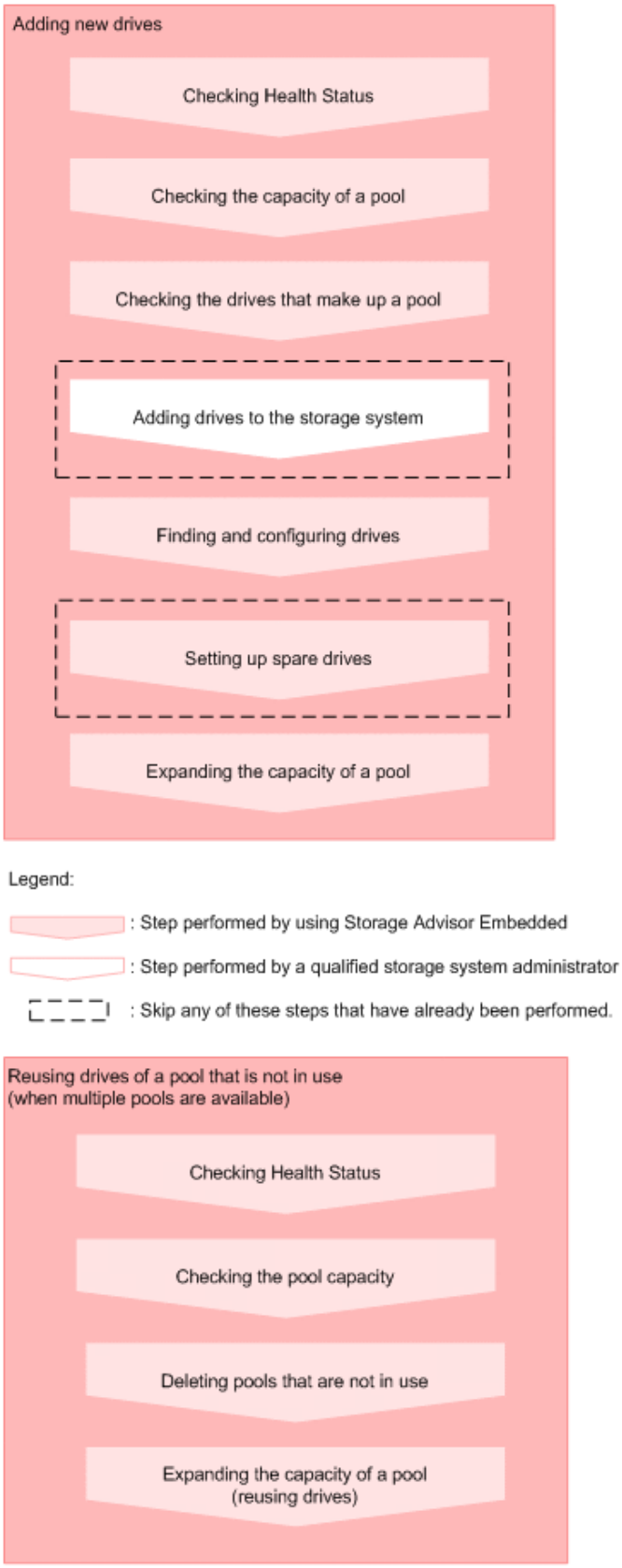
The following REST API can be executed to automate the operations:

- Expanding the capacity of a volume

Workflow for expanding the capacity of a pool

If a pool's usage rate increases, you can expand the capacity of the pool to continue with your operations.

Add new drives to expand the capacity of a pool, as shown in the following workflow. Check the drive type and the drive speed in the details page of the pool. If you are using multiple pools, you can delete a pool that is not in use and reuse the drives from that pool, as shown in the subsequent workflow.


Adding new drives
Checking Health Status
Checking the capacity of a pool
Checking the drives that make up a pool
Adding drives to the storage system
Finding and configuring drives
Setting up spare drives
Expanding the capacity of a pool
Legend:
: Step performed by using Storage Advisor Embedded
: Step performed by a qualified storage system administrator
: Skip any of these steps that have already been performed.
Reusing drives of a pool that is not in use
(when multiple pools are available)
Checking Health Status
Checking the pool capacity
Deleting pools that are not in use
Expanding the capacity of a pool
(reusing drives)

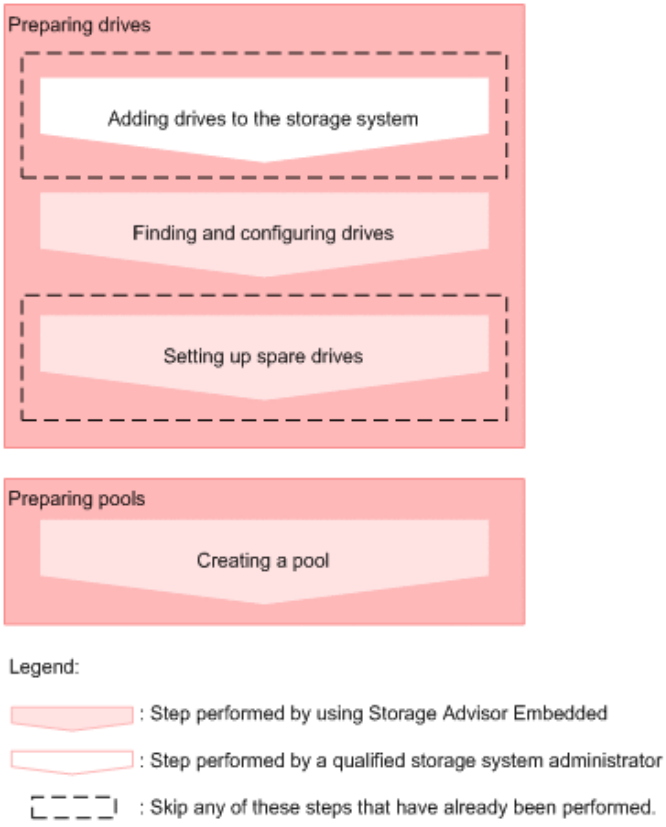
When using the REST API:

The following REST APIs can be executed to automate the operations:

- Getting pool information
- Getting information about a specific pool
- Getting drive information
- Getting information about a specific drive
- Getting storage system information
- Setting up a spare drive
- Adding a drive to a pool
- Deleting a pool

Workflow for creating additional pools

Create additional pools to improve fault-tolerance and ensure I/O performance, as shown in the following workflow.



When using the REST API:

The following REST APIs can be executed to automate the operations:

- Setting up a spare drive
- Creating a pool
- Changing pool settings

Expanding the volume capacity

You can expand the capacity of a volume to increase the capacity that can be attached to the server.

Before you begin

- Ensure that the volume is attached to a server.
- Identify the following items:
  - Server name
  - Volume name

Procedure

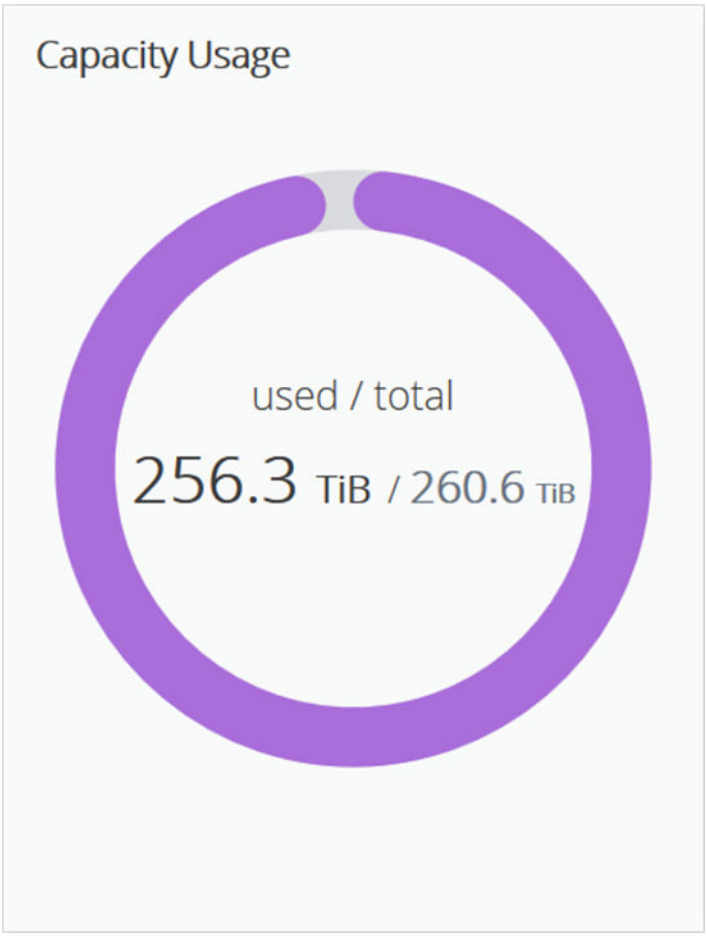
1. In the dashboard or in the navigation bar, click **Servers**.
2. Click the server name to open the server details page.
3. Select one or more volumes and click (**Expand Volumes**).
4. Expand the capacity of the selected volumes by specifying the amount of capacity to be added or by specifying the total capacity after expansion.

Checking the pool capacity

You can check the capacity of the pools managed by Storage Advisor Embedded. If you are using multiple pools, you can check the capacity of each pool and the total capacity of all pools.

Procedure

1. Check the capacity of pools in the following pages:
   - The total capacity of the pools: View **Capacity Usage** on the dashboard



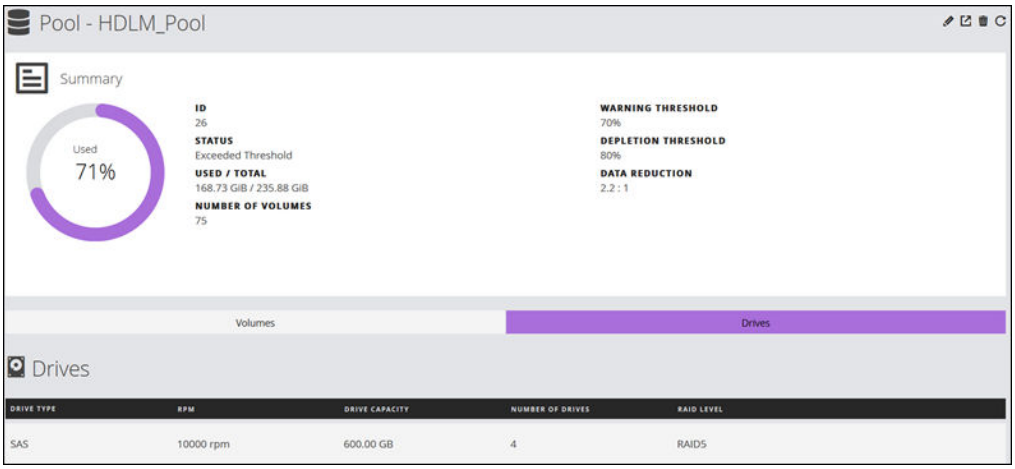
   - The capacity of each pool: Click the name of the pool in the pool inventory to display the pool details page

Checking the drives that make up a pool

You can check the information about the drives that make up a pool (such as the drive type and the RAID level).

Procedure

1. In the navigation bar, click **Pools**.
2. Click the pool name to open the pool details page.
3. Click **Drives**, and then check the drive type, the RAID level, and other information about the drives that make up the pool.



Expanding the capacity of a pool

If a pool's usage rate becomes high, you can continue using the pool by expanding the capacity of the pool.

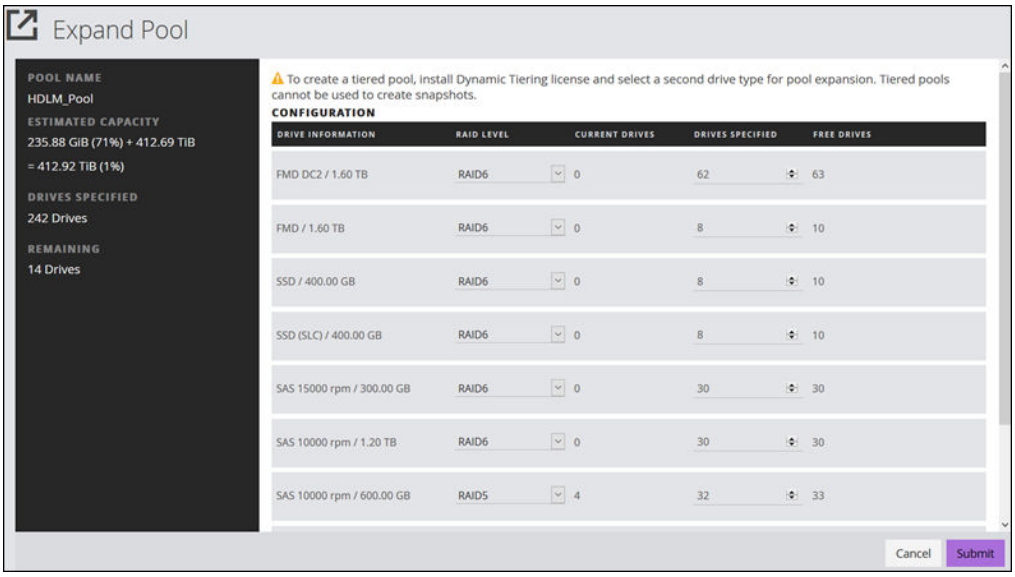
Pools for thin provisioning are virtual areas in a storage system that are associated with multiple drives. Thin provisioning provides virtual volumes to a server and uses the actual capacity from a pool when a server makes a write request.

Storage Advisor Embedded displays a recommended configuration, from among various configurations supported by the storage system. This configuration consists of the combination of drives that allow drive space to be used most effectively.

For storage systems with a valid Dynamic Tiering license, if you include drives that have different drive information (drive type, drive rotation speed, and drive capacity), the expanded pool will be a tiered pool where tiers are arranged by performance. In a tiered pool, data is automatically stored in optimum tiers.

Procedure

1. In the navigation bar, click **Pools**.
2. Click (**Expand Pool**) for the pool being expanded.



3. Click **Submit** to accept the recommended configuration and create a pool.

   To expand a pool by specifying its configuration, change the number of drives. Next, click **Check** to check the available combinations, and then click **Submit**.

Checking the volume capacity

You can check the capacity of a volume managed by Storage Advisor Embedded.

Procedure

1. In the dashboard or in the navigation bar, click **Servers**.
2. Click the server name to open the server details page.
3. Click the volume name to open the volume details page.

4. Check the amount of used capacity, the usage rate, and other settings displayed in **Summary**.

> **Note:**
>
> The volume capacity of a volume of the type **Snapshot** is not used. Data is stored directly in a pool.

Deleting pools

You can delete pools that are no longer used. Make sure no volumes have been created for the pool being deleted.

Procedure

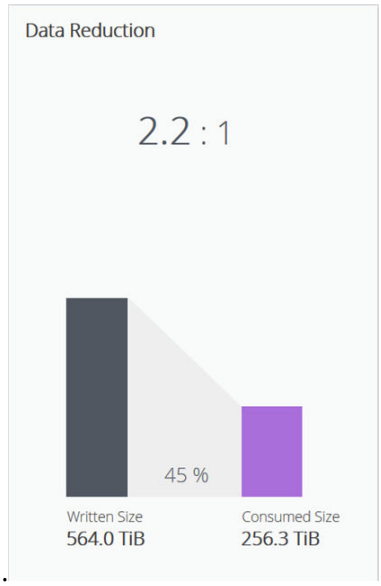
1. In the navigation bar, click **Pools**.
2. Select one or more pools and click (**Delete Pools**).

Checking data reduction achieved using the capacity saving feature

You can check how effectively the storage system is being utilized when the capacity saving feature (deduplication and compression) is enabled in the storage system.

Procedure

1. In the following windows, check the amount of data reduction achieved by using the capacity saving feature (deduplication and compression) for the storage system:
   - Data reduction for the entire storage system: View the **Data Reduction** report on the dashboard.



   - Data reduction for each pool: Click a pool name in the **Pools** inventory to display the details page.

Chapter 8: Configuring a global-active device environment

When configuring an environment for using global-active device (GAD), use Storage Advisor Embedded to configure remote paths and quorum disks.

Workflow for configuring a global-active device environment

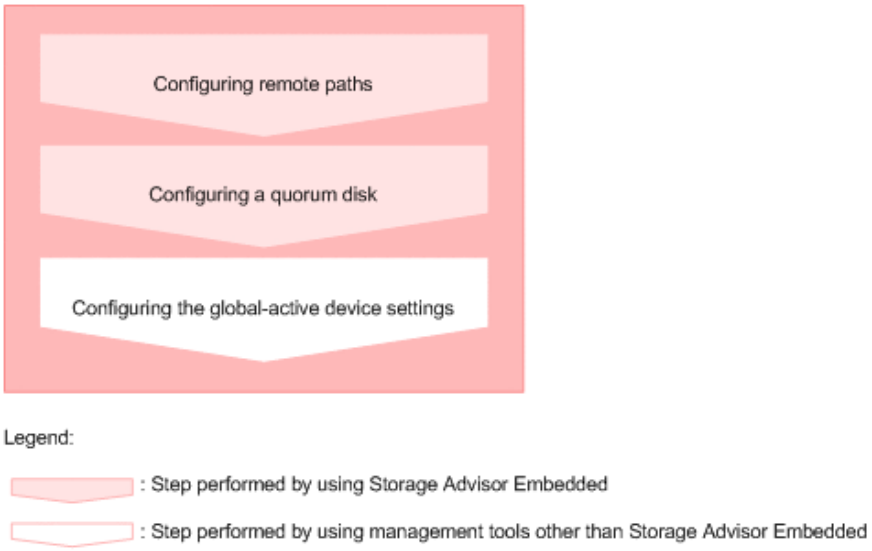
With global-active device, you can achieve data redundancy for volumes between two storage systems, and provide a high availability environment.

Data written to a volume of one storage system is automatically synchronized to a volume of the other storage system. The user can then use global-active device without having to be aware of the configuration of the volumes accessed by the server. For details about global-active device, see the *Global-Active Device User Guide*.

You can use Storage Advisor Embedded to configure the remote paths and the quorum disk required for using global-active device. These settings must be configured on both storage systems of the global-active device.

Note:

When using Storage Advisor Embedded to configure the environment, the remote paths and the external path for the quorum disk must be connected using Fibre Channel.



When using the REST API:

The following REST APIs can be executed to automate the operations:

- Creating a remote connection

  For details about how to use the API request for creating remote connection, see the *REST API Reference Guide*.

  The request line is as follows:

  ```
  POST base-URL/v1/objects/remotepath-groups
  ```

- Creating an external volume
- Registering information about a quorum disk

Configuring remote paths

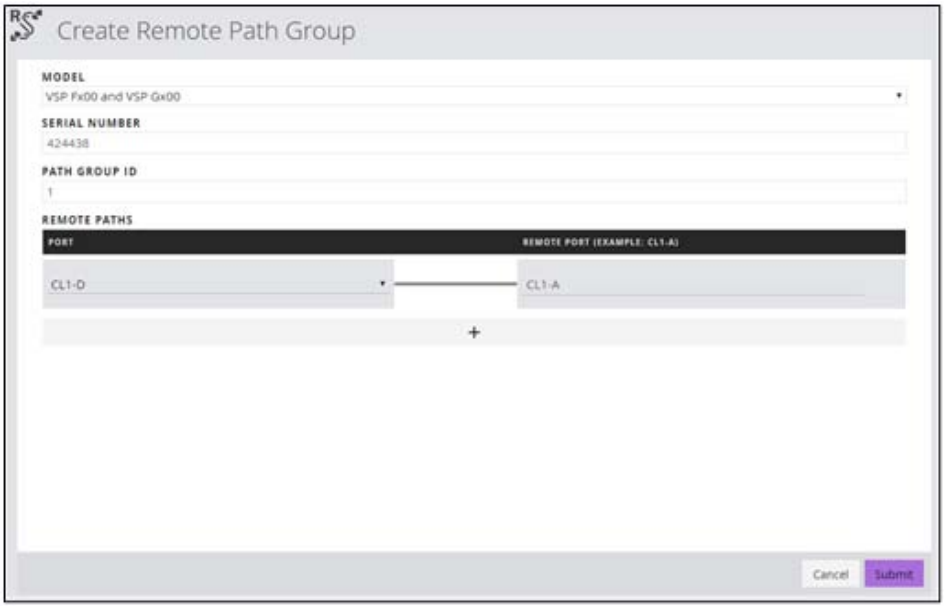
You can configure remote paths between the two storage systems that make up a global-active device environment. Because a remote path is required to send data to the connection-destination storage system, you need to configure remote paths for both storage systems so they can send data to each other.

Before you begin

- Connect physical paths between the two storage systems that make up the global-active device environment.
- Identify the following items:
  - The model and serial number of the connection-destination storage system
  - The ID of the path group
  - The port to be used on the connection-source storage system and the port of the connection-destination storage system

Procedure

1. In the navigation bar, click (**Settings**), and then select **Remote Path Groups**.
2. Click the plus sign (**+**).



3. Specify the required items, and then configure the remote path.
4. Click the ID of the path group to open the details page. Make sure that the status of the remote path you configured is **Normal**.

> **Note:**
>
> If the status of the remote path is not **Normal**, see the *Global-Active Device User Guide*.

Configuring a quorum disk

To use global-active device, the two storage systems that make up the global-active device environment must be able to access the quorum disk in the external storage system. If a path or a storage system fails, the quorum disk determines the storage system in which I/O operations from the server will continue.

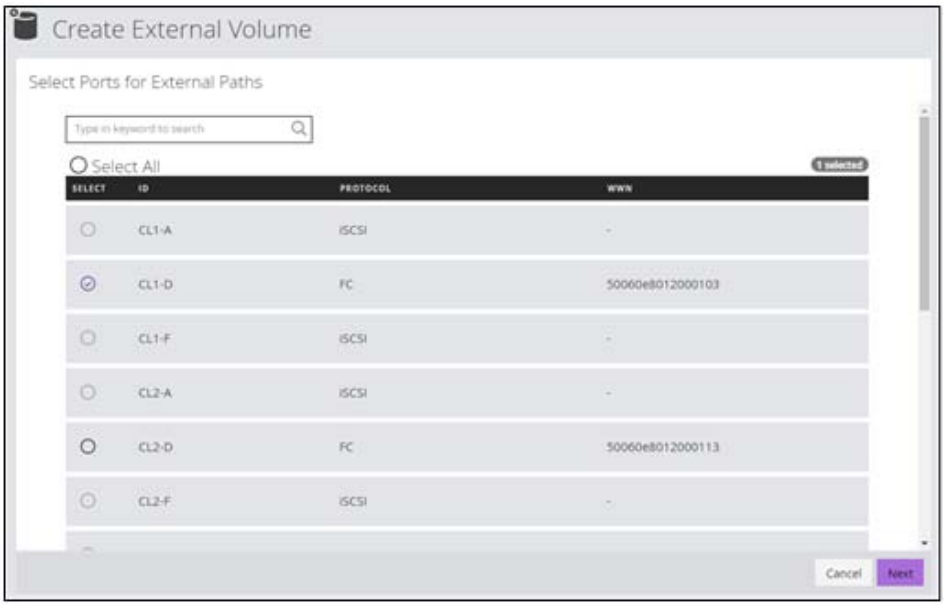
Before you begin

- Ensure that the two storage systems that make up the global-active device environment and the external storage system in which the volume used as the quorum disk is configured are connected by physical paths.
- Ensure that the volume used as the quorum disk is assigned a port on each of the two storage systems that make up the global-active device environment.

- Identify the following items required to configure the external volume:
  - The port to be used for external connection
  - The model and serial number of the external storage system
  - The WWN of the port of the external storage system that is connected with the port for external connection
  - The LUN of the volume used as the quorum disk
  - The name of the external volume
  - The ID of the external parity group
  - The ID of the external path group
- Identify the following information required to configure the quorum disk:
  - The ID of the quorum disk
  - The models and serial numbers of the paired storage system that shares the quorum disk

Procedure

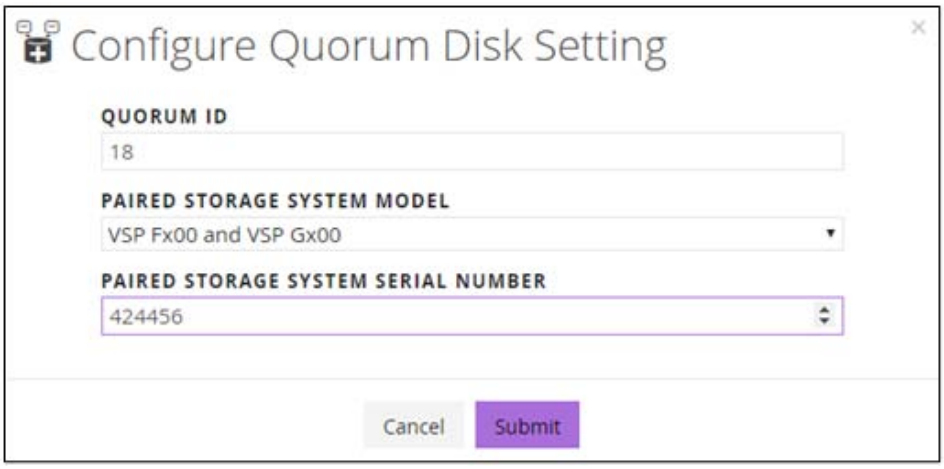
1. In the navigation bar, click (**Settings**), and then select **External Volumes**.
2. Click the plus sign (**+**).



3. Select the port for external connection, and then click **Next**.
4. Select the external path to be used, and then click **Next**.
5. Select the LUN of the volume used as the quorum disk, and specify the name of the external volume, the ID of the external parity group, and the ID of the external path group. Click **Submit**.

   Information about the newly configured external volume is displayed in the list of external volumes.

6. Click (**Configure Quorum Disk Setting**) for the external volume for which the quorum disk is to be configured.



7. Specify the required items, and then click **Submit**.

Adding remote paths

Add remote paths to an already-configured remote path group to configure redundant remote paths between the two storage systems that make up a global-active device environment.

Before you begin

Identify the following items:

- The ID of the path group
- The port to be used on the connection-source storage system and the port of the connection-destination storage system

Procedure

1. In the navigation bar, click (**Settings**), and then select **Remote Path Groups**.
2. Click the path group ID of the remote path group to which you want to add the remote path.
3. Click (**Add Remote Paths**).

4. Specify the required items, and then add the remote path.

   Make sure that the status of the remote path you added is **Normal**.

   Note:

   If the status of the remote path is not **Normal**, see the *Global-Active Device User Guide*.

Removing remote paths

Remove remote paths from a remote path group. You cannot remove all of the remote paths from a remote path group. To remove all of the remote paths in a remote path group, remove the remote path group itself.

Before you begin

Identify the ID of the path group.

Procedure

1. In the navigation bar, click (**Settings**), and then select **Remote Path Groups**.
2. Click the path group ID of the remote path group from which the remote path is to be removed.
3. From the list of remote paths, select one or more remote paths, and then click (**Remove Remote Paths**).

Removing remote path groups

When you no longer need to use a global-active device, remove the paths configured between the two storage systems for which global-active device is used.

Before you begin

Identify the ID of the path group of the remote path groups to be removed.

Procedure

1. In the navigation bar, click (**Settings**), and then select **Remote Path Groups**.
2. Select one or more remote path groups and click (**Delete Remote Path Groups**).

Changing the name of the external volume of a quorum disk

You can change the external volume name set for a quorum disk.

Procedure

1. In the navigation bar, click (**Settings**), and then select **External Volumes**.
2. Click (**Edit External Volume**) for the quorum disk whose external volume name you want to edit.
3. Enter a different external volume name.

Adding external paths to a quorum disk

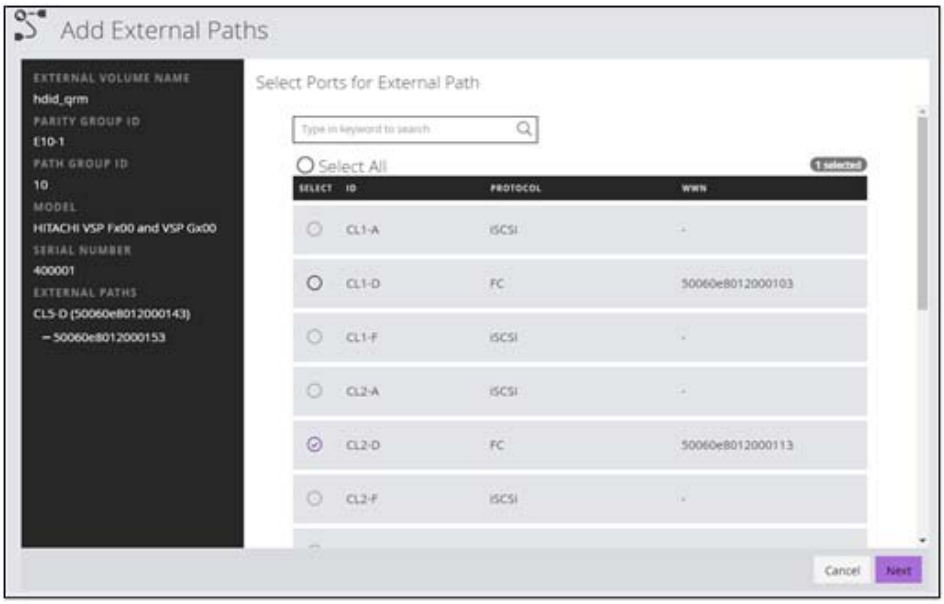
To configure redundant paths to a quorum disk, add external paths to the quorum disk.

Before you begin

- Ensure that the two storage systems that make up the global-active device environment and the external storage system in which the quorum disk is configured are connected by physical paths.
- Ensure that the quorum disk is assigned a port on each of the two storage systems that make up the global-active device environment.
- Identify the following items:
  - The port to be used for external connection
  - The WWN of the port of the external storage system that is connected with the port for external connection

Procedure

1. In the navigation bar, click (**Settings**), and then select **External Volumes**.
2. Click (**Add External Paths**) for the external volume to which you want to add a path.



3. Select the port for external connection, and then click **Next**.
4. Select the external path to be used, and then click **Submit**.
5. Click the volume name of the quorum disk to which you added an external path. On the volume details page, verify the external path you added.

   If other external volumes are included in the same path group, external paths are also added to those volumes. To check information about volumes other than those for which you performed the operation, refresh the list of external volumes.

Removing external paths to a quorum disk

Remove external paths set to a quorum disk.

Procedure

1. In the navigation bar, click (**Settings**), and then select **External Volumes**.
2. Click the volume name of the quorum disk whose external path you want to delete.
3. Select one or more external paths, and then click (**Remove External Paths**).

   If other external volumes are included in the same path group, external paths to those volumes are also removed. To check information about volumes other than those for which you performed the operation, refresh the list of external volumes.

Disabling a quorum disk configuration

When you finish using a global-active device, disable the settings of the quorum disk that is no longer required for the two storage systems for which global-active device is used. Delete the external volume used as the quorum disk if it is no longer required.

Before you begin

Identify the external volume name of the quorum disk for which settings are to be disabled.

Procedure

1. In the navigation bar, click (**Settings**), and then select **External Volumes**.
2. Select one or more volumes used as the quorum disk whose settings you want to disable, and then click (**Remove Quorum Disk Setting**).
3. Make sure that the target volumes are correct, and then click **Submit**.
4. To delete external volumes, click (**Delete External Volume**) for the external volume to be deleted.
5. Make sure that the target external volume is correct, select either **Delete with Destaging** or **Delete without Destaging**, and then click **Submit**.

   If you select **Delete with Destaging**, the processing to write data stored in cache memory to the external volume (destage processing) is performed, and after the connection is disabled, the external volume is deleted. If you select **Delete without Destaging**, the destage processing is not performed, and the external volume is forcibly deleted.

Chapter 9: Checking the operating status of volumes

You can check the volume operation status.

Checking the operating status of volumes

You can check whether any irregularities occur during your daily operations. For example, you can check for any volumes for which no I/O is issued, or for volumes for which more I/O than usual is issued. You can also check the operating status of volumes when you receive an inquiry from a server administrator.

Checking the operating status of volumes in the performance graphs

You can display and check the operating status of each volume with respect to IOPS, response time (time required to respond to a server request), and transfer rate. You can also download the contents of each performance graph as a CSV file.

To export the contents of performance graphs to a CSV file by using Windows Server and Internet Explorer, either disable Internet Explorer Enhanced Security Configuration, or use Google Chrome.

Procedure

1. In the dashboard or in the navigation bar, click **Servers**.
2. Click the server name to open the server details page.
3. Click the volume name to open the volume details page.
4. On the **Performance Monitor**, check the operating status based on the graphs of IOPS, response time, and transfer rate.

> **Note:**
>
> Performance reports may not show the current information if performance data is not obtained from the storage system. This issue should resolve itself after some time when the storage system is less busy.

Chapter 10: Monitoring storage system problems

You can identify hardware failures, problems in pools, and problems in snapshots by checking Health Status and taking appropriate action. You can also receive information about problems via an email message or SNMP trap notification. If a failure is identified, you can check SIMs in the alert list from the maintenance utility and take appropriate action.

Monitoring the storage system

You can check whether an error has occurred in a storage system by checking the LED alert for Health Status. Failure information can be also sent via an email message or SNMP trap notification. If a failure is identified, you can check SIMs in the alert list from the maintenance utility and take appropriate action. If the usage rate of a pool exceeds a threshold, or if an error occurs during the operation of a snapshot, you can check the messages in the Health Status message box to identify the problem, and then take specific action.

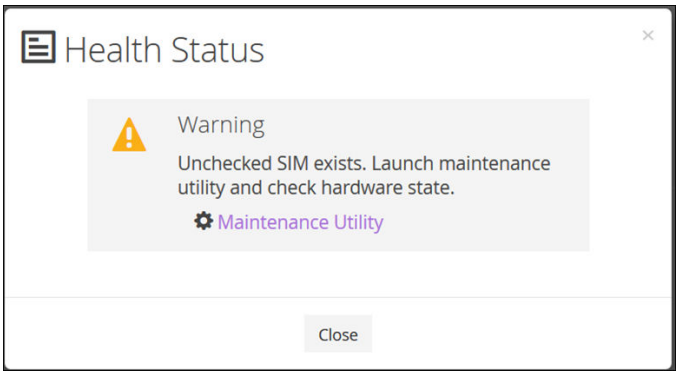
Checking health status

You can check for problems in the storage system. If a problem occurs in a pool or snapshot, take action according to the displayed message.

Procedure

1. In the navigation bar, click **Error** or **Warning**.

   If there is no problem, **Normal** displays.



2. Review any message displayed to identify pools or snapshots where a problem has occurred, and then take appropriate action.

Checking storage system alerts in the maintenance utility

When a failure detection notification for the storage system is sent via Health Status, an email message, or an SNMP trap message, you can check the alert information by using the maintenance utility, and then take appropriate action.

Before you begin

The logged-in user is part of the Maintenance user group (a built-in user group).

Procedure

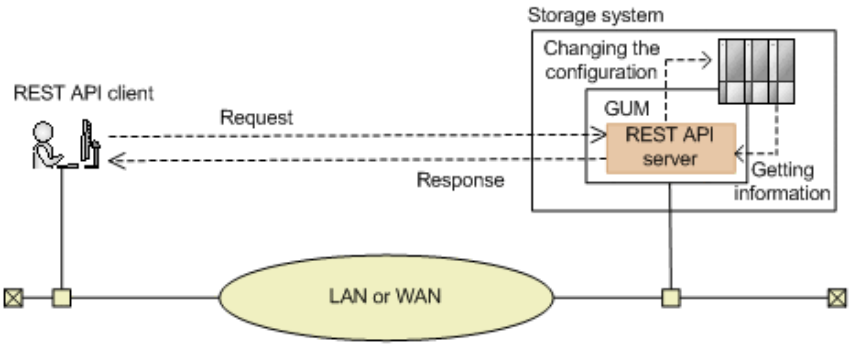
1. In the navigation bar, click (**Settings**), and then select **Maintenance Utility** to open the maintenance utility.
2. Click the **Alerts** tab to display the list of alerts.
3. Check the alerts and then take appropriate action based on the notification information.
4. In the maintenance utility, click **Log Out**.

Chapter 11: Overview of the REST API

Review general information about API requests, such as the basic system configuration for the REST API, how to specify resources to be managed, the information needed to execute an API request, and the information that is output when an API request is executed.

System configuration of REST API

The following figure shows the basic system configuration of REST API.



Storage system

By using the REST API, you can get information from the storage system or change the configuration of the storage system.

GUM (Gateway for Unified Management)

This is the computer that has basic management functions for the storage system. You can manage the storage system from an external device through communications with GUM.

GUM exists in each controller, controller 1 (CTL1) and controller 2 (CTL2).

REST API server

This component acts as a server to receive REST API requests from REST API clients, issue orders to the storage system, and then return the execution results to the REST API clients.

The REST API server is on the GUM of the storage system.

REST API clients

The REST API clients issue requests to the REST API server. The term "REST API client" refers to software or scripts that use the REST API.

Note:

In addition to the API requests described in this manual, API requests are available for getting more detailed information and for changing the configuration of storage systems. For information on how to use these types of API requests, see Hitachi Vantara Knowledge.

Requirements for SSL communications

You can use SSL communication between the REST API clients and the REST API server.

For SSL communications between REST API clients and the REST API server, the server certificate for HTTPS installed in the GUM is used. By default, this is a self-signed certificate. If the API client only allows communication with the REST API server installed with a signed certificate, then do one of the following:

- Change the certificate of the storage system to a server certificate trusted by a certificate authority, such as VeriSign.

  For details on how to obtain a certificate signed by a certificate authority and how to update the certificate on the GUM, see the *System Administrator Guide* for your storage system.

- Correct the client programs to avoid errors.

  The approach taken to correct client programs to avoid errors varies by programming language.

  For example, if the Requests library is used with Python, verification of the server certificate can be omitted if `verify=False` is specified when a request is issued.

Note:

TLS version 1.2 can be used for SSL communication between REST API clients and the REST API server.

The following encryption methods (cipher suites) can be used:

- `ECDHE-RSA-AES256-GCM-SHA384`
- `ECDHE-RSA-AES128-GCM-SHA256`
- `AES256-GCM-SHA384`
- `AES256-SHA256`
- `AES128-GCM-SHA256`
- `AES128-SHA256`

Roles and access permissions of users executing API requests

To use the REST API to perform operations on storage system resources, users who issue API requests must have the appropriate roles (execution permissions) and access permissions for the target resources.

To execute REST API requests, users must also have the User Maintenance role.

Before using the REST API, register the user into a user group that has the necessary roles. Users who can operate the Storage Advisor Embedded web-based user interface can issue all API requests.

For details on the roles required for executing API requests, see information about the API requests.

Specifying resources to be managed (URL)

In the REST API, the resources to be managed must be specified in URL format.

To specify resources to be managed, specify the URL in the following format:

```
protocol://host:port/application/version/domain
```

- `protocol`

  Specify `https` or `http`.

  We recommend specifying `https` for security.

- `host`

  Specify the GUM IP address or a host name that can be resolved. Specify either the GUM IP address for controller 1 (CTL1) or the GUM IP address for controller 2 (CTL2).

- `port`

  Specify the port number to be used for communication.

  The default port number is `443` (for SSL communication) and `80` (for non-SSL communication). The port number can be omitted if the default port number is used for communications.

- `application`

  Specify `ConfigurationManager/simple`.

Note:

In this document, *protocol*`://`*host*`:`*port*`/ConfigurationManager` is indicated as a base URL.

- `version`

  Specify the version of the REST API. Currently, only `v1` can be specified.

- `domain`

  Specify the domain. Currently only `objects` can be specified.

  The URL formats are as follows.

  Under `objects`, specify the type of object on which operations will be performed:

  - `command-status`
  - `drives`
  - `external-path-groups`
  - `external-parity-groups`
  - `external-volumes`
  - `health-status`
  - `pools`
  - `ports`
  - `quorum-disks`
  - `servers`
  - `snapshot-groups`
  - `snapshots`
  - `storages`
  - `volumes`
  - `volume-server-connections`

The following tables describe the URL formats supported for each object type.

Resources for the object type `command-status`:

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/command-status/`*`object-ID`*	GET	Synchronous	Obtains status information about an API function that performs asynchronous processing

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
	DELETE	Synchronous	Clears status information about an API function that performs asynchronous processing

Resources for the object type `drives`:

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/drives`	GET	Synchronous	Obtains drive information
`/simple/v1/objects/drives/actions/set-spare/invoke`	POST	Asynchronous	Sets a specific spare drive
`/simple/v1/objects/drives/actions/release-spare/invoke`	POST	Asynchronous	Disables the settings of a specific spare drive
`/simple/v1/objects/drives/`*`object-ID`*	GET	Synchronous	Obtains information about a specific drive

Resources for the object type `external-path-groups`:

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/external-path-groups`	GET	Synchronous	Obtains information about external path groups
`/simple/v1/objects/external-path-groups/`*`object-ID`*	GET	Synchronous	Obtains information about a specific external path group

Resources for the object type `external-parity-groups`:

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/external-parity-groups`	GET	Synchronous	Obtains information about external parity groups
`/simple/v1/objects/external-parity-groups/`*`object-ID`*	GET	Synchronous	Obtains information about a specific external parity group

Resources for the object type `external-volumes`:

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/external-volumes`	GET	Synchronous	Obtains information about external volumes
	POST	Asynchronous	Creates an external volume
`/simple/v1/objects/external-volumes/`*`object-ID`*	GET	Synchronous	Obtains information about a specific external volume
	PATCH	Synchronous	Changes the nickname of an external volume

Resources for the object type `health-status`:

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/health-status`	GET	Synchronous	Obtains the operating status of storage resources

Resources for the object type `pools`:

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/pools`	GET	Synchronous	Obtains pool information
	POST	Asynchronous	Creates a pool
`/simple/v1/objects/pools/`*`object-ID`*	GET	Synchronous	Obtains information about a specific pool
	PATCH	Synchronous	Changes pool settings
	DELETE	Asynchronous	Deletes a pool
`/simple/v1/objects/pools/`*`object-ID`*`/actions/expand/invoke`	POST	Asynchronous	Adds drives to a pool

Resources for the object type `ports`:

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/ports`	GET	Synchronous	Obtains port information
`/simple/v1/objects/ports/`*`object-ID`*	GET	Synchronous	Obtains information about a specific port
	PATCH	Synchronous	Changes port settings

Resources for the object type `quorum-disks`:

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/quorum-disks`	POST	Asynchronous	Registers information about a quorum disk
`/simple/v1/objects/quorum-disks/`*`object-ID`*	GET	Synchronous	Obtains information about a specific quorum disk

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
	DELETE	Asynchronous	Deletes information about a quorum disk

Resources for the object type `servers`:

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/servers`	GET	Synchronous	Obtains server information
	POST	Asynchronous	Registers a server
`/simple/v1/objects/servers/`*`object-ID`*	GET	Synchronous	Obtains information about a specific server
	PATCH	Asynchronous	Changes server settings
	DELETE	Asynchronous	Deletes server information
`/simple/v1/objects/servers/`*`object-ID`*`/hbas`	GET	Synchronous	Obtains information about the WWN (iSCSI name) of the HBA for the specified server
	POST	Asynchronous	Adds information about the WWN (iSCSI name) of the HBA for the specified server
`/simple/v1/objects/servers/`*`object-ID`*`/hbas/`*`object-ID`*	GET	Synchronous	Obtains information about the server specified by the server ID and the WWN (iSCSI name) of the HBA
	DELETE	Asynchronous	Deletes information from the server specified by the server ID and the WWN (iSCSI name) of the HBA
`/simple/v1/objects/servers/`*`object-ID`*`/paths`	GET	Synchronous	Obtains server path information
	POST	Asynchronous	Adds pass information to a server

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/servers/`*`object-ID`*`/paths/`*`object-ID`*	GET	Synchronous	Obtains path information for a specific server
	DELETE	Asynchronous	Deletes path information from the server
`/simple/v1/objects/servers/`*`object-ID`*`/target-iscsi-ports`	GET	Synchronous	Obtains information about iSCSI targets
`/simple/v1/objects/servers/`*`object-ID`*`/target-iscsi-ports/`*`object-ID`*	GET	Synchronous	Obtains information about a specific iSCSI target
	PATCH	Asynchronous	Changes the settings of an iSCSI target

Resources for the object type `snapshot-groups`:

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/snapshot-groups`	GET	Synchronous	Obtains snapshot group information
`/simple/v1/objects/snapshot-groups/`*`object-ID`*	GET	Synchronous	Obtains information about a specific snapshot group
	DELETE	Asynchronous	Deletes a snapshot group

Resources for the object type `snapshots`:

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/snapshots`	GET	Synchronous	Obtains snapshot information
	POST	Asynchronous	Creates a snapshot

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/snapshots/`*`object-ID`*	GET	Synchronous	Obtains information about a specific snapshot
	DELETE	Asynchronous	Deletes a snapshot
`/simple/v1/objects/snapshots/`*`object-ID`*`,` *`object-ID`*`/actions/map/invoke`	POST	Asynchronous	Maps a snapshot
`/simple/v1/objects/snapshots/`*`object-ID`*`,` *`object-ID`*`/actions/restore/invoke`	POST	Asynchronous	Restores a snapshot

Resources for the object type `storages`:

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/storages`	GET	Synchronous	Obtains storage system information

Resources for the object type `volumes`:

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/volumes`	GET	Synchronous	Obtains volume information
	POST	Asynchronous	Creates a volume
`/simple/v1/objects/volumes/`*`object-ID`*	GET	Synchronous	Obtains information about a specific volume
	PATCH	▪ Synchronous ▪ Asynchronous	Changes volume settings

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
	DELETE	Asynchronous	Deletes a volume
`/simple/v1/objects/volumes/`*`object-ID`*`/actions/expand/invoke`	POST	Synchronous	Expands the capacity of a volume

Resources for the object type `volume-server-connections`:

URL formats (excluding the base URL)	HTTP method	Processing method	Operations
`/simple/v1/objects/volume-server-connections`	GET	Synchronous	Obtains information about the connection between a volume and a server
	POST	Asynchronous	Connects a volume and a server
`/simple/v1/objects/volume-server-connections/`*`object-ID`*	GET	Synchronous	Obtains information about the connection between a volume and a specific server
	DELETE	Asynchronous	Disconnects a volume and a specific server

The REST API has the following two processing methods:

- Synchronous processing

  The processing results are returned as the response.
- Asynchronous processing

  The object information and the HTTP status code (`202`), which indicates that the processing has been accepted, are returned as the response.

Note:

A maximum of 16 requests can be executed at the same time.

Specifying an object ID

An object ID is used to uniquely identify a resource.

An object ID is used when specifying a specific resource in a URL. To specify an object ID, execute the GET operation and then obtain the object ID from the execution results.

Example: If the object ID of a volume is `100`

```
volumes/100
```

Note:

If a reserved character specified in RFC 3986 is included in the value of an attribute obtained by executing the GET operation to obtain an object ID, the REST API server returns an encoded value. To use the object ID obtained by performing the GET operation in a request of another operation, use the object ID without decoding it.

Supported HTTP methods

Specify operations to be performed on resources in the HTTP method.

The REST API supports the following HTTP methods.

HTTP method	Description
GET	This method gets object information. Alternatively, this method gets a list of objects. For example, this method can obtain a list of pools. When information about multiple objects is obtained, the information is not sorted by object ID (`id`), such as the pool ID or volume ID. For this reason, filter the information to be obtained by specifying query parameters. Alternatively, if you know the object IDs of the objects for which you want to get information, run the API request for obtaining information about specific objects.
POST	This method performs operations such as creating, adding, or expanding objects. For example, this method can create pools.
PATCH	This method partially changes the attributes or the state of an object. For example, this method can change the pool threshold.
DELETE	This method deletes objects. For example, this method can delete pools.

The methods that can be used vary depending on the objects. For details, see the descriptions for each API function.

User authentication

User authentication is required to operate the storage system. To perform user authentication, the Authorization header must be specified.

Authentication by user ID and password

When you create a session, specify authentication information in the following format in the Authorization header:

```
Authorization: Basic authentication-information
```

authentication-information

Specify a Base64-encoded character string in which the user ID and password are concatenated with a colon (`:`). Use the user ID and password of a user account that can perform operations on storage system resources.

When using the REST API, you can use the characters in the following table for the user ID and password.

Item	Number of characters	Specifiable characters
User ID	1 through 63 characters	▪ Alphanumeric characters ▪ The following symbols: Hyphens (-), periods (`.`), forward slashes (`/`), at marks (`@`), underscores (_)
Password	6 through 63 characters	▪ Alphanumeric characters ▪ The following symbols: Commas (`,`), hyphens (-), periods (`.`), forward slashes (`/`), at marks (`@`), underscores (_)

The following is an example of the Authorization header where the user ID is `sample-user`, and the password is `sample-password`:

```
Authorization: Basic c2FtcGxlLXVzZXI6c2FtcGxlLXBhc3N3b3Jk
```

Authentication by sessions

In the Authorization header, specify the authentication information for the session in the following format:

```
Authorization: Session token
```

Token

A token is authentication information that is returned after a session is created. This information is used to determine whether the request was issued from an authorized user.

Example of the Authorization header:

```
Authorization: Session b74777a3f9f04ea8bd8f09847fac48d3
```

In the REST API, session-based user authentication is performed. A session is always generated first when a REST API client accesses the REST API server and starts an operation. In the request that generates a session, the user ID and password are used for authentication to access the storage system. After a session is created, specify session information for the Authorization header to perform authentication based on the session information.

For details about how to use the API request for generating sessions, see *REST API Reference Guide*

Request line:

```
POST base-URL/v1/objects/sessions
```

Response message:

```
{
  "token": "b74777a3f9f04ea8bd8f09847fac48d3",
  "sessionId": 3
}
```

Request headers

Request headers are request messages sent from REST API clients to the REST API server.

The following table lists the request headers supported by the REST API.

Header	Category	Description
Accept	Optional	Specifies the media type of the response. Specifiable value: `*/* (json)` Default value: `*/* (json)`
Accept-Language	Optional	Specifies the language of the messages to be received by the client.

Header	Category	Description
		Specify `ja` to set the language to Japanese. If any language other than Japanese is specified, or if this header is omitted, the language will be set to English (`en`).
Content-Type	Optional	Specifies the media type of the request body. Specifiable value: `application/json` Default value: `application/json`
Content-Length	Optional	Specifies the size of the request body. You can specify the Content-Length header when specifying a request body. This header is automatically assigned depending on the specifications of the client software. Specifiable value: Specify the header, in bytes. Default value: None.
Authorization	Required	Specifies the authorization method and authorization credentials. Specify the header in one of the following formats: ▪ If a session is to be generated: `Basic` *`authorization-credentials`* For the authorization credentials, specify the user ID and password in a character string encoded by using Base64. Use the user ID and password of a user account that can perform operations on storage system resources. ▪ If a session is not to be generated: `Session` *`token`* When the session was generated: Specify a token that was obtained when the session was generated.

If a header other than the above is specified, the specified header is ignored.

Response headers

The response headers are response messages from the REST API server to REST API clients.

The following table describes the response headers returned by the REST API server.

Header	Description
Content-Type	Indicates the media type of the response data. Default: `application/json;charset=UTF-8`

WWW-Authenticate	Indicates that authentication is required when the HTTP status code `401` is returned.
	Default:
	▪ Authentication by using a user ID and password
	`Basic realm="Block storage"`
	▪ Authentication by using a session
	`Session realm="Block storage"`

HTTP status codes

The REST API uses the following standard HTTP status codes to indicate the processing results.

HTTP status codes	Description
200	OK The request was processed correctly.
202	Accepted The request for asynchronous processing was accepted.
400	Bad Request The request header, the query parameter, or the request body is invalid.
401	Unauthorized The Authorization header is not specified in the request header. Alternatively, authentication by using the information specified in the Authorization header fails.
403	Forbidden You do not have the permission required to perform the operation.
404	Not Found

HTTP status codes	Description
	The resource specified by the URL could not be found. Alternatively, there are no resources for the specified URL.
406	Not acceptable The media type specified in the Accept header is not supported.
415	Unsupported Media Type The specified media type is not supported.
500	Internal Server Error An internal error occurred on the server. Only the most critical error is returned. If an API function for operating multiple resources is run, and if partial information is obtained or if part of the operation is successful, the following attribute appears: `partialSuccessResponses` The displayed content differs depending on the type of HTTP method that was used: ▪ If the GET method was used: The obtained information (data object) appears. ▪ If the POST method was used: The execution results (`statusResource`) appear.
503	Service Unavailable Indicates that the service is temporarily unavailable because it is busy or under maintenance. Only the most critical error is returned. If an API function for operating multiple resources is run, and if partial information is obtained or if part of the operation is successful, the following attribute appears: `partialSuccessResponses` The displayed content differs depending on the type of HTTP method that was used: ▪ If the GET method was used: The obtained information (data object) appears. ▪ If the POST method was used: The execution results (`statusResource`) appear.
504	Gateway Timeout

HTTP status codes	Description
	No response was received from the device in the allotted time. Only the most critical error is returned. If an API function for operating multiple resources is run, and if partial information is obtained or if part of the operation is successful, the following attribute appears: `partialSuccessResponses` The displayed content differs depending on the type of HTTP method that was used: ▪ If the GET method was used: The obtained information (data object) appears. ▪ If the POST method was used: The execution results (`statusResource`) appear.

Request and response formats

JSON format is used for creating or adding resources (POST), for specifying the attribute value when changing resources (PATCH), and for getting the results of resource information (GET).

The supported character encoding is UTF-8.

Request format

- If you specified a null character for a string-type attribute, the value of the attribute is assumed to be null.
- If you specified a null character for an attribute whose type is not string, the attribute is assumed to be unspecified.

Response format

- If the API processing succeeds, a response is returned in JSON format.
- If the processing fails, depending on the contents of the error, a response in HTML format, instead of JSON format, might be returned.

  To resolve the error based on the HTTP status code in the program, check the value of Content-Type in the response header.

Query parameters

If the GET method is used to obtain an object, query parameters can be specified to filter the execution results based on specific conditions.

Query parameters can be specified at the end of the URL in the following format:

```
?parameter=value
```

To specify multiple parameters, concatenate them by using ampersands (&). Multiple parameters are specified in the following example:

```
?parameter=value&parameter=value...
```

For details on parameters that can be specified for queries, see the section explaining the specific API request.

Parameters are case-sensitive. If you specify a parameter other than those that can be specified for each API, the invalid parameter is ignored, and only the valid parameters are used to filter the execution results.

Data type

This section describes the data types that can be specified by using the REST API.

The following table shows the data types supported by the REST API and the corresponding JSON data types.

Data type	JSON type	Description
boolean	boolean	A type that represents `true` or `false`. The type is not case-sensitive. Example: `true`
int	number	A type that represents a 32-bit signed integer. Example: `100`
long	number	A type that represents a 64-bit signed integer. Example: `1048576`
string	string	A type that represents a character string. Example: `"DKR5D-J900SS"`
ISO8601string	string	A type that represents time in the ISO 8601 extended format (YYYY-MM-DDThh:mm:ssZ). The only time zone that you can specify is UTC. Example: `"2017-09-30T09:27:35Z"`

In addition to the preceding data types, the following JSON data types are also used:

Object type

The character string, in which the attribute and value are connected with a colon (:), is enclosed in { and }. If more than one attribute-value pair exists, the pairs are separated by commas.

Array type

The character string, in which multiple values are separated by commas, is enclosed in [and].

Output format

After an API request is issued, a response is returned based on the API processing method, the API processing type, and the execution result.

The following table shows the response output formats when the processing of the request is successful.

API processing method	API processing type	Status code of the execution result	Output format
Synchronous processing	GET (getting a single object)	200	See the description in the response message for each API function.
	GET (getting multiple objects)	200	Data object
	Other than the preceding	200	See the description in the response message for each API function.
Asynchronous processing	All	202	commandStatus object

If the processing of the request fails, an error object is returned as a response.

Data object

Data object is an object for returning the object list.

The following table shows the data object schema.

Attribute	Data type	Description
data	array	Object list

The following attributes are displayed along with the data object.

Attribute	Data type	Description
count	int	Number of objects
totalCount	int	Total number of data items in the storage system This attribute appears when one of the following API requests is executed: ▪ Getting volume information ▪ Getting information about the connections between volumes and servers ▪ Getting external volume information ▪ Getting information about external parity groups ▪ Getting information about external path groups ▪ Getting snapshot information ▪ Getting information about a specific snapshot group
hasNext	boolean	Whether there is information that has not yet been obtained ▪ `true`: Some information has not yet been obtained. ▪ `false`: All information has been obtained. If there is information that has not yet been obtained, filter the information to be obtained by specifying query parameters, or obtain the IDs (for example, volume IDs or snapshot IDs) that are larger than the largest ID in the information that is already obtained by splitting the IDs to be collected into groups and executing the API function multiple times.

Attribute	Data type	Description
		This attribute appears when one of the following API requests is executed: ▪ Getting volume information ▪ Getting information about the connections between volumes and servers ▪ Getting external volume information ▪ Getting information about external parity groups ▪ Getting information about external path groups ▪ Getting snapshot information ▪ Getting information about a specific snapshot group

The following shows an example of a data object:


```
{
  "data": [
    {
      "id": 100,
      "nickname": "JH-26216_DP",
      "poolId": 63,
      "poolName": "NASOS",
      "totalCapacity": 1024,
      "freeCapacity": 982,
      "numberOfConnectingServers": 2,
      "numberOfSnapshots": 2
    },
    {
      "id": 101,
      "nickname": "JH-26216_DP",
      "poolId": 63,
      "poolName": "NASOS",
      "totalCapacity": 1024,
      "freeCapacity": 1024,
      "numberOfConnectingServers": 2,
      "numberOfSnapshots": 2
    }
  ],
  "count": 2,
  "totalCount": 2,
  "hasNext": false
}
```

commandStatus object

A commandStatus object is the object of API status information that is returned when an API request for asynchronous processing is issued.

The following table explains the schema of a commandStatus object.

Attribute	Data type	Description
progress	string	Progress of the API function. The following values can be returned: ▪ `unexecuted`: The API function has not been run. ▪ `processing`: The API function is running. ▪ `completed`: The API function has completed.
status	string	Status of the execution results of the API function. The following values can be returned: ▪ `normal`: The API function finished successfully. ▪ `error`: The API function failed. This attribute appears only if the execution of the API function has completed.
affectedResources	string[]	URL used to access the resource targeted by the operation. If one API function request performs operations on multiple resources, the URLs of all those resources are returned. If an API function failed, only the URLs of the resources for which processing is confirmed as completed are returned. If the resource is deleted successfully, the URL of the deleted resource is returned. A `404` error occurs if this URL is accessed. This confirms that the resource was deleted successfully. This attribute appears if the execution of the API function finished successfully.
error	Error Object	Object that retains error information. This attribute appears if the execution of the API function failed.
operationDetails	object[]	Details about a resource targeted by the operation.

Attribute	Data type	Description
		This information appears when an operation is performed for the resource indicated by the resource type: ▪ `operationType` (string) The type of the operation performed for the resource: • `CREATE`: The resource has been created. • `UPDATE`: The resource has been updated. • `DELETE`: The resource has been deleted. ▪ `resourceType` (string) The resource type of a resource that was created, updated, or deleted: • `CommandStatus`: A resource related to status information about the API function that performs asynchronous processing • `Pool`: A resource related to pools • `Port`: A resource related to ports • `Server`: A resource related to servers • `Snapshot`: A resource related to snapshots • `Volume`: A resource related to volumes • `VolumeServerConnection`: A resource related to connections between volumes and servers • `ExternalVolume`: A resource related to external volumes ▪ `resourceId` (string) The resource ID of a resource that was created, updated, or deleted

Note:

- A maximum of 65,280 items of object information can be retained. If the number of items of API information exceeds 65,280, delete the object information by running the API that deletes the API status information in asynchronous processing.
- The object information is deleted even if the user logs out from the session.

Example of a commandStatus object when the API function starts running:

```
{
  "progress": "processing"
}
```

Example of a commandStatus object when the API function ends successfully:

```
{
  "progress": "completed",
  "status": "normal",
  "affectedResources": [
    "/ConfigurationManager/simple/v1/objects/volumes/100"
  ],
  "operationDetails": [
    {
      "operationType": "CREATE",
      "resourceType": "Volume",
      "resourceId": "100"
    }
  ]
}
```

Example of a commandStatus object when the API function failed:

```
{
  "progress": "completed",
  "status": "error",
  "errorResource": "/ConfigurationManager/simple/v1/objects/pools/63",
  "errorCode": {
    "SSB1": "2e11",
    "SSB2": "001f"
  },
  "errorMessage": "Snapshot or volume exists. Operation could not be
completed.",
  "operationDetails": [
    {
      "operationType": "DELETE",
      "resourceType": "Pool",
      "resourceId": "63"
    }
  ]
}
```

Error object

The following describes the error object that is returned together with an HTTP status code if an error occurs during the processing of a request.

The following table explains the schema of an error object.

Attribute	Data type	Description
errorSource	string	URL where the error occurs
messageId	string	Message ID
message	string	Content of the error message
cause	string	Cause of the error
solution	string	Solution to the error
errorCode	string	Error code from the storage system A value is returned only if an error occurs in the storage system. The error codes are as follows: ▪ SSB1 code ▪ SSB2 code Storage system error codes are required for maintenance of the storage system.
partialSuccessResponses	string	An error occurred, but the result is returned if the information is partially obtained or if the processing is partially successful.

The following is an example of an error object when an error occurs in a storage system:

```
{
  "errorSource": "/ConfigurationManager/simple/v1/objects/volumes/100/
actions/expand/invoke",
  "messageId": "KART70000-E",
  "message": "The specified volume is being used by another program
product, or format in progress. Operation could not be completed. Wait for
a while and try again. Confirm the volume is not in use by other program
product.",
  "errorCode": {
    "SSB1": "B96B",
    "SSB2": "AF2E"
  }
}
```

The following is an example of when an error occurs in a storage system but the processing is partially successful:

```
{
  "errorResponses": [
    {
      "errorSource": "/ConfigurationManager/simple/v1/objects/snapshots",
      "messageId": "KART70000-E",
      "message": "The specified volume does not exist. Check the parameter
and try again. If this problem occurs repeatedly, contact customer
support.",
      "errorCode": {
        "SSB1": "2E20",
        "SSB2": "0000"
      }
    }
  ],
  "partialSuccessResponses": [
    {
      "statusResource": "/ConfigurationManager/simple/v1/objects/command-
status/3"
    }
  ]
}
```

Chapter 12: Common operations in the REST API

Common operations in the REST API include obtaining and deleting status information for an asynchronous processing API request.

Getting status information about an API function that performs asynchronous processing

The following request obtains the status information about an API function that performs asynchronous processing. For the Authorization header of the request, specify the token of the session that was used when an API function that performs asynchronous processing was run.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/command-status/object-ID
```

Request message

Object ID

Specify the numerical value displayed at the end of the URL for `statusResource` that was returned when an API function for asynchronous processing was run.

The following is an example of a returned `statusResource`:

```
"statusResource": "/ConfigurationManager/simple/v1/objects/command-status/3"
```

Attribute	Type	Description
id	int	(Required) The ID used to obtain the execution results of the API function that performs asynchronous processing.

Query parameters

None.

Body

None.

Response message

Body

Example of a response message when the API function that creates a volume is run:

```
{
  "progress": "completed",
  "status": "normal",
  "affectedResources": [
    "/ConfigurationManager/simple/v1/objects/volumes/100"
  ],
  "operationDetails": [
    {
      "operationType": "CREATE",
      "resourceType": "Volume",
      "resourceId": "100"
    }
  ]
}
```

Example of a response message when the API function that deletes a pool fails:

```
{
  "progress": "completed",
  "status": "error",
  "errorResource": "/ConfigurationManager/simple/v1/objects/pools/
63",
  "errorCode": {
    "SSB1": "2e11",
    "SSB2": "001f"
  },
  "errorMessage": "Snapshot or volume exists. Operation could not be
completed.",
  "operationDetails": [
    {
      "operationType": "DELETE",
      "resourceType": "Pool",
      "resourceId": "63"
    }
  ]
}
```

Attribute	Type	Description
progress	string	Progress of the API function: ▪ `unexecuted`: The API function has not been run. ▪ `processing`: The API function is running. ▪ `completed`: The API function has finished.
status	string	Status of the execution results of the API function: ▪ `normal`: The API function finished successfully. ▪ `error`: The API function failed. This attribute appears only if the execution of the API function has completed.
affectedResources	string[]	List of URLs used to access the resource targeted by the operation. If one API function request performs operations on multiple resources, the URLs of all of those resources are returned. If an API function failed, only the URLs of the resources for which processing is confirmed as completed are returned. If the resource is deleted successfully, the URL of the deleted resource is returned. A 404 error occurs if this URL is accessed. This confirms that the resource was deleted successfully. This attribute appears if the execution of the API function finished successfully.
errorResource	string	URL where the error occurs. This attribute appears if the execution of the API function failed.
errorCode	object	Error code from the storage system. A value is returned only if an error occurs in the storage system. The error codes are as follows: ▪ SSB1 code ▪ SSB2 code

Attribute	Type	Description
		Storage system error codes are required for maintenance of the storage system. This attribute appears if the execution of the API function failed.
errorMessage	string	Content of the error message. This attribute appears if the execution of the API function failed.
operationDetails	object[]	Details about a resource targeted by the operation

Attribute	Type	Description
		This information appears when an operation is performed for the resource indicated by the resource type: ▪ `operationType` (string) The type of the operation performed for the resource • `CREATE`: The resource has been created. • `UPDATE`: The resource has been updated. • `DELETE`: The resource has been deleted. ▪ `resourceType` (string) The resource type of a resource that was created, updated, or deleted • `CommandStatus`: A resource related to status information about the API function that performs asynchronous processing • `Pool`: A resource related to pools • `Port`: A resource related to ports • `Server`: A resource related to servers • `Snapshot`: A resource related to snapshots • `Volume`: A resource related to volumes • `VolumeServerConnection`: A resource related to connections between volumes and servers • `ExternalVolume`: A resource related to external volumes ▪ `resourceId` (string) The resource ID of a resource that was created, updated, or deleted

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/command-status/3
```

Deleting status information about an API function that performs asynchronous processing

The following request deletes status information about an API function that performs asynchronous processing.

Execution permission

Storage Administrator (System Resource Management)

Request line

```
DELETE base-URL/simple/v1/objects/command-status/object-ID
```

Request message

Object ID

Specify the numerical value displayed at the end of the URL for `statusResource` that was returned when an API function for asynchronous processing was run.

The following is an example of a returned `statusResource`:

```
"statusResource": "/ConfigurationManager/simple/v1/objects/command-
status/3"
```

Attribute	Type	Description
id	int	(Required) The ID used to obtain the execution results of the API function that performs asynchronous processing.

Query parameters

None.

Body

None.

Response message

Body

Attribute	Type	Description
affectedResources	string[]	List of URLs for referencing the deleted status information of the API
operationDetails	object[]	Details about a resource for which the status information was deleted For details, see the description of the `operationDetails` attribute of the `commandStatus` object.

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE
https://192.0.2.100/ConfigurationManager/simple/v1/objects/command-status/3
```

Chapter 13: Performing common storage system operations using the REST API

You can perform common operations on a storage system, such as getting various information about the storage system and getting the operating status of pools and snapshots.

Getting information about a storage system

The following request obtains information about a storage system.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/storage
```

Request message

Object ID

None.

Query parameters

None.

Body

None.

Response message

Body

```
{
  "modelName": "VSP G700",
  "serial": "123456",
  "nickname": "VSP G700 #1",
  "numberOfTotalVolumes": 2,
  "numberOfFreeDrives": 0,
  "numberOfTotalServers": 2,
  "totalPhysicalCapacity": 4119873,
```

```
  "totalPoolCapacity": 2996994,
  "usedPoolCapacity": 38,
  "freePoolCapacity": 2996956,
  "savingEffects": {
    "efficiencyDataReduction": 105,
    "preCapacityDataReduction": 40,
    "postCapacityDataReduction": 38,
    "efficiencyFmdSaving": 0,
    "preCapacityFmdSaving": 0,
    "postCapacityFmdSaving": 0
  },
  "gumVersion": "88-02-00/23",
  "dkcMicroVersion": "88-02-00-60/22",
  "warningLedStatus": "OFF",
  "ipAddressIpv4Ctl1": "192.0.2.100",
  "ipAddressIpv4Ctl2": "192.0.2.101",
  "ipAddressIpv6Ctl1": "0:0:0:0:0:ffff:c000:264",
  "ipAddressIpv6Ctl2": "0:0:0:0:0:ffff:c000:265"
}
```


Attribute	Type	Description
modelName	string	Model name of the storage system
serial	string	Serial number of the storage system
nickname	string	Nickname of the storage system
numberOfTotalVolumes	int	Number of created volumes
numberOfFreeDrives	int	Number of free drives
numberOfTotalServers	int	Number of registered servers
totalPhysicalCapacity	long	Total capacity of the drives (MiB)
totalPoolCapacity	long	Total capacity of the pools (MiB)
usedPoolCapacity	long	Total used capacity of the pools (MiB)
freePoolCapacity	long	Total free capacity of the pools (MiB)

Attribute	Type	Description
savingEffects	object	Information about the capacity (amount and percentage) saved throughout the system by the capacity saving feature and accelerated compression function: ▪ `efficiencyDataReduction` (int) Percentage of the entire system's capacity saved by using the capacity saving feature (deduplication and compression) The value before reduction appears. This value is calculated under the assumption that the value after reduction is `100`. Example: If the value before reduction is `105` and the value after reduction is `100`, `"efficiencyDataReduction": 105` appears. You can convert it a ratio by dividing it by 100. Example: 105 / 100 = 1.05:1. The capacity after reduction is calculated based on a value that includes metadata and garbage data generated by the storage system, in addition to user data. For this reason, the value after reduction is sometimes greater than the value before reduction. If `-1` appears, this value is invalid. ▪ `preCapacityDataReduction` (long) Entire system's capacity before reduction by using the capacity saving feature (deduplication and compression) (MiB) ▪ `postCapacityDataReduction` (long) Entire system's capacity after reduction by using the capacity saving feature (deduplication and compression) (MiB) The capacity after reduction is calculated based on a value that includes metadata and garbage data generated by the storage system, in addition to user data. For this reason, the value after reduction is sometimes greater than the value before reduction.

Attribute	Type	Description
		▪ `efficiencyFmdSaving` (int) Percentage of the entire system's capacity saved by using the accelerated compression function. The value before reduction appears. This value is calculated under the assumption that the value after reduction is `100`. Example: If the value before reduction is `105` and the value after reduction is `100`, `"efficiencyFmdSaving": 105` appears. You can convert it a ratio by dividing it by 100. Example: 105 / 100 = 1.05:1. If `-1` appears, this value is invalid. ▪ `preCapacityFmdSaving` (long) Entire system's capacity before reduction by using the accelerated compression function (MiB) ▪ `postCapacityFmdSaving` (long) Entire system's capacity after reduction by using the accelerated compression function (MiB)
gumVersion	string	GUM version
dkcMicroVersion	string	Microcode version of the storage system
warningLedStatus	string	Status of failures in the storage system: ▪ OFF: The status is normal. ▪ ON: A failure has occurred. ▪ BLINK: An unidentified SIM exists.
ipAddressIpv4Ctl1	string	IP address for IPv4 on controller 1 (CTL1)
ipAddressIpv4Ctl2	string	IP address for IPv4 on controller 2 (CTL2)
ipAddressIpv6Ctl1	string	IP address for IPv6 on controller 1 (CTL1)
ipAddressIpv6Ctl2	string	IP address for IPv6 on controller 2 (CTL2)

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/storage
```

Getting the operating status of snapshots and pools

The following request gets status information about pools and snapshots, and enables you to check the operating status of storage resources.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/health-status
```

Request message

Object ID

None.

Query parameters

None.

Body

None.

Response message

Body

```
{
  "poolStatus":{
    "summary":"PartiallyBlocked",
    "abnormalItems":[
       {
        "status":"PartiallyBlocked",
        "poolIds":[
          0
        ]
       },
       {
        "status":"ExceededThreshold",
        "poolIds":[
          0,77,39,26
        ]
```

```
      }
    ]
  },
  "snapshotStatus":{
    "summary":"Normal"
  }
}
```

Attribute	Type	Description
poolStatus	object	Status of the pool: ▪ `summary` (string) • `Normal`: All pools are normal. • `ExceededThreshold`: There is at least one pool where usage rate exceeds the threshold value. • `PartiallyBlocked`: There is at least one pool where some of the volumes that make up the pool are blocked. • `Error`: At least one pool is in the error status because it is full. ▪ `abnormalItems` (object[]) Information about abnormal pools: • `status` (string) - `PartiallyBlocked`: Some of the volumes that make up the pool are blocked. - `ExceededThreshold`: The pools where usage rate exceeds the threshold value. - `Error`: The pools that are in the error status because they are full. • `poolIds` (int[]) Pool ID

Attribute	Type	Description
snapshotStatus	object	Status of the snapshot: ▪ `summary` (string) • `Normal`: All snapshots are normal. • `Error`: There is at least one snapshot where an error occurred. ▪ `abnormalItems` (object[]) Information about abnormal snapshots: • `status` (string) - `Error`: Snapshots where an error occurred. • `masterVolumeIds` (int[]) ID of the master volume when the snapshots were created.

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/health-status
```

Chapter 14: Preparing managed resources and attaching volumes using the REST API

REST API operations are used to register server information on a storage system, add path information between servers and ports, and create pools by specifying drives. This section also describes how to create new volumes and attach volumes to servers, how to expand the capacity of volumes that have a high usage rate, and how to expand the capacity of a pool by adding drives to the pool.

Drive management

You can use the REST API to get a list of information about drives, get information about specific drives, specify multiple drives as spare drives, or release spare drives.

Getting drive information

The following request gets information about drives. You can specify filter conditions.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/drives
```

Request message

Object ID

None.

Query parameters

Parameter	Type	Filter Condition
poolId	int	(Optional) ID of the pool to which the drives belong If `poolName` is specified, do not specify this parameter.

Parameter	Type	Filter Condition
poolName	string	(Optional) Name of the pool to which the drives belong Drive information will also be obtained if part of the specified value matches a pool name. If `poolId` is specified, do not specify this parameter. If both `poolId` and this parameter are specified, this parameter is ignored.
status	string	(Optional) Specify any of the following values as the status of the drives to be obtained: ▪ `Normal`: The drive with a status of normal. ▪ `Warning`: An error has occurred in a part of the drive. ▪ `Copying`: Copying is in progress. ▪ `CopyIncomplete`: Copying is incomplete. ▪ `Reserved`: The spare drive cannot be used. ▪ `Failed`: The drive is in the error status because of a failure. ▪ `Blocked`: The drive is in the error status because it is under maintenance.
useOfTheDrive	string	(Optional) Specify any of the following values as the purpose of the drives to be obtained: ▪ `data`: Data drive ▪ `spare`: Spare drive ▪ `free`: Unused drive

Body

None.

Response message

Body

The following is an example of output of obtained information for a drive that belongs to the pool with ID `63` (`63` is specified in the `poolId` query parameter):


```
{
  "data": [
    {
      "location": "0-0",
      "status": "Normal",
      "typeRpmCapacity": "SAS,10000,600",
      "driveType": "SAS",
      "driveRpm": "NUMBER_10000",
      "driveCapacity": 600,
      "typeCode": "DKR5D-J600SS",
      "poolIds": [
        63
      ],
      "useOfTheDrive": "data"
    },
    {
      "location": "0-1",
      "status": "Normal",
      "typeRpmCapacity": "SAS,10000,600",
      "driveType": "SAS",
      "driveRpm": "NUMBER_10000",
      "driveCapacity": 600,
      "typeCode": "DKR5D-J600SS",
      "poolIds": [
        63
      ],
      "useOfTheDrive": "data"
    },
    {
      "location": "0-2",
      "status": "Normal",
      "typeRpmCapacity": "SAS,10000,600",
      "driveType": "SAS",
      "driveRpm": "NUMBER_10000",
      "driveCapacity": 600,
      "typeCode": "DKR5D-J600SS",
      "poolIds": [
        63
      ],
      "useOfTheDrive": "data"
    },
    {
      "location": "0-3",
      "status": "Normal",
      "typeRpmCapacity": "SAS,10000,600",
```

```
      "driveType": "SAS",
      "driveRpm": "NUMBER_10000",
      "driveCapacity": 600,
      "typeCode": "DKR5D-J600SS",
      "poolIds": [
        63
      ],
      "useOfTheDrive": "data"
    }
  ],
  "count": 4
}
```


Attribute	Type	Description
location	string	Mounted location of the drive. Also commonly referred to as "name" or "ID".
status	string	Status of the drive: ▪ `Normal`: The drive with a status of normal. ▪ `Warning`: An error has occurred in a part of the drive. ▪ `Copying`: Copying is in progress. ▪ `CopyIncomplete`: Copying is incomplete. ▪ `Reserved`: The spare drive cannot be used. ▪ `Failed`: The drive is experiencing an error because of a failure. ▪ `Blocked`: The drive is experiencing an error because it is under maintenance. ▪ `Unknown`: The status is unknown.
typeRpmCapacity	string	The drive type, drive rotation speed, and drive capacity appear in a concatenated format, linked by commas.
driveType	string	Drive type: ▪ `SAS` ▪ `SSD` ▪ `FMD DC2`: A Hitachi flash-based SSD with compression capability.

Attribute	Type	Description
driveRpm	string	Drive rotation speed (rpm): ▪ `NUMBER_0` ▪ `NUMBER_7200` ▪ `NUMBER_10000` ▪ `NUMBER_15000` ▪ `High` ▪ `Middle` ▪ `Low` ▪ `Unknown`
driveCapacity	int	Drive capacity (GB)
typeCode	string	Drive type code
poolIds	int[]	List of pool IDs to which the drives belong
useOfTheDrive	string	Purpose of the drive: ▪ `data`: Data drive ▪ `spare`: Spare drive ▪ `free`: Unused drive

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/drives?poolId=63
```

Getting information about a specific drive

The following request gets information about a specific drive by using the specified drive location.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/drives/object-ID
```

Request message

Object ID

Specify a value for the `location` that was obtained by getting information about drives.

Attribute	Type	Description
location	string	(Required) Mounted location of the drive. Also commonly referred to as "name" or "ID".

Query parameters

None.

Body

None.

Response message

Body

The following is an example of the output when getting information about a specific drive (`location`:`0-0`):

```
{
  "location": "0-0",
  "status": "Normal",
  "typeRpmCapacity": "SAS,10000,600",
  "driveType": "SAS",
  "driveRpm": "NUMBER_10000",
  "driveCapacity": 600,
  "typeCode": "DKR5D-J600SS",
  "poolIds": [
    63
  ],
  "useOfTheDrive": "data"
}
```

For details on attributes to be obtained, see the description of the API function for getting drive information.

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/drives/0-0
```

Setting up spare drives

The following request sets up a spare drive with the specified number of drives.

Execution permission

Storage Administrator (Initial Configuration)

Request line

```
POST base-URL/simple/v1/objects/drives/actions/set-spare/invoke
```

Request message

Object ID

None.

Query parameters

None.

Body

The following is an example of setting up four data drives that have the drive type code `DKR5D-J600SS` as spare drives:

```
{
  "additionalSpareDrives": [
    {
      "driveTypeCode": "DKR5D-J600SS",
      "driveCount": 4
    }
  ]
}
```

Attribute	Type	Description
additionalSpareDrives	object[]	Drive information to be set for the spare drive: ▪ (Required) `driveTypeCode` (string) Specify a drive type code consisting of 12 characters.* Example: `DKR5D-J900SS` ▪ (Required) `driveCount` (int) Specify the number of data drives as an integer from `1` through `1440`.
* The spare drive setting can be specified for drives with a different drive type code, even if the drive conditions (drive type, drive rotation speed, and drive capacity) are the same. The system specifies the spare drive setting for drives that match the drive conditions, regardless of the drive type code you specify. You can check information about the drives specified as spare drives by running the API request for getting drive information.		

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the set up the spare drives request

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97) .

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/drives/actions/set-spare/invoke
```

Releasing the spare drive setting

The following request releases the spare drive setting with the specified number of drives. The drive can then be used as a normal drive.

Execution permission

Storage Administrator (Initial Configuration)

Request line

```
POST base-URL/simple/v1/objects/drives/actions/release-spare/invoke
```

Request message

Object ID

None.

Query parameters

None.

Body

The following is an example of releasing four spare drives that have the drive type code `DKR5D-J600SS`:

```
{
  "releasedSpareDrives": [
    {
      "driveTypeCode": "DKR5D-J600SS",
      "driveCount": 4
    }
  ]
}
```

Attribute	Type	Description
releasedSpareDrives	object[]	Drive information for which the spare drive setting is to be released: ▪ (Required) `driveTypeCode` (string) Specify a drive type code consisting of 12 characters.* Example: `DKR5D-J900SS` ▪ (Required) `driveCount` (int) Specify the number of data drives as an integer from `1` through `1440`.

Attribute	Type	Description
* The spare drive setting can be released for drives with a different drive type code, even if the drive conditions (drive type, drive rotation speed, and drive capacity) are the same. The system releases the spare drive setting for drives that match the drive conditions, regardless of the drive type code you specify. You can check information about the drives for which the spare drive setting is released by running the API request for getting drive information.		

Response message

Body

Attribute	Type	Description
statusResource	string	URL to be used to obtain the execution results of drives for which the spare drive setting is released

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97) .

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/drives/actions/release-spare/invoke
```

Pool management

You can use the REST API to get information about pools, get detailed information about specific pools, create pools by specifying drives, or increase the capacity of a pool by adding drives to the pool.

Getting pool information

The following request gets information about the pool. You can specify filter conditions.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/pools
```

Request message

Object ID

None.

Query parameters

Parameter	Type	Filter Condition
name	string	(Optional) Pool name Pool information will also be obtained if part of the specified value matches a pool name.
status	string	(Optional) Specify the following values to obtain information about the pools with a specific status: ▪ `Normal`: The pools with a status of normal. ▪ `ExceededThreshold`: The pools where the usage rate exceeds the threshold value. ▪ `Error`: The pools that are in the error status because they are full.
configStatus	string	(Optional) Specify `PartiallyBlocked` (some of the volumes that make up the pools are blocked) as the status of the volumes that make up the pools to be obtained.

Body

None.

Response message

Body

```
{
  "data": [
    {
      "id": 63,
```

```
      "name": "NASOS",
      "status": "Normal",
      "totalCapacity": 2996994,
      "usedCapacity": 38,
      "freeCapacity": 2996956,
      "capacityManage": {
        "usedCapacityRate": 0,
        "thresholdWarning": 70,
        "thresholdDepletion": 80
      },
      "savingEffects": {
        "efficiencyDataReduction": 105,
        "efficiencyFmdSaving": 0,
        "preCapacityFmdSaving": 0,
        "postCapacityFmdSaving": 0
      },
      "configStatus": [],
      "numberOfVolumes": 2,
      "numberOfTiers": 2,
      "numberOfDriveTypes": 2,
      "tiers": [
        {
          "driveType": "Flash Drive",
          "driveRpm": "NUMBER_0",
          "totalCapacity": 1352148,
          "usedCapacity": 28
        },
        {
          "driveType": "SAS",
          "driveRpm": "NUMBER_10000",
          "totalCapacity": 1644846,
          "usedCapacity": 10
        }
      ]
    }
  ],
  "count": 1
}
```


Attribute	Type	Description
id	int	Pool ID
name	string	Pool name

Attribute	Type	Description
status	string	State of the pool: ▪ `Normal`: The pool is in normal status. ▪ `ExceededThreshold`: The pool usage rate exceeds the threshold value. ▪ `Error`: The pools that are in the error status because they are full.
totalCapacity	long	Total capacity of the pool (MiB)
usedCapacity	long	Used capacity of the pool (MiB)
freeCapacity	long	Free capacity of the pool (MiB)
capacityManage	object	Configuration information about the pool capacity: ▪ `usedCapacityRate` (int) Pool usage range (%) ▪ `thresholdWarning` (int) Warning threshold value of the pool (%) ▪ `thresholdDepletion` (int) Depletion threshold value of the pool (%)

Attribute	Type	Description
savingEffects	object	Information such as the rate of saving capacity: ▪ `efficiencyDataReduction` (int) Percentage of the pool's capacity saved by using the capacity saving feature (deduplication and compression) The value before reduction appears. This value is calculated under the assumption that the value after reduction is `100`. Example: If the value before reduction is `105` and the value after reduction is `100`, `"efficiencyDataReduction": 105` appears. The capacity after reduction is calculated based on a value that includes metadata and garbage data generated by the storage system, in addition to user data. Therefore, the value after reduction is sometimes greater than the value before reduction. If `-1` appears, this value is invalid. ▪ `efficiencyFmdSaving` (int) Percentage of the pool's capacity saved by using the accelerated compression function The value before reduction appears. This value is calculated under the assumption that the value after reduction is `100`. Example: If the value before reduction is `105` and the value after reduction is `100`, `"efficiencyFmdSaving": 105` appears. If `-1` appears, this value is invalid. ▪ `preCapacityFmdSaving` (long) Pool's capacity before reduction by using the accelerated compression function ▪ `postCapacityFmdSaving` (long) Pool's capacity after reduction by using the accelerated compression function

Attribute	Type	Description
configStatus	string[]	Status of the volumes that make up the pool ▪ `PartiallyBlocked`: Some of the volumes that make up the pool are blocked
numberOfVolumes	int	Number of volumes associated with the pool
numberOfTiers	int	Number of tiers that make up the pool
numberOfDriveTypes	int	Number of drive types
tiers	object[]	Attributes about tiers for each tier: ▪ `driveType` (string) One of the following drive types appears: • `Flash Drive` • `SAS` • `SATA` • `External Storage` • `Mixed` • `Unknown` ▪ `driveRpm` (string) Drive rotation speed per minute ▪ `totalCapacity` (long) Total capacity of the tier (MiB) ▪ `usedCapacity` (long) Used capacity of the tier (MiB)

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/pools
```

Getting information about a specific pool

The following request gets information about a specific pool by using the specified pool ID.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/pools/object-ID
```

Request message

Object ID

Specify a value for the `id` that was obtained by getting information about pools.

Attribute	Type	Description
id	int	(Required) Pool ID

Query parameters

None.

Body

None.

Response message

Body

The following is an example of output when getting information about a pool with ID `63`:

```
{
  "id": 63,
  "name": "NASOS",
  "status": "Normal",
  "totalCapacity": 2996994,
  "usedCapacity": 38,
  "freeCapacity": 2996956,
  "capacityManage": {
    "usedCapacityRate": 0,
    "thresholdWarning": 70,
    "thresholdDepletion": 80
  },
  "savingEffects": {
    "efficiencyDataReduction": 105,
    "efficiencyFmdSaving": 0,
```

```
    "preCapacityFmdSaving": 0,
    "postCapacityFmdSaving": 0
  },
  "configStatus": [],
  "numberOfVolumes": 2,
  "numberOfTiers": 2,
  "numberOfDriveTypes": 2,
  "tiers": [
    {
      "driveType": "Flash Drive",
      "driveRpm": "NUMBER_0",
      "totalCapacity": 1352148,
      "freeCapacity": 1352110
    },
    {
      "driveType": "SAS",
      "driveRpm": "NUMBER_10000",
      "totalCapacity": 1644846,
      "freeCapacity": 1644846
    }
  ],
  "drives": [
    {
      "driveType": "SAS",
      "driveRpm": "NUMBER_10000",
      "driveCapacity": 600,
      "totalCapacity": 2400,
      "numberOfDrives": 4,
      "locations": [
        "0-0",
        "0-1",
        "0-2",
        "0-3"
      ],
      "raidLevel": "RAID5"
    },
    {
      "driveType": "SSD",
      "driveRpm": "NUMBER_0",
      "driveCapacity": 480,
      "totalCapacity": 1920,
      "numberOfDrives": 4,
      "locations": [
        "1-0",
        "1-1",
        "1-2",
        "1-3"
      ],
      "raidLevel": "RAID5"
    },
```

```
  ]
}
```

This operation obtains the values of attributes obtained as part of the pool information, as well as the values of the following attributes.

Attribute	Type	Description
drives	object[]	Drive information: ▪ `driveType` (string) Drive type: • `SAS` • `SSD` • `FMD DC2`: A Hitachi flash-based SSD with compression capability. ▪ `driveRpm` (string) Drive rotation speed (rpm): • `NUMBER_0` • `NUMBER_7200` • `NUMBER_10000` • `NUMBER_15000` • `High` • `Middle` • `Low` • `Unknown` ▪ `driveCapacity` (int) Drive capacity (GB) ▪ `totalCapacity` (long) Total capacity of the drive (MiB) ▪ `numberOfDrives` (int) Number of drives that belong to the pool

Attribute	Type	Description
		▪ `locations` (string[]) Mounted locations of the drive. Also commonly referred to as "name" or "ID". ▪ `raidLevel` (string) RAID level: • `RAID1` • `RAID5` • `RAID6` • `Mixed`

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/pools/63
```

Creating a pool

The following request creates a pool by using the specified pool name and drive information.

For storage systems with a Dynamic Tiering license, if you specify drives that have different drive conditions (drive type, drive rotation speed, or drive capacity) at the same time, tiered pools, where the tiers are arranged by performance, are created. In a tiered pool, data is automatically stored in optimum tiers.

Note: If the differential data of a snapshot is stored in a pool and the usage rate of the pool exceeds the depletion threshold value, the created snapshot might become unusable. To change the depletion threshold value, change the pool settings after you create the pool.

Execution permission

Storage Administrator (Provisioning)

Request line

```
POST base-URL/simple/v1/objects/pools
```

Request message

Object ID

None.

Query parameters

None.

Body

The following is an example of creating a pool, with the pool name and drive information specified as follows:

- Pool name: `NASOS`
- Drive type code: `DKR5D-J600SS`
- Number of data drives: `4`
- RAID level: `RAID5`

```
{
  "name": "NASOS",
  "drives": [
    {
      "driveTypeCode": "DKR5D-J600SS",
      "dataDriveCount": 4,
      "raidLevel": "RAID5"
    }
  ]
}
```

Attribute	Type	Description
name	string	(Required) Specify a pool name consisting of up to 32 characters. You can use alphanumeric characters (`0` through `9`, `A` through `Z`, `a` through `z`), space characters, and the following symbols: `- . / : @ \ _` This attribute is case-sensitive. You can use a space character between characters, but cannot use it at the beginning or end of the pool name. You cannot use a hyphen (-) at the beginning of the pool name.

Attribute	Type	Description
drives	object[]	Specify the drive information to be used for the pool: ▪ (Required) `driveTypeCode` (string) Specify a drive type code consisting of 12 characters.* Example: `DKR5D-J900SS` ▪ (Required) `dataDriveCount` (int) Specify the number of data drives as an integer from `1` through `1440`. If `RAID1` or `RAID5` is specified as the RAID level, specify `4` or a greater value for the number of data drives. If `RAID6` is specified as the RAID level, specify `8` or a greater value for the number of data drives. ▪ (Required) `raidLevel` (string) RAID level Specify one of the following RAID levels: • `RAID1` • `RAID5` • `RAID6`
* The storage system can contain drives that have the same drive conditions (drive type, drive rotation speed, and drive capacity), but different drive type codes. In this case, regardless of the drive type code you specified, drives that match the drive conditions are used to create a pool. You can check information about the drives belonging to the newly created pool by specifying the corresponding pool ID as a query parameter for the API request to getting drive information and then executing the request.		

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the create pool request

> **Note:**
>
> Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/pools
```

Changing pool settings

The following request changes pool settings (such as the pool name or threshold values) by using the specified pool ID.

If the differential data of a snapshot is stored in a pool and the usage rate of the pool exceeds the depletion threshold value, the created snapshot might become unusable. Change the depletion threshold value as necessary.

Execution permission

Storage Administrator (Provisioning)

Request line

```
PATCH base-URL/simple/v1/objects/pools/object-ID
```

Request message

Object ID

Specify a value for the `id` that was obtained by getting the pool information.

Attribute	Type	Description
id	int	(Required) Pool ID

Query parameters

None.

Body

Only the specified attributes are changed.

The following is an example of changing the warning threshold value to `80` and the depletion threshold value to `90` for a pool with ID `63`:

```
{
  "thresholdWarning": 80,
  "thresholdDepletion": 90
}
```

Attribute	Type	Description
name	string	(Optional) Specify a pool name consisting of up to 32 characters. You can use alphanumeric characters (`0` through `9`, `A` through `Z`, `a` through `z`), space characters, and the following symbols: Hyphens (-), periods (`.`), forward slashes (`/`), colons (`:`), at marks (`@`), back slashes, (`\`), underscores (_) This attribute is case-sensitive. You can use a space character between characters, but cannot use it at the beginning or end of the pool name. You cannot use a hyphen (-) at the beginning of the name.
thresholdWarning	int	(Optional) Warning threshold value of the pool (%) Specify the value as an integer from `1` through `100`.
thresholdDepletion	int	(Optional) Depletion threshold value of the pool (%) Specify the value as an integer from `1` through `100`.

Response message

Body

Attribute	Type	Description
affectedResources	string[]	List of URLs for referencing information about a changed pool
operationDetails	object[]	Details about the changed pool

Attribute	Type	Description
		For details, see the description of the `operationDetails` attribute of the `CommandStatus` object.

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X PATCH --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/pools/63
```

Deleting a pool

The following request deletes a pool by using the specified pool ID.

Execution permission

Storage Administrator (Provisioning)

Request line

```
DELETE base-URL/simple/v1/objects/pools/object-ID
```

Request message

Object ID

Specify the `id` value obtained by getting information about the pool.

Attribute	Type	Description
id	int	(Required) Pool ID

Query parameters

None.

Body

None.

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the delete pool request

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97) .

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE
https://192.0.2.100/ConfigurationManager/simple/v1/objects/pools/63
```

Expanding the capacity of a pool

The following request expands the capacity of the pool by adding a drive to the pool.

For storage systems with a Dynamic Tiering license, if you include drives that have different drive conditions (drive type, drive rotation speed, and drive capacity), the expanded pool will be a tiered pool where the tiers are arranged by performance. In a tiered pool, data is automatically stored in optimum tiers.

Execution permission

Storage Administrator (Provisioning)

Request line

```
POST base-URL/simple/v1/objects/pools/object-ID/actions/expand/invoke
```

Request message

Object ID

Specify the `id` value obtained by getting information about the pool.

Attribute	Type	Description
id	int	(Required) Pool ID

Query parameters

None.

Body

The following is an example of adding four data drives that have the drive type code `SLB5F-M480SS` to a pool with ID `63`:

```
{
  "additionalDrives": [
    {
      "driveTypeCode": "SLB5F-M480SS",
      "dataDriveCount": 4,
      "raidLevel": "RAID5"
    }
  ]
}
```

Attribute	Type	Description
additionalDrives	object[]	Information of drives to be added to the pool: ▪ (Required) `driveTypeCode` (string) Specify a drive type code consisting of 12 characters.* Example: `DKR5D-J900SS` ▪ (Required) `dataDriveCount` (int) Specify the number of data drives as an integer from `1` through `1440`. If `RAID1` or `RAID5` is specified as the RAID level, specify `4` or a greater value for the number of data drives. If `RAID6` is specified as the RAID level, specify `8` or a greater value for the number of data drives. ▪ (Required) `raidLevel` (string) Specify one of the following RAID levels: • `RAID1` • `RAID5` • `RAID6`

*

- The storage system can contain drives that have the same drive conditions (drive type, drive rotation speed, and drive capacity), but different drive type codes. In this case, regardless of the drive type code you specified, drives that match the drive conditions are added to expand the pool.

  You can check information about the drives added to a pool by specifying the corresponding pool ID as a query parameter for the API request of getting drive information and then executing the request.
- If a snapshot is created in an expanded pool, drives that have different drive conditions cannot be specified.

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the expand the pool capacity request

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97) .

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/pools/63/actions/expand/invoke
```

Port management

You can use the REST API to get a list of information about ports, get information about a specific port, or change the port settings.

Getting port information

The following request obtains information about ports.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/ports
```

Request message

Object ID

None.

Query parameters

Parameter	Type	Filter Condition
protocol	string	(Optional) Specify one of the following values as the protocol of the port: ▪ `FC` ▪ `iSCSI`

Body

None.

Response message

Body

The following in an example of the output when information is obtained about all ports.


```
{
  "data": [
    {
      "id": "CL1-A",
      "protocol": "FC",
      "portWwn": "50060e80124e3b00",
      "portSpeed": "NUMBER_0",
      "portSecurity": true,
      "fcInformation": {
        "alPa": "EF",
        "fabricSwitchSetting": true,
        "connectionType": "Point_To_Point",
        "sfpDataTransferRate": "NUMBER_16"
      }
    },
    {
      "id": "CL1-B",
      "protocol": "iSCSI",
      "portSpeed": "NUMBER_10",
      "portSecurity": true,
      "iscsiInformation": {
        "vlanUse": false,
        "ipMode": "ipv4",
        "ipv4Information": {
          "address": "192.168.116.19",
          "subnetMask": "255.255.0.0",
```

```
          "defaultGateway": "0.0.0.0"
      },
        "ipv6Information": {
          "linklocal": "Auto",
          "linklocalAddress": "fe80::",
          "global": "Auto",
          "globalAddress": "::",
          "defaultGateway": "::"
      },
      "tcpPort": 3260,
      "selectiveAck": true,
      "delayedAck": true,
      "windowSize": "NUMBER_64K",
      "mtuSize": "NUMBER_1500",
      "keepAliveTimer": 60,
      "isnsServerMode": false,
      "isnsServerIpAddress": "0.0.0.0",
      "isnsServerPort": 3205
    }
  ],
  "count": 2
}
```


The following output is an example of information obtained about ports whose protocol is iSCSI (when `iSCSI` is specified in the `protocol` query parameter):


```
{
  "data": [
    {
      "id": "CL1-B",
      "protocol": "iSCSI",
      "portSpeed": "NUMBER_10",
      "portSecurity": true,
      "iscsiInformation": {
        "vlanUse": false,
        "ipMode": "ipv4",
        "ipv4Information": {
          "address": "192.168.116.19",
          "subnetMask": "255.255.0.0",
          "defaultGateway": "0.0.0.0"
      },
        "ipv6Information": {
          "linklocal": "Auto",
          "linklocalAddress": "fe80::",
          "global": "Auto",
          "globalAddress": "::",
          "defaultGateway": "::"
      },
      "tcpPort": 3260,
      "selectiveAck": true,
```

```
      "delayedAck": true,
      "windowSize": "NUMBER_64K",
      "mtuSize": "NUMBER_1500",
      "keepAliveTimer": 60,
      "isnsServerMode": false,
      "isnsServerIpAddress": "0.0.0.0",
      "isnsServerPort": 3205,
      "virtualPortEnabled": false
    }
  ],
  "count": 1
}
```


Attribute	Type	Description
id	string	Port ID
protocol	string	Protocol that can be any of the following values: ▪ `FC` ▪ `iSCSI`
portWwn	string	WWN of the port The WWN in use when the protocol is Fibre Channel.
portSpeed	string	Data transfer speed of the port: ▪ `NUMBER_0`: Auto ▪ `NUMBER_1`: 1 Gbps ▪ `NUMBER_2`: 2 Gbps ▪ `NUMBER_4`: 4 Gbps ▪ `NUMBER_8`: 8 Gbps ▪ `NUMBER_10`: 10 Gbps ▪ `NUMBER_16`: 16 Gbps ▪ `NUMBER_32`: 32 Gbps
portSecurity	boolean	Whether the port security setting is enabled: ▪ `true`: Enabled ▪ `false`: Disabled

Attribute	Type	Description
fcInformation	object	FC information when the protocol is Fibre Channel: ▪ `alPa` (string) Address of the port used for Arbitrated Loop Physical Address (AL_PA). ▪ `fabricSwitchSetting` (boolean) Whether the fabric switch setting is enabled: • `true`: Enabled • `false`: Disabled ▪ `connectionType` (string) Topology setting: • `Point_To_Point` • `FC_AL` ▪ `sfpDataTransferRate` (string) Transfer rate: • `NUMBER_16`: 16 Gbps • `NUMBER_32`: 32 Gbps
iscsiInformation	object	iSCSI information when the protocol is iSCSI: ▪ `vlanUse` (boolean) Whether the VLANs are enabled: • `true`: Enabled • `false`: Disabled ▪ `vlanId` (int) VLAN ID Appears only if VLANs are enabled. ▪ `ipMode` (string) IP address format: • `ipv4` • `ipv4v6`

Attribute	Type	Description
		▪ `ipv4Information` (object) IPv4 information: • `address` (string) IP address • `subnetMask` (string) Subnet mask • `defaultGateway` (string) Default gateway address ▪ `ipv6Information` (object) IPv6 information: • `linklocal` (string) Method for setting the link local address: - `Auto` - `Manual` • `linklocalAddress` (string) Link local address • `global` (string) Method for setting the global address: - `Auto` - `Manual` • `globalAddress` (string) Global address • `defaultGateway` (string) Gateway address ▪ `tcpPort` (int) TCP port number for iSCSI communications. ▪ `selectiveAck` (boolean) Whether selective ACK is enabled: • `true`: Enabled • `false`: Disabled

Attribute	Type	Description
		▪ `delayedAck` (boolean) Whether delayed ACK is enabled: • `true`: Enabled • `false`: Disabled ▪ `windowSize` (string) Window size ▪ `mtuSize` (int) MTU size ▪ `keepAliveTimer` (int) Value (in seconds) of the Keep Alive timer for iSCSI communications. ▪ `isnsServerMode` (boolean) Whether iSNS server mode is enabled: • `true`: Enabled • `false`: Disabled ▪ `isnsServerIpAddress` (string) IP address of the iSNS server (IPv4 or IPv6) The IP address in use when the iSNS server mode setting is enabled. If virtual ports are enabled, this attribute is hidden. ▪ `isnsServerPort` (int) TCP port number of the iSNS server The TCP port number in use when the iSNS server mode setting is enabled. If virtual ports are enabled, this attribute is hidden. ▪ `virtualPortEnabled` (boolean) Whether the virtual ports are enabled: • `true`: Enabled • `false`: Disabled

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/ports
```

Getting information about a specific port

This request gets information about the port for a specified port ID.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/ports/object-ID
```

Request message

Object ID

Specify the `id` value obtained by getting information about the port.

Attribute	Type	Description
id	int	(Required) Port ID

Query parameters

None.

Body

None.

Response message

Body

The following output is an example of information obtained about a Fibre Channel port (port ID: `CL1-A`):

```
{
  "id": "CL1-A",
  "protocol": "FC",
  "portWwn": "50060e80124e3b00",
  "portSpeed": "NUMBER_0",
  "portSecurity": true,
```

```
  "fcInformation": {
    "alPa": "EF",
    "fabricSwitchSetting": true,
    "connectionType": "Point_To_Point",
    "sfpDataTransferRate": "NUMBER_16"
  }
}
```


The following output is an example of information obtained about an iSCSI port (port ID: `CL1-B`):


```
{
  "id": "CL1-B",
  "protocol": "iSCSI",
  "portSpeed": "NUMBER_10",
  "portSecurity": true,
  "iscsiInformation": {
    "vlanUse": false,
    "ipMode": "ipv4",
    "ipv4Information": {
      "address": "192.168.116.19",
      "subnetMask": "255.255.0.0",
      "defaultGateway": "0.0.0.0"
  },
  "ipv6Information": {
    "linklocal": "Auto",
    "linklocalAddress": "fe80::",
    "global": "Auto",
    "globalAddress": "::",
    "defaultGateway": "::"
  },
  "tcpPort": 3260,
  "selectiveAck": true,
  "delayedAck": true,
  "windowSize": "NUMBER_64K",
  "mtuSize": "NUMBER_1500",
  "keepAliveTimer": 60,
  "isnsServerMode": false,
  "isnsServerIpAddress": "0.0.0.0",
  "isnsServerPort": 3205,
  "virtualPortEnabled": false
}
```


For details on the attributes that are obtained, see the description of the API function for getting port information.

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/ports/CL1-A
```

Changing port settings

The following request changes the settings of the port for the specified port ID.

Execution permission

Storage Administrator (Provisioning)

Request line

```
PATCH base-URL/simple/v1/objects/ports/object-ID
```

Request message

Object ID

Specify the `id` value obtained by getting information about the port.

Attribute	Type	Description
id	int	(Required) Port ID

Query parameters

None.

Body

Only the specified attributes are changed.

The following is an example of changing the settings of a Fibre Channel port (port ID: `CL1-A`):

- Change the fabric switch setting to `true`.
- Change the connection type to `Point_To_Point`.

```
{
  "fcInformation": {
    "fabricSwitchSetting": true,
    "connectionType": "Point_To_Point"
  }
}
```

The following is an example of changing the settings of an iSCSI port (port ID: `CL1-B`):

- Change the IPv6 link local address setting to `Auto`.
- Change the IPv6 global address setting to `Auto`.
- Change the selective ACK setting to `true`.
- Change the value for the MTU size to `NUMBER_4500`.

```
{
  "iscsiInformation": {
    "ipv6Information": {
      "linklocal": "Auto",
      "global": "Auto"
    },
    "selectiveAck": true,
    "mtuSize": "NUMBER_4500"
  }
}
```

Attribute	Type	Description
portSpeed	string	(Optional) Data transfer speed of the port You can specify the following values: ▪ `NUMBER_0`: Auto ▪ `NUMBER_1`: 1 Gbps ▪ `NUMBER_2`: 2 Gbps ▪ `NUMBER_4`: 4 Gbps ▪ `NUMBER_8`: 8 Gbps ▪ `NUMBER_10`: 10 Gbps ▪ `NUMBER_16`: 16 Gbps ▪ `NUMBER_32`: 32 Gbps For iSCSI (optical) ports, the data transfer speed is specified and fixed as `NUMBER_10` (10 Gbps). If a value other than `NUMBER_10` is specified, that value will be ignored.
portSecurity	boolean	(Optional) Specify whether to enable the port security setting: ▪ `true`: Enable the port security setting ▪ `false`: Disable the port security setting

Attribute	Type	Description
fcInformation	object	Change the FC settings. You change these settings only if the protocol is Fibre Channel: ▪ (Optional) `alPa` (string) Specify the port address (AL_PA) by using a two-digit hexadecimal number (`01` through `EF`). ▪ (Optional) `fabricSwitchSetting` (boolean) Specify whether to enable the fabric switch setting: • `true`: Enable the fabric switch setting • `false`: Disable the fabric switch setting ▪ (Optional) `connectionType` (string) Specify the connection type setting: • `Point_To_Point` • `FC_AL`
iscsiInformation	object	Change the iSCSI settings. You can change these settings only if the protocol is iSCSI: ▪ (Optional) `vlanUse` (boolean) Enable or disable the VLANs: • `true`: Enable the VLANs • `false`: Disable the VLANs ▪ (Optional) `addVlanId` (int) Specify the VLAN ID to be added, as an integer from `1` through `4094`. ▪ (Optional) `deleteVlanId` (int) Specify the VLAN ID to be deleted, as an integer from `1` through `4094`. ▪ (Optional) `ipMode` (string) Specify the format of the IP address: • `ipv4`: Use IPv4 • `ipv4v6`: Use IPv4 and IPv6

Attribute	Type	Description
		▪ `ipv4Information` (object) Set IPv4 information: • (Optional) `address` (string)[1] Specify an IP address in IPv4 format by using 7 through 15 characters. • (Optional) `subnetMask` (string) Specify the subnet mask by using 7 through 15 characters. • (Optional) `defaultGateway` (string) Specify the address of the default gateway by using 7 through 15 characters. ▪ `ipv6Information` (object) Set IPv6 information: • (Optional) `linklocal` (string)[2] Specify `Auto` (automatic), or specify a link local address by using 2 through 45 characters. • (Optional) `global` (string)[2] Specify `Auto` (automatic), or specify a global address by using 2 through 45 characters. • (Optional) `defaultGateway` (string)[2] Specify the address of the default gateway by using 2 through 45 characters. ▪ (Optional) `tcpPort` (int) Specify the TCP port number for iSCSI communication as an integer from `1` through `65535`. ▪ (Optional) `selectiveAck` (boolean) Enable or disable selective ACK: • `true`: Enable selective ACK • `false`: Disable selective ACK

Attribute	Type	Description
		▪ (Optional) `delayedAck` (boolean) Enable or disable delayed ACK: • `true`: Enable selective ACK • `false`: Disable selective ACK ▪ (Optional) `windowSize` (string) Specify the window size. You can specify `NUMBER_64K`, `NUMBER_128K`, `NUMBER_256K`, `NUMBER_512K`, or `NUMBER_1024K`. ▪ (Optional) `mtuSize` (int) Specify the MTU size. You can specify `NUMBER_1500`, `NUMBER_4500`, or `NUMBER_9000`. ▪ (Optional) `keepAliveTimer` (int) Specify the value (in seconds) of the Keep Alive timer for iSCSI communications as an integer from `30` through `64800`. ▪ (Optional) `isnsServerMode` (boolean) Enable or disable the iSNS server mode: • `true`: Enable the iSNS server mode • `false`: Disable the iSNS server mode ▪ (Optional) `isnsServerIpAddress` (string) Specify the IP address of the iSNS server in IPv4 or IPv6 format by using 2 through 45 characters. ▪ (Optional) `isnsServerPort` (int) Specify the TCP port number of the iSNS server as an integer from `1` through `65535`. ▪ (Optional) `virtualPortEnabled` (boolean) Enable or disable the virtual ports: • `true`: Enable the virtual ports • `false`: Disable the virtual ports

Attribute	Type	Description
1 If you specify this attribute, you cannot specify the following addresses: - Network address (Example: `192.168.10.0`) - Broadcast address (Example: `255.255.255.255`) - Loopback address (Example: `127.0.0.1`) 2 If you specify this attribute, you cannot specify the following addresses: - No address specified (Example: `::`) - Multicast address (Example: `ff00:1024:1215::01`) - Loopback address (Example: `::1`)		

Response message

Body

Attribute	Type	Description
affectedResources	string[]	List of URLs for referencing information about the changed port
operationDetails	object[]	Details about the changed port For details, see the description of the `operationDetails` attribute of the `commandStatus` object.

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X PATCH --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/ ports/CL1-A
```

Server management

You can use the REST API to get information about servers, get detailed information about specific servers, register server information in a storage system, add information about paths between servers and ports, or add the WWNs (or iSCSI names) of HBAs.

Getting server information

The following request obtains information about servers. You can specify filter conditions.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/servers
```

Request message

Object ID

None.

Query parameters

Parameter	Type	Filter Condition
nickname	string	(Optional) Server nickname Information is obtained about the server that has the nickname is a perfect match of the specified value.
hbaWwn	string	(Optional) WWN of the HBA Information is obtained about the server that has the WWN of the HBA that is a perfect match to the specified value.
iscsiName	string	(Optional) iSCSI name Information is obtained about the server that has the iSCSI name that is a perfect match to the specified value.

Body

None.

Response message

Body

The following is an example of output when getting information about the servers whose nicknames are hostA and hostB respectively (if `hostA` and `hostB` are specified for the query parameter `nickname` of the corresponding servers).

```
{
  "data": [
```

```
    {
      "id": 10,
      "nickname": "hostA",
      "protocol": "FC",
      "osType": "Linux",
      "totalCapacity": 1024,
      "usedCapacity": 42,
      "numberOfPaths": 2,
      "isInconsistent": false,
      "modificationInProgress": false,
      "compatibility": "DKCMAIN_8802010000",
      "isReserved": false,
      "hasUnalignedOsTypes": false
    },
    {
      "id": 11,
      "nickname": "hostB",
      "protocol": "iSCSI",
      "osType": "Linux",
      "totalCapacity": 1024,
      "usedCapacity": 42,
      "numberOfPaths": 2,
      "isInconsistent": false,
      "modificationInProgress": false,
      "compatibility": "DKCMAIN_8802010000",
      "isReserved": false,
      "hasUnalignedOsTypes": false
    }
  ],
  "count": 2
}
```


The following is an example of output when getting information about the servers whose nicknames are hostC and hostD respectively, and to which host groups are added (if `hostC` and `hostD` are specified for the query parameter `nickname` of the corresponding servers).


```
{
  "data": [
    {
      "id": 8,
      "nickname": "hostC",
      "protocol": "Undefined",
      "osType": "Undefined",
      "totalCapacity": 0,
      "usedCapacity": 0,
      "numberOfPaths": 0,
      "isInconsistent": false,
      "modificationInProgress": false,
      "compatibility": "DKCMAIN_8802010000",
```

```
      "isReserved": true,
      "hasUnalignedOsTypes": false
    },
    {
      "id": 9,
      "nickname": "hostD",
      "protocol": "Undefined",
      "osType": "Undefined",
      "totalCapacity": 0,
      "usedCapacity": 0,
      "numberOfPaths": 0,
      "isInconsistent": false,
      "modificationInProgress": false,
      "compatibility": "DKCMAIN_8802010000",
      "isReserved": true,
      "hasUnalignedOsTypes": false
    }
  ],
  "count": 2
}
```


Attribute	Type	Description
id	int	Server ID
nickname	string	Server nickname
protocol	string	Protocol One of the following values appears: ▪ `FC` ▪ `iSCSI` ▪ `Undefined` This value appears if host groups are added to the server.
osType	string	OS type One of the following values appears: ▪ `Linux` ▪ `DeprecatedVMware` ▪ `HP-UX` ▪ `OpenVMS` ▪ `Tru64` ▪ `Solaris` ▪ `NetWare`

Attribute	Type	Description
		▪ `DeprecatedWindows` ▪ `AIX` ▪ `VMware` ▪ `Windows` ▪ `Undefined` This value appears if host groups are added to the server. ▪ `Unknown`
totalCapacity	long	Total capacity of attached volumes (MiB)
usedCapacity	long	Used capacity of attached volumes (MiB)
numberOfPaths	int	Number of HBAs registered on the server
isInconsistent	boolean	Whether the server configuration information is consistent or not: ▪ `true`: Inconsistent ▪ `false`: Consistent
modificationInProgress	boolean	Not currently used
compatibility	string	Not currently used
isReserved	boolean	The following information appears: whether host groups are added to the server. ▪ `true`: Host groups are added to the server. ▪ `false`: Host groups are not added to the server.
hasUnalignedOsTypes	boolean	The following information appears: whether inconsistencies exist in the information on the server if a host group (or an iSCSI target) is added to a server for which the OS type is defined, and if the value of the host mode defined for the host group differs from the value of the server's OS type. ▪ `true`: Inconsistencies exist ▪ `false`: Inconsistencies do not exist

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/servers
```

Getting information about a specific server

The following request gets information about the server corresponding to the specified server ID.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/servers/object-ID
```

Request message

Object ID

Specify the `id` value obtained by getting information about the server.

Attribute	Type	Description
id	int	(Required) Server ID

Query parameters

None.

Body

None.

Response message

Body

The following is an example of the output when getting information about a server (ID `10`) that is connected by using Fibre Channel:

```
{
  "id": 10,
  "nickname": "hostA",
  "protocol": "FC",
  "osType": "Linux",
```

```
  "osTypeOptions":[2,22,25,68],
  "totalCapacity": 1024,
  "usedCapacity": 42,
  "numberOfVolumes": 2,
  "numberOfPaths": 2,
  "paths": [
    {
      "hbaWwn": "000000102ceceec9",
      "portIds": [
        "CL1-A"
      ]
    },
    {
      "hbaWwn": "1111111111111111",
      "portIds": [
        "CL1-A"
        ]
    }
  ],
  "isInconsistent": false,
  "modificationInProgress": false,
  "compatibility": "DKCMAIN_8802010000",
  "isReserved": false,
  "hasNonFullmeshLuPaths": false,
  "hasUnalignedOsTypes": false,
  "hasUnalignedOsTypeOptions": false
}
```


The following is an example of the output when getting information about a server (server ID: `11`) that is connected by using iSCSI:


```
{
  "id": 11,
  "nickname": "hostB",
  "protocol": "iSCSI",
  "osType": "Linux",
  "osTypeOptions": [2,22,25,68],
  "totalCapacity": 1024,
  "usedCapacity": 42,
  "numberOfPaths": 2,
  "paths": [
    {
      "iscsiName": "iqn.1994-05.com.redhat:496799ba93",
      "portIds": [
        "CL1-B"
      ]
    },
    {
      "iscsiName": "iqn.1994-05.com.redhat:a7526e46aa",
      "portIds": [
```

```
        "CL1-B"
        ]
    }
  ],
  "isInconsistent": false,
  "modificationInProgress": false,
  "compatibility": "DKCMAIN_8802010000",
  "isReserved": false,
  "hasNonFullmeshLuPaths": false,
  "hasUnalignedOsTypes": false,
  "hasUnalignedOsTypeOptions": false
}
```


This operation obtains the values of attributes obtained as part of server information, as well as the values of the following attributes.

Attribute	Type	Description
osTypeOptions	int[]	Option for the OS type
numberOfVolumes	int	Number of attached volumes
paths	object[]	Path information defined in the WWN of the HBA: ▪ `hbaWwn` (string) WWN of the HBA Appears when the protocol is FC. ▪ `iscsiName` (string) iSCSI name Appears when the protocol is iSCSI. ▪ `portIds` (string[]) Port IDs of assigned ports
hasNonFullmeshLuPaths	boolean	The following information appears: whether there are any volumes for which paths are not defined between all ports registered on the specified server and all volumes connected to the server: ▪ `true`: There is at least one volume for which no path is defined. ▪ `false`: Paths are defined for all volumes.

Attribute	Type	Description
hasUnalignedOsTypeOptions	boolean	The following information appears: whether inconsistencies exist in the information on the server for which the OS type option is defined, and if the value of the host mode option defined for the host group differs from the value of the server's OS type option. ▪ `true`: Inconsistencies exist ▪ `false`: Inconsistencies do not exist

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/servers/10
```

Registering a server

The following request registers, in a storage system, information about a server. You can also create a server to which host groups are to be added by specifying only the nickname of that server.

Execution permission

Storage Administrator (Provisioning)

Request line

```
POST base-URL/simple/v1/objects/servers
```

Request message

Object ID

None.

Query parameters

None.

Body

In the example shown below, the following information is specified to register, to a storage system, a server that is connected by using Fibre Channel:

- Server nickname: `hostA`
- Protocol: `FC`
- OS type: `Linux`

```
{
  "serverNickname": "hostA",
  "protocol": "FC",
  "osType": "Linux"
}
```

In the example shown below, the following information is specified to register, to a storage system, a server that is connected by using iSCSI:

- Nickname: `hostB`
- Protocol: `iSCSI`
- OS type: `Linux`

```
{
  "serverNickname": "hostB",
  "protocol": "iSCSI",
  "osType": "Linux"
}
```

In the following example, a server to which host groups are to be added is created:

```
{
  "serverNickname": "hostC",
  "isReserved": true
}
```

Attribute	Type	Description
serverNickName	string	(Required) Specify a server nickname consisting of 1 through 229 characters. You can use alphanumeric characters (`0` through `9`, `A` through `Z`, `a` through `z`), space characters, and the following symbols: Comma (`,`), Hyphen (`-`), periods (`.`), forward slash (`/`), colon (`:`), at sign (`@`), back slash (`\`), underscore (`_`) This attribute is case-sensitive.

Attribute	Type	Description
		You can use a space character between usable characters, but cannot use it at the beginning or end of the server nickname. You cannot use a hyphen (–) at the beginning of the server nickname.
protocol	string	(Optional) Protocol You can specify the following values: ▪ `FC` ▪ `iSCSI` If you did not specify `true` for the `isReserved` attribute, you must specify this attribute. If you specified `true` for the `isReserved` attribute, any value specified for this attribute will be ignored.
osType	string	(Optional) OS type You can specify the following values: ▪ `Linux` ▪ `HP-UX` ▪ `OpenVMS` ▪ `Tru64` ▪ `Solaris` ▪ `NetWare` ▪ `AIX` ▪ `VMware` ▪ `Windows` If you did not specify `true` for the `isReserved` attribute, you must specify this attribute. If you specified `true` for the `isReserved` attribute, any value specified for this attribute will be ignored.
osTypeOptions	int[]	(Optional) Specify the Option for the OS type as an integer from `0` through `127`. For the valid values, see the *Provisioning Guide*.

Attribute	Type	Description
		If you omit this attribute but specify the OS type, the value for the specified OS type will be automatically set. If you specified `true` for the `isReserved` attribute, any value specified for this attribute will be ignored.
isReserved	boolean	(Optional) Specify whether to create, in the storage system, a server to which host groups are to be added. ▪ `true`: Create a server to which host groups are to be added. ▪ `false`: Do not create a server to which host groups are to be added. If this attribute is omitted, `false` is assumed.

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the register server information request

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/servers
```

Changing server settings

The following request changes the settings of the server corresponding to the specified server ID.

Execution permission

Storage Administrator (Provisioning)

Request line

```
PATCH base-URL/simple/v1/objects/servers/object-ID
```

Request message

Object ID

Specify the `id` value obtained by getting information about the server.

Attribute	Type	Description
id	int	(Required) Server ID

Query parameters

None.

Body

Only the specified attributes are changed.

The following is an example of changing the OS type to `Solaris` for a Fibre Channel-connected server (ID `10`):

```
{
  "osType": "Solaris"
}
```

The following is an example of changing the nickname of an iSCSI-connected server (ID `11`) to `My_REST_API_HOST`:

```
{
  "nickname": "My_REST_API_HOST"
}
```

Attribute	Type	Description
nickname	string	(Optional) Specify a server nickname consisting of 1 through 229 characters.

Attribute	Type	Description
		You can use alphanumeric characters (`0` through `9`, `A` through `Z`, `a` through `z`), space characters, and the following symbols: Comma (`,`), Hyphen (`-`), period (`.`), forward slash (`/`), colon (`:`), at sign (`@`), back slash (`\`), underscore (`_`) This attribute is case-sensitive. You can use a space character between characters, but cannot use it at the beginning or end of the server nickname. You cannot use a hyphen (`-`) at the beginning of the server nickname. This attribute cannot be specified together with the following attributes: ▪ `osType` ▪ `osTypeOptions`
osType	string	(Optional) OS type You can specify the following values: ▪ `Linux` ▪ `HP-UX` ▪ `OpenVMS` ▪ `Tru64` ▪ `Solaris` ▪ `NetWare` ▪ `AIX` ▪ `VMware` ▪ `Windows` This attribute cannot be specified together with the following attribute: ▪ `nickName`
osTypeOptions	int[]	(Optional) Specify the option for the OS type as an integer from `0` through `127`. For the specifiable numbers, see the *Provisioning Guide*.

Attribute	Type	Description
		If you do not specify the OS type, this value will be overwritten by the value that is currently set for the OS type. If a value is already set for the OS type option and you do not want to overwrite the value, specify the OS type option that is currently set. This attribute cannot be specified together with the following attribute: ▪ `nickName`

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the change server settings request

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X PATCH --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/servers/10
```

Deleting a server

The following request deletes information registered in a storage system about the server corresponding to the specified server ID.

Execution permission

Storage Administrator (Provisioning)

Request line

```
DELETE base-URL/simple/v1/objects/servers/object-ID
```

Request message

Object ID

Specify the `id` value obtained by getting information about the server.

Attribute	Type	Description
id	int	(Required) Server ID

Query parameters

None.

Body

Attribute	Type	Description
keepLunConfig	boolean	(Optional) Specify whether to delete information about the server while maintaining the resource attachment information. ▪ `true`: Delete the information. ▪ `false`: Do not delete the information. If you mistakenly add the wrong host group to a server, specify `true` to delete information about the server. Then, re-create the server and add the correct host group. If you specify `true`, you will no longer be able to manage the resource attachment information. Normally, we recommend that you do not specify this attribute.

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the delete server information request

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE
https://192.0.2.100/ConfigurationManager/simple/v1/objects/servers/10
```

Adding host groups (iSCSI target) to a server

The following request adds host groups (iSCSI target) to a specified server (specified by its server ID).

Important:

You cannot add a host group (iSCSI target) if it meets any of the following conditions:

- The security of the port associated with a host group (iSCSI target) is invalid.
- The protocol of the host group (iSCSI target) differs from that of the server to which the host group is to be added.
- The host group (iSCSI target) contains a volume that is not a virtual volume created from a pool.
- The host group (iSCSI target) is already added to another server.
- The host group (iSCSI target) is attached to a virtual storage machine.

Execution permission

Storage Administrator (Provisioning)

Request line

```
POST base-URL/simple/v1/objects/servers/object-ID/actions/add-host-groups/
invoke
```

Request message

Object ID

Specify the `id` value obtained by getting information about the server.

Attribute	Type	Description
id	int	(Required) Server ID

Query parameters

None.

Body

The following is an example of adding a host group by specifying the host group ID:

```
{
  "hostGroups": [
    {
      "portId": "CL1-C",
      "hostGroupId": 1
    }
  ]
}
```

The following is an example of adding an iSCSI target by specifying the host group name:

```
{
  "hostGroups": [
    {
      "portId": "CL1-D",
      "hostGroupName": "My_REST_API_HOST"
    }
  ]
}
```

Attribute	Type	Description
hostGroups	object[]	Information about the host group (iSCSI target) to be added ▪ (Required) `portId`: (string) Specify the port ID by using five characters. ▪ (Optional) `hostGroupId`: (int) Specify the host group (iSCSI target) ID by using an integer in the range from 1 through 254. Specify either `hostGroupId` or `hostGroupName`. Do not specify both of these parameters together. ▪ (Optional) `hostGroupName`: (string) Specify the host group name (iSCSI target name) by using a character string consisting of 1 through 64 characters. Specify either `hostGroupId` or `hostGroupName`. Do not specify both of these parameters together.

Response message

Body

Attribute	Type	Description
statusResource	string	URL to be used to obtain the execution results of the processing to add host groups (iSCSI targets)

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/servers/8/actions/add-host-groups/invoke
```

Getting information about the WWN of an HBA (iSCSI name)

The following request obtains information about the WWN of an HBA (or the iSCSI name) of the server that corresponds to the specified server ID.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/servers/object-ID/hbas
```

Request message

Object ID

Specify the `id` value obtained by getting information about the server.

Attribute	Type	Description
id	int	(Required) Server ID

Query parameters

None.

Body

None.

Response message

Body

The following is an example of the output when getting information about the WWN of an HBA of a server (ID `10`) that is connected by using Fibre Channel.

```
{
  "data": [
    {
      "serverId": 10,
      "hbaWwn": "000000102ceccc9",
      "portIds": [
          "CL1-A"
```

```
      ]
    },
    {
      "serverId": 10,
      "hbaWwn": "1111111111111111",
      "portIds": [
          "CL1-A"
        ]
    }
  ],
  "count": 2
}
```


The following is an example of getting information about the iSCSI name of a server (ID `11`) that is connected by using iSCSI.


```
{
  "data": [
    {
      "serverId": 11,
      "iscsiName": "iqn.1994-05.com.redhat:496799ba93",
      "portIds": [
          "CL1-B"
      ]
    },
    {
      "serverId": 11,
      "iscsiName": "iqn.1994-05.com.redhat:a7526e46aa",
      "portIds": [
          "CL1-B"
        ]
    }
  ],
  "count": 2
}
```


Attribute	Type	Description
serverId	int	Server ID
hbaWwn	string	WWN of the HBA
iscsiName	string	iSCSI name
portIds	string[]	List of port IDs of assigned ports

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/servers/10/hbas
```

Getting information about the WWN of an HBA (iSCSI name) of a specific server

The following request obtains information about the WWN of an HBA (or iSCSI name) of a specific server.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/servers/object-ID/hbas/object-ID
```

Request message

Object ID

Combines multiple object IDs that are to be specified.

- To specify the WWN of the HBA

  Specify the `serverId` value and the `hbaWwn` value, obtained by using the API function for getting information about the WWN of the HBA (iSCSI name), in the following format:

  ```
  serverId/hbas/hbaWwn
  ```

Attribute	Type	Description
serverId	int	(Required) Server ID
hbaWwn	string	(Required) WWN of the HBA

- To specify the iSCSI name

  Specify the `serverId` value and the `iscsiName` value, obtained by using the API function for getting information about the WWN of the HBA (iSCSI name), in the following format:

  ```
  serverId/hbas/iscsiName
  ```

Attribute	Type	Description
serverId	int	(Required) Server ID
iscsiName	string	(Required) iSCSI name

Query parameters

None.

Body

None.

Response message

Body

The following is an example of the output when getting information about a server (ID `10`) that is connected using Fibre Channel, and the WWN of the HBA is `000000102ceceec9`:

```
{
  "serverId": 10,
  "hbaWwn": "000000102ceceec9",
  "portIds": [
    "CL1-A"
  ]
}
```

The following is an example of the output when getting information about a server (ID `11`) that is connected using iSCSI, and that has the iSCSI name `iqn.1994-05.com.redhat:496799ba93`:

```
{
  "serverId": 11,
  "iscsiName": "iqn.1994-05.com.redhat:496799ba93",
  "portIds": [
    "CL1-B"
  ]
}
```

Attribute	Type	Description
serverId	int	Server ID
hbaWwn	string	WWN of the HBA If you specified the WWN of the HBA, a valid value appears.
iscsiName	string	iSCSI name If you specified the iSCSI name, a valid value appears.
portIds	string[]	List of port IDs of the assigned ports

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/servers/10/hbas/
000000102ceccc9
```

Adding information about the WWN of an HBA (iSCSI name)

The following request adds information about the WWN of an HBA (or iSCSI name) to the server corresponding to the specified server ID.

For cluster configurations, add the WWNs (or iSCSI names) of all of the servers (nodes) that make up the cluster to the server with the same server ID.

Execution permission

Storage Administrator (Provisioning)

Request line

```
POST base-URL/simple/v1/objects/servers/object-ID/hbas
```

Request message

Object ID

Specify the `id` value obtained by getting information about the server.

Attribute	Type	Description
id	int	(Required) Server ID

Query parameters

None.

Body

The following is an example of adding the WWN `210003e08b0256f9` of an HBA to a Fibre Channel-connected server (ID `10`):

```
{
  "hbas": [
    {
      "hbaWwn": "210003e08b0256f9"
    }
  ]
}
```

The following is an example of adding the iSCSI name `iqn.myrestapiiscsi20150907` in the iqn format to an iSCSI-connected server (ID `11`):

```
{
  "hbas": [
    {
      "iscsiName": "iqn.myrestapiiscsi20150907"
    }
  ]
}
```

Attribute	Type	Description

hbas	object[]	Information about the WWN of an HBA or the iSCSI name:
		▪ (Optional) `hbaWwn` (string)
		Specifies the WWN of the HBA in 16 characters.
		▪ (Optional) `iscsiName` (string)
		Specifies the iSCSI name.
		Specify this item in the iqn or eui format:
		• iqn format
		Specify a value that begins with `iqn.`. Use no more than 223 characters.
		You can use alphanumeric characters (`0` through `9`, `A` through `Z`, `a` through `z`), and the following symbols:
		Hyphen (`-`), period (`.`), colon (`:`)
		Specification example:
		`iqn.rest.example.of.iqn.form`
		• eui format
		After `eui.`, specify a hexadecimal number. Specify a value consisting of 20 characters.
		You can use alphanumeric characters (`0` through `9`, `A` through `F`, `a` through `f`).
		Specification example:
		`eui.0900ABDC32598D26`
		Specify either `hbaWwn` or `iscsiName`. Do not specify both of these attributes together.

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the add server information about the WWN of an HBA (or iSCSI name) request

> **Note:**
>
> Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97) .

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/servers/10/hbas
```

Deleting information about the WWN of an HBA (iSCSI name) from a server

The following request deletes information about the WWN of an HBA (or the iSCSI name) from a specific server.

Execution permission

Storage Administrator (Provisioning)

Request line

```
DELETE base-URL/simple/v1/objects/servers/object-ID/hbas/object-ID
```

Request message

Object ID

Combines multiple object IDs that are to be specified.

- To specify the WWN of the HBA

  Specify the `serverId` value and the `hbaWwn` value, obtained by using the API function for getting information about the WWN of the HBA (iSCSI name), in the following format:

  ```
  serverId/hbas/hbaWwn
  ```

Attribute	Type	Description
serverId	int	(Required) Server ID
hbaWwn	string	(Required) WWN of the HBA

- To specify the iSCSI name

  Specify the `serverId` value and the `iscsiName` value, obtained by using the API function for getting information about the WWN of the HBA (iSCSI name), in the following format:

  ```
  serverId/hbas/iscsiName
  ```

Attribute	Type	Description
serverId	int	(Required) Server ID
iscsiName	string	(Required) iSCSI name

Query parameters

None.

Body

None.

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the delete the information about the WWN of an HBA (or the iSCSI name) from the server request

> **Note:**
>
> Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97) .

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE
https://192.0.2.100/ConfigurationManager/simple/v1/objects/servers/10/
hbas/210003e08b0256f9
```

Getting a list of server path information

The following request obtains the path information of the specific server that corresponds to the specified server ID. You can also specify filter conditions.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/servers/object-ID/paths
```

Request message

Object ID

Specify the `id` value obtained by getting information about the server.

Attribute	Type	Description
id	int	(Required) Server ID

Query parameters

Parameter	Type	Filter Condition
hbaWwn	string	(Optional) WWN of the HBA Path information is obtained about the server that has the WWN of the HBA that is a perfect match to the specified value.

Parameter	Type	Filter Condition
iscsiName	string	(Optional) iSCSI name Path information is obtained about the server that has the iSCSI name that is a perfect match to the specified value.
portId	string	(Optional) Port ID of the assigned port Path information is obtained about the server that has the port ID of the assigned port that is a perfect match to the specified value.

Body

None.

Response message

Body

The following is an example of the output when getting the path information of a Fibre Channel-connected server (ID `10`):

```
{
  "data": [
    {
      "id": "000000102cceccc9,CL1-A",
      "serverId": 10,
      "hbaWwn": "000000102cceccc9",
      "portId": "CL1-A"
    },
    {
      "id": "1111111111111111,CL1-A",
      "serverId": 10,
      "hbaWwn": "1111111111111111",
      "portId": "CL1-A"
    }
  ],
  "count": 2
}
```

The following is an example of the output when getting the path information of an iSCSI-connected server (ID `11`):

```
{
  "data": [
    {
      "id": "iqn.1994-05.com.redhat:496799ba93,CL1-B",
```

```
      "serverId": 11,
      "iscsiName": "iqn.1994-05.com.redhat:496799ba93",
      "portIds": "CL1-B"
    },
    {
      "id": "iqn.1994-05.com.redhat:a7526e46aa,CL1-B",
      "serverId": 11,
      "iscsiName": "iqn.1994-05.com.redhat:a7526e46aa",
      "portIds": "CL1-B"
    }
  ],
  "count": 2
}
```


The following is an example of the output when getting the path information of server with no paths defined:


```
{
  "data": [],
  "count": 0
}
```


Attribute	Type	Description
id	string	The WWN of the HBA (or iSCSI name) and the port ID of the assigned port appear in a format in which they are linked by a comma.
serverId	int	Server ID
hbaWwn	string	WWN of the HBA
iscsiName	string	iSCSI name
portId	string	Port ID of the assigned port

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/servers/10/paths
```

Getting information about the paths between a server and a specific port

The following request obtains information about the paths between the server and the specific port that corresponds to the specified server ID, WWN of an HBA (or iSCSI name), and port ID.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/servers/object-ID/paths/object-ID
```

Request message

Object ID

Combine multiple object IDs to be specified.

Specify the `serverId` value and the `id` value, obtained by using the API function for getting the path information of the server, in the following format.

```
serverId/paths/id
```

For the *id* value, you can specify whether to obtain the information in a format that is a combination of the `hbaWwn` (or `iscsiName`) value and the `portId` value.

- To specify the WWN of the HBA:

  ```
  serverId/paths/hbaWwn,portId
  ```

Attribute	Type	Description
serverId	int	(Required) Server ID
hbaWwn	string	(Required) WWN of the HBA
portId	string	(Required) Port ID

- To specify the iSCSI name:

```
serverId/paths/iscsiName,portId
```

Attribute	Type	Description
serverId	int	(Required) Server ID
iscsiName	string	(Required) iSCSI name
portId	string	(Required) Port ID

Query parameters

None.

Body

None.

Response message

Body

The following is an example of the output when getting information about paths between a server (ID `10`, WWN of the HBA: `000000102ceeccc9`) that is connected using Fibre Channel, and a specific port (ID `CL1-A`):

```
{
  "id": "000000102ceeccc9,CL1-A",
  "serverId": 10,
  "hbaWwn": "000000102ceeccc9",
  "portId": "CL1-A"
}
```

The following is an example of the output when getting information about paths between a server (ID `11`, iSCSI name: `iqn.1994-05.com.redhat:496799ba93`) that is connected using iSCSI, and a specific port (ID `CL1-B`):

```
{
  "id": "iqn.1994-05.com.redhat:496799ba93,CL1-B",
  "serverId": 11,
  "iscsiName": "iqn.1994-05.com.redhat:496799ba93",
  "portId": "CL1-B"
}
```

For details about the attributes to be obtained, see the description of the API function for getting server path information.

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/servers/10/paths/
000000102ceccc9,CL1-A
```

Adding path information to a server

The following request adds path information to the specific server that corresponds to the specified server ID.

Execution permission

Storage Administrator (Provisioning)

Request line

```
POST base-URL/simple/v1/objects/servers/object-ID/paths
```

Request message

Object ID

Specify the `id` value obtained by getting information about the server.

Attribute	Type	Description
id	int	(Required) Server ID

Query parameters

None.

Body

The following is an example of adding path information for a server (ID `10`) that is connected using Fibre Channel:

```
{
  "hbaWwn": "210003e08b0256f9",
  "portIds": [
    "CL1-A"
  ]
}
```

The following is an example of adding path information for a server (ID `11`) that is connected using iSCSI:

```
{
  "iscsiName": "iqn.myrestapiiscsi20150907",
  "portIds": [
    "CL1-B"
  ]
}
```

Attribute	Type	Description
hbaWwn	string	(Optional) The WWN of the HBA in 16 characters. Specify either `hbaWwn` or `iscsiName`. Do not specify both of these attributes together.
iscsiName	string	(Optional) The iSCSI name. Specify this item in iqn or eui format: ▪ iqn format Specify a value that begins with `iqn.`. Use no more than 223 characters. You can use alphanumeric characters (`0` through `9`, `A` through `Z`, `a` through `z`), and the following symbols: Hyphen (`-`), period (`.`), colon (`:`) Specification example: `iqn.rest.example.of.iqn.form` ▪ eui format After "`eui.`", specify a hexadecimal number. Specify a value consisting of 20 characters. You can use alphanumeric characters (`0` through `9`, `A` through `F`, `a` through `f`). Specification example: `eui.0900ABDC32598D26` Specify either `hbaWwn` or `iscsiName`. Do not specify both of these attributes together.
portIds	string[]	(Required) The port IDs of the assigned ports.

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the add path information to the server request

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/servers/10/paths
```

Deleting path information from a server

The following request deletes path information from a specific server.

Execution permission

Storage Administrator (Provisioning)

Request line

```
DELETE base-URL/simple/v1/objects/servers/object-ID/paths/object-ID
```

Request message

Object ID

Combines multiple object IDs that are to be specified.

Specify the `serverId` value and the `id` value, obtained by using the API function for getting the path information of the server, in the following format.

```
serverId/paths/id
```

For the *id* value, you can specify whether to obtain the information in a format that is a combination of the `hbaWwn` (or `iscsiName`) value and the `portId` value.

- To specify the WWN of the HBA:

```
serverId/paths/hbaWwn,portId
```

Attribute	Type	Description
serverId	int	(Required) Server ID
hbaWwn	string	(Required) WWN of the HBA
portId	string	(Required) Port ID

- To specify the iSCSI name:

```
serverId/paths/iscsiName,portId
```

Attribute	Type	Description
serverId	int	(Required) Server ID
iscsiName	string	(Required) iSCSI name
portId	string	(Required) Port ID

Query parameters

None.

Body

None.

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the delete the path information from the server request

> **Note:**
>
> Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE
https://192.0.2.100/ConfigurationManager/simple/v1/objects/servers/10/
paths/210003e08b0256f9,CL1-A
```

Getting information about iSCSI targets

The following request obtains information about the iSCSI target that corresponds to the specified server ID.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/servers/object-ID/target-iscsi-ports
```

Request message

Object ID

Specify the `id` value obtained by getting information about the server.

Attribute	Type	Description
id	int	(Required) Server ID

Query parameters

None.

Body

None.

Response message

Body

The following is an example of the output when getting information about the iSCSI target of an iSCSI-connected server (ID `11`):

```
{
  "data": [
    {
      "portId": "CL1-B",
      "targetIscsiName": "iqn.rest.example.of.iqn.hostB"
    }
  ],
  "count": 1
}
```

Attribute	Type	Description
portId	string	Port ID of the assigned port
targetIscsiName	string	iSCSI name of the iSCSI target

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/servers/11/target-iscsi-
ports
```

Getting information about a specific iSCSI target

The following request obtains information about the iSCSI target that corresponds to the specified server ID and port ID.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/servers/object-ID/target-iscsi-ports/object-
ID
```

Request message

Object ID

Specify the `id` value and the `portId` value, obtained by using the API function for getting information about the server, in the following format:

```
id/target-iscsi-ports/portId
```

Attribute	Type	Description
id	int	(Required) Server ID
portId	string	(Required) Port ID

Query parameters

None.

Body

None.

Response message

Body

The following is an example of the output when getting information about the iSCSI target of an iSCSI-connected server (server ID: `11`, port ID: `CL1-B`):

```
{
  "portId": "CL1-B",
  "targetIscsiName": "iqn.rest.example.of.iqn.hostB"
}
```

For details about the attributes to be obtained, see the description of the API function for getting information about iSCSI targets.

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/servers/11/target-iscsi-
ports/CL1-B
```

Changing the settings of an iSCSI target

The following request changes the iSCSI name of the iSCSI target that corresponds to the specified server ID and port ID.

Execution permission

Storage Administrator (Provisioning)

Request line

```
PATCH base-URL/simple/v1/objects/servers/object-ID/target-iscsi-ports/
object-ID
```

Request message

Object ID

Combines multiple object IDs that are to be specified.

Specify the `id` value and the `portId` value, obtained by using the API function for getting information about the server, in the following format:

```
id/target-iscsi-ports/portId
```

Attribute	Type	Description
id	int	(Required) Server ID
portId	string	(Required) Port ID

Query parameters

None.

Body

The following is an example of changing the iSCSI name to `iqn.rest.example.of.iqn.hostB` for an iSCSI target of an iSCSI-connected server (server ID: `11`, port ID: `CL1-B`):

```
{
  "targetIscsiName": "iqn.rest.example.of.iqn.hostB"
}
```

Attribute	Type	Description
targetIscsiName	string	(Required) Specifies the iSCSI name of the iSCSI target.

Attribute	Type	Description

		Specify this item in iqn or eui format:
		▪ iqn format
		Specify a value that begins with `iqn.`. Use no more than 223 characters.
		You can use alphanumeric characters (`0` through `9`, `A` through `Z`, `a` through `z`), and the following symbols:
		Hyphen (-), period (`.`), colon (`:`)
		Specification example:
		`iqn.rest.example.of.iqn.form`
		▪ eui format
		After "`eui.`", specify a hexadecimal number. Specify a value consisting of 20 characters.
		You can use alphanumeric characters (`0` through `9`, `A` through `F`, `a` through `f`).
		Specification example:
		`eui.0900ABDC32598D26`

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the change the iSCSI name of the server request

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X PATCH --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/servers/11/target-iscsi-ports/CL1-B
```

Volume management

You can use the REST API to obtain volume information, create new volumes, attach existing volumes to servers, and increase the capacity of volumes that have high usage rates.

Getting volume information

The following request obtains volume information. You can also specify filter conditions.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/volumes
```

Request message

Object ID

None.

Query parameters

With each request, you can obtain information about a maximum of 500 volumes. To obtain information about additional volumes, execute the API request multiple times by using a combination of the `count` and `startVolumeId` parameters. By specifying the `count` parameter, you can also filter the volume information you require.

Parameter	Type	Filter Condition
poolId	int	(Optional) ID of the pool to which the volumes belong If `poolName` is specified, do not specify this parameter.
poolName	string	(Optional) Name of the pool to which the volumes belong

Parameter	Type	Filter Condition
		Volume information will also be obtained if part of the specified value matches the pool name. If `poolId` is specified, do not specify this parameter. If both `poolId` and this parameter are specified, this parameter is ignored.
serverId	int	(Optional) ID of the server to which the volumes have been attached If `serverNickName` is specified, do not specify this parameter.
serverNickName	string	(Optional) Nickname of the server to which the volumes have been attached Volume information will also be obtained if part of the specified value matches the nickname of a server. If `serverId` is specified, do not specify this parameter. If both `serverId` and this parameter are specified, this parameter is ignored.
nickname	string	(Optional) Nickname of the volumes Volume information will also be obtained if part of the specified value matches the nickname of the volume.
minTotalCapacity	long	(Optional) Minimum capacity of the volumes (MiB) Specify the minimum capacity of the volumes for which you want to obtain information as an integer in the range from `47` through `268435456`.
maxTotalCapacity	long	(Optional) Maximum capacity of the volumes (MiB) Specify the maximum capacity of the volumes for which you want to obtain information as an integer in the range from `47` through `268435456`.
minUsedCapacity	long	(Optional) Minimum usage capacity of the volumes (MiB)

Parameter	Type	Filter Condition
		Specify the minimum usage capacity of the volumes for which you want to obtain information as an integer in the range from `0` through `268435456`.
maxUsedCapacity	long	(Optional) Maximum usage capacity of the volumes (MiB) Specify the maximum usage capacity of the volumes for which you want to obtain information an integer in the range from `0` through `268435456`.
startVolumeId	int	(Optional) Specify the ID of the volume for which you want to start acquiring information, by using a value in the range from `0` through `65279`.
count	int	(Optional) Specify the number of volumes for which information is to be obtained, by using a value in the range from `1` through `500`. If this parameter is omitted, `500` is considered.

Body

None.

Response message

Body

The following is an example of output of obtained information for a volume that belongs to the pool with ID `63` (`63` is specified in the `poolId` query parameter):

```
{
  "data": [
    {
      "id": 100,
      "nickname": "JH-26216_DP",
      "poolId": 63,
      "poolName": "NASOS",
      "totalCapacity": 1024,
      "usedCapacity": 42,
      "numberOfConnectingServers": 2,
      "numberOfSnapshots": 2,
      "volumeTypes": []
    },
    {
      "id": 101,
```

```
      "nickname": "JH-26216_DP",
      "poolId": 63,
      "poolName": "NASOS",
      "totalCapacity": 1024,
      "usedCapacity": 0,
      "numberOfConnectingServers": 2,
      "numberOfSnapshots": 2,
      "volumeTypes": [
        "Snapshot"
      ]
    }
  ],
  "count": 2,
  "totalCount": 2,
  "hasNext": false
}
```


Attribute	Type	Description
id	int	Volume ID
nickname	string	Nickname
poolId	int	ID of the pool to which the volume belongs
poolName	string	Name of the pool to which the volume belongs This attribute do not appear when the volume is being created or deleted.
totalCapacity	long	Total capacity of the volume (MiB)
usedCapacity	long	Used capacity of the volume (MiB) `0` appears when the volume is being created or deleted.
numberOfConnectingServers	int	Number of connected servers
numberOfSnapshots	int	Number of snapshots
volumeTypes	object[]	Volume Types ▪ `Snapshot`: Volumes that are created from snapshots and for which volume IDs are defined. ▪ `System`: Volumes used for deduplication system data volumes or journal volumes. ▪ `Command Device`: Volumes used as command devices.

Attribute	Type	Description
		▪ `SLU`: Volumes with the SLU (Subsidiary Logical Unit) attribute. ▪ `AttachedUnmanaged`: Volumes are attached using other management tools.

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/volumes?poolId=63
```

Getting information about a specific volume

The following request gets information about the volume for the specified volume ID.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/volumes/object-ID
```

Request message

Object ID

Specify the `id` value obtained by getting information about volumes.

Attribute	Type	Description
id	int	(Required) Volume ID

Query parameters

None.

Body

None.

Response message

Body

The following is an example of getting information about a volume with ID `100`:


```
{
  "id": 100,
  "nickname": "JH-26216_DP",
  "poolId": 63,
  "poolName": "NASOS",
  "totalCapacity": 1024,
  "usedCapacity": 42,
  "freeCapacity": 982,
  "reservedCapacity": 0,
  "savingSetting": "DISABLE",
  "capacitySavingStatus": "Rehydrating",
  "capacitySavingProgress": 40,
  "numberOfConnectingServers": 2,
  "numberOfSnapshots": 2,
  "luns": [
    {
      "lun": 1,
      "serverId": 10,
      "portId": "CL1-A"
    },
    {
      "lun": 1,
      "serverId": 11,
      "portId": "CL2-A"
    }
  ],
  "volumeTypes": []
}
```


The obtained attributes include the following attributes, in addition to the attributes obtained when volume information is obtained:

Attribute	Type	Description
freeCapacity	long	Free space in the volume (MiB) The total capacity of the volume appears when the volume is being created or deleted.
reservedCapacity	long	Capacity of the reserved pages in the volume (MiB)

Attribute	Type	Description
savingSetting	string	Whether the capacity saving feature (deduplication and compression) is enabled: ▪ `DEDUPLICATION_AND_COMPRESSION`: The capacity saving feature (deduplication and compression) is enabled. ▪ `COMPRESSION`: The capacity saving feature (compression) is enabled. ▪ `DISABLE`: No settings are specified.
capacitySavingStatus	string	Setting of the capacity saving feature (deduplication and compression) ▪ `Disabled`: The capacity saving feature is disabled. ▪ `Enabled`: The capacity saving feature is enabled. ▪ `Rehydrating`: The capacity saving feature is being disabled. ▪ `Enabling`: The capacity saving feature is being enabled. ▪ `Deleting`: The volumes for which the capacity saving feature is enabled are being deleted. ▪ `Failed`: The data on the deduplication system data volumes is invalid.
capacitySavingProgress	int	Progress rate of the capacity saving feature (deduplication and compression) (%)
luns	object[]	Array of LUN configuration information: ▪ `lun` (int): LU number ▪ `serverId` (int): Server ID ▪ `portId` (string): ID of the port to which the LUN is assigned

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/volumes/100
```

Creating a volume

The following request creates a volume with the specified capacity and nickname.

Execution permission

Storage Administrator (Provisioning)

Request line

```
POST base-URL/simple/v1/objects/volumes
```

Request message

Object ID

None.

Query parameters

None.

Body

The following is an example of creating a volume, in the pool with ID 63, with a capacity of 100 MiB, and the nickname `REST_API_100MVolume`, for which deduplication and compression are enabled:

```
{
  "capacity": 100,
  "number": 1,
  "nicknameParam": {
    "baseName": "REST_API_100MVolume"
  },
  "savingSetting": "DEDUPLICATION_AND_COMPRESSION",
  "poolId": 63
}
```

Attribute	Type	Description
capacity	long	(Required) Capacity of the volume to be created (MiB) Specify a value in the range from `47` through `268435456`.

Attribute	Type	Description
number	int	(Optional) Number of volumes to be created Specify a value in the range from `1` through `1000`. If this attribute is omitted, a single volume is created.
nicknameParam	object	Information about the nickname to be assigned to the new volume: ▪ (Required) `baseName` (string) The nickname can contain up to 32 characters. You can use alphanumeric characters (`0` through `9`, `A` through `Z`, `a` through `z`), space characters, and the following symbols: `, - . / : @ \ _` This attribute is case-sensitive. You can use a space character between characters, but you cannot use it at the beginning or end of the nickname. You cannot use a hyphen (–) at the beginning of the nickname. ▪ (Optional) `startNumber` (int) Specify the lowest serial number to be added after the nickname as an integer in the range from `0` to `65279`. ▪ (Optional) `numberOfDigits` (int) Specify the number of digits in the serial number to be added after the nickname as an integer in the range from `1` to `5`. If you specify this attribute, you must also specify `startNumber`. If you specify `startNumber` but omit this attribute, `1` is assumed.

Attribute	Type	Description
		Example: If you specify `AAA` for `baseName`, `1` for `startNumber`, and `3` for `numberOfDigits`, and specify `100` for `number`, indicating the number of volumes to be created, nicknames from `AAA001` through `AAA100` are assigned to the volumes. Specify the values so that the sum of the number of characters specified for `baseName`, and the number of digits in the number generated from the values specified for `startNumber` and `numberOfDigits`, is not more than 32.
savingSetting	string	(Optional) Settings for the capacity saving feature (deduplication and compression) Specify the following values: ▪ `DEDUPLICATION_AND_COMPRESSION`: The capacity saving feature (deduplication and compression) is enabled. The capacity saving feature (compression) is enabled when you enable the capacity saving feature (deduplication). ▪ `COMPRESSION`: The capacity saving feature (compression) is enabled. ▪ `DISABLE`: No settings are specified. If this attribute is omitted, `DISABLE` is considered.
poolId	int	(Required) Specify the ID of the pool in which the volumes are to be created as a value in the range from `0` through `127`.

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the create volume request

> **Note:**
>
> Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/volumes
```

Changing the volume settings

The following request changes the nickname and capacity saving setting for the specified volume.

Execution permission

Storage Administrator (Provisioning)

Request line

```
PATCH base-URL/simple/v1/objects/volumes/object-ID
```

Request message

Object ID

Specify the `id` value obtained by getting information about volumes.

Attribute	Type	Description
id	int	(Required) Volume ID

Query parameters

None.

Body

Only the specified attributes are changed.

The following is an example of changing the nickname of a volume:

```
{
  "nickname": "REST_API_10GVolume"
}
```

The following is an example of disabling the value set for the capacity saving feature (deduplication and compression) of a volume:

```
{
  "savingSetting": "DISABLE"
}
```

Attribute	Type	Description
nickname	string	(Optional) Specify a new nickname of up to 32 characters. You can use alphanumeric characters (`0` through `9`, `A` through `Z`, `a` through `z`), space characters, and the following symbols: Comma (`,`), Hyphen (`-`), period (`.`), forward slash (`/`), colon (`:`), at sign (`@`), back slash (`\`), underscore (`_`) This attribute is case-sensitive. You can use a space character between characters, but you cannot use it at the beginning or end of the nickname. You cannot use a hyphen (`-`) at the beginning of the name. If you specify this attribute, do not specify `savingSetting`.
savingSetting	string	(Optional) Settings for the capacity saving feature (deduplication and compression)

Attribute	Type	Description
		Specify the following values: ▪ `DEDUPLICATION_AND_COMPRESSION`: The capacity saving feature (deduplication and compression) is enabled. The capacity saving feature (compression) is enabled when you enable the capacity saving feature (deduplication). ▪ `COMPRESSION`: The capacity saving feature (compression) is enabled. ▪ `DISABLE`: No settings are specified. If you specify this attribute, the API request might take a long time to complete. If you specify this attribute, do not specify `nickname`.

Response message

Body

- If you changed the volume nickname:

Attribute	Type	Description
affectedResources	string[]	List of URLs for referencing information about the changed volume
operationDetails	object[]	Details about the changed volume For details, see the description of the `operationDetails` attribute of the `commandStatus` object.

- If you changed the settings for the capacity saving feature (deduplication and compression):

Attribute	Type	Description
statusResource	string	URL for referencing information about the changed volume

> **Note:**
>
> Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X PATCH --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/volumes/100
```

Expanding the capacity of a volume

The following request expands the capacity of the volume for the specified volume ID.

Execution permission

Storage Administrator (Provisioning)

Request line

```
POST base-URL/simple/v1/objects/volumes/object-ID/actions/expand/invoke
```

Request message

Object ID

Specify the `id` value obtained by getting information about volumes.

Attribute	Type	Description
id	int	(Required) Volume ID

Query parameters

None.

Body

The following is an example of adding 100 MiB of capacity to a volume:

```
{
  "capacity": 100
}
```

Attribute	Type	Description
capacity	long	(Required) Capacity to be added to the volume (MiB) Specify the capacity to be added, as an integer in the range from `1` through `268435456`.

Response message

Body

Attribute	Type	Description
affectedResources	string[]	List of URLs for referencing information about the capacity added to a volume
operationDetails	object[]	Details about a volume to which capacity was added For details, see the description of the `operationDetails` attribute of the `commandStatus` object.

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/volumes/100/actions/expand/invoke
```

Deleting a volume

The following request deletes the volume corresponding to the specified volume ID.

Note:

The request to delete a volume for which the capacity saving feature (deduplication and compression) is enabled might take some time to complete. For this reason, we recommend that you plan when to delete such volumes.

Execution permission

Storage Administrator (Provisioning)

Request line

```
DELETE base-URL/simple/v1/objects/volumes/object-ID
```

Request message

Object ID

Specify the `id` value obtained by getting information about volumes.

Attribute	Type	Description
id	int	(Required) Volume ID

Query parameters

None.

Body

None.

Response message

Body

Attribute	Type	Description
statusResource	string	URL for referencing information about a volume that was deleted

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE
https://192.0.2.100/ConfigurationManager/simple/v1/objects/volumes/100
```

Getting information about the connections between volumes and servers

The following request gets information about the connections between volumes and servers by using the server IDs or server nicknames of the specified servers.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/volume-server-connections
```

Request message

Object ID

None.

Query parameters

With each request, you can obtain information about a maximum of 2,048 connections between volumes and the server. To obtain information about additional connections between volumes and the server, execute the API request multiple times by using a combination of the `count` and `startVolumeId` parameters. By specifying the `count` parameter, you can also filter the volume information you require.

Parameter	Type	Filter Condition
serverId	int	(Optional) Specify the server ID of the server connected to the volume. The connection information of the server with the specified ID will be obtained. Specify either `serverId` or `serverNickname`. Do not specify both of these parameters together.
serverNickname	string	(Optional) Specify the server nickname of the server connected to the volume. The connection information of the server with the specified server nickname will be obtained. Specify either `serverId` or `serverNickname`. Do not specify both of these parameters together.

Parameter	Type	Filter Condition
startVolumeId	int	(Optional) Specify a value in the range from `0` through `65279` as the volume ID from which to start obtaining information about the connections between volumes and the server.
count	int	(Optional) Specify a value in the range from `1` through `2048` as the number of connections for which to obtain information about connections between volumes and the server. If this parameter is omitted, `2048` is considered.

Body

None.

Response message

Body

The following is an example of getting information about volumes and connection of the server with ID `10` (specifying `10` for the `serverId` query parameter).

```
{
  "data": [
    {
      "id": "100,10",
      "volumeId": 100,
      "serverId": 10,
      "luns": [
        {
          "lun": 1,
          "portId": "CL1-A"
        }
      ]
    },
    {
      "id": "101,10",
      "volumeId": 101,
      "serverId": 10,
      "luns": [
        {
          "lun": 2,
          "portId": "CL1-A"
        }
      ]
```

```
    }
  ],
  "count": 2,
  "totalCount": 2,
  "hasNext": false
}
```

Attribute	Type	Description
id	string	The volume ID and the server ID appear in a format in which they are connected by a comma.
volumeId	int	Volume ID
serverId	int	Server ID
luns	object[]	Information about the LUN appears: ▪ `lun` (int) LU number ▪ `portId` (string) ID of the port to which the LU number is assigned

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/volume-server-
connections?serverId=10
```

Getting information about the connection between a volume and a specific server

The following request gets information about the connection between a volume and a specific server by using a specified volume ID and server ID.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/volume-server-connections/object-ID
```

Request message

Object ID

Specify the `id` value obtained by getting information about the connection between the volume and the server. The value of the obtained attribute `id` appears in the following format, in which the volume ID and the server ID are linked by a comma. You can specify these values (obtained by getting information about the connection between the volume and the server) in the command by using the same format: `volumeId` followed by `serverId`, linked by a comma.

```
volumeId,serverId
```

Attribute	Type	Description
volumeId	int	(Required) Volume ID
serverId	int	(Required) Server ID

Query parameters

None.

Body

None.

Response message

Body

The following is an example of getting information about the connection between the volume with ID `100` and the server with ID `10`:

```
{
  "id": "100,10",
  "volumeId": 100,
  "serverId": 10,
  "luns": [
    {
      "lun": 1,
      "portId": "CL1-A"
    }
  ]
}
```

For details on the attributes that are obtained, see the description of the API function for getting information about the connections between volumes and servers.

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/volume-server-
connections/100,10
```

Attaching a volume to a server

The following request connects all paths between a server and a volume.

Note:

You cannot connect a path between a volume and a server to which a host group (or iSCSI target) that is attached to a virtual storage machine has been added.

Execution permission

Storage Administrator (Provisioning)

Request line

```
POST base-URL/simple/v1/objects/volume-server-connections
```

Request message

Object ID

None.

Query parameters

None.

Body

The following is an example of setting up a connection for a path between the volume with ID `100` and the server with ID `10`:

```
{
  "volumeIds": [
    100
  ],
  "serverIds": [
```

```
    10
  ]
}
```

Parameter	Type	Description
volumeIds	int[]	(Required) Specify the ID of the volume to be attached as a value in the range from `0` through `65279`.
serverIds	int[]	(Required) ID of the server to which the volume is to be attached Specify all of the server IDs between the specified server and the volume as values in the range from `0` through `4095`.

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the connect the volume to the server request

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/volume-server-connections
```

Dettaching a volume from a specific server

The following request disconnects all paths that are registered between a volume and server. The volume and server are specified by the volume ID and server nickname, respectively.

Execution permission

Storage Administrator (Provisioning)

Request line

```
DELETE base-URL/simple/v1/objects/volume-server-connections/object-ID
```

Request message

Object ID

Specify the `id` value obtained by getting information about the connection between the volume and the server. The value of the obtained attribute `id` appears in the following format, in which the volume ID and the server ID are linked by a comma. You can specify these values (obtained by getting information about the connection between the volume and the server) in the command by using the same format: `volumeId` followed by `serverId`, linked by a comma.

```
volumeId,serverId
```

Attribute	Type	Description
volumeId	int	(Required) Volume ID
serverId	int	(Required) Server ID

Query parameters

None.

Body

None.

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the disconnect the volume and the server request

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97) .

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE
https://192.0.2.100/ConfigurationManager/simple/v1/objects/volume-server-
connections/100,10
```

Chapter 15: Snapshot operations using the REST API

Some REST API operations are related to snapshots, such as the operations for acquiring snapshots of volumes in a storage system and saving them to create copies with good cost performance, creating volumes to clone a snapshot so that data in an operation can be used on other servers, and mapping snapshots so that these volumes can be used.

For details about snapshot-related functions, see the *Hitachi Thin Image User Guide*.

Getting a list of snapshot group names

The following request gets a list of snapshot group names.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/snapshot-groups
```

Request message

Object ID

None.

Query parameters

None.

Body

None.

Response message

Body

```
{
  "data": [
    {
      "name": "snapshotGroup"
    },
    {
```

```
      "name": "snapshotGroup2"
    }
  ],
  "count": 2
}
```

Attribute	Type	Description
name	string	Snapshot group name

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/snapshot-groups
```

Getting information about a specific snapshot group

The following request gets information about a snapshot group for the specified snapshot group name.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/snapshot-groups/object-ID
```

Request message

Object ID

Specify a value for the `name` that was obtained by getting the list of snapshot group names.

Attribute	Type	Description
name	string	(Required) Snapshot group name

Note:

Do not specify a snapshot group name that contains a slash (/) or a backslash (\).

To get information about a snapshot group whose name contains a slash (/) or a backslash (\), specify the query parameter `snapshotGroupName`, and execute the API request for getting snapshot information.

Query parameters

With each request, you can obtain information for a maximum of 1000 snapshot groups. To obtain information for more than 1000 snapshot groups, execute the API request several times by using a combination of the `count` and `startId` parameters. By specifying the `count` parameter, you can also filter the snapshot group information to be obtained.

Parameter	Type	Filter Condition
startSnapshotId	string	(Optional) Specify the ID of the snapshot from which to obtain information. If this parameter is omitted, `"0,0"` is assumed.
count	int	(Optional) Specify the number of snapshot groups for which you want to obtain information by using a value in the range from `1` through `1000`. If this parameter is omitted, `1000` is assumed.

Body

None.

Response message

Body

The following is an example of the output when getting information about the snapshot group with the name `snapshotGroup`:

```
{
  "name": "snapshotGroup",
  "snapshots": [
    {
      "masterVolumeId": 100,
      "snapshotId": 3
    },
    {
      "masterVolumeId": 100,
```

```
      "snapshotId": 4
    }
  ],
  "count": 2,
  "totalCount": 2,
  "hasNext": false
}
```

Attribute	Type	Description
name	string	Snapshot group name
snapshots	object[]	The following information about the snapshot is obtained: ▪ `masterVolumeId` (int) ID of the master volume when the snapshot was created ▪ `snapshotId` (int) Snapshot ID The mirror unit number is obtained.

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/snapshot-groups/
snapshotGroup
```

Getting snapshot information

The following request gets information about snapshots. You can specify filter conditions.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/snapshots
```

Request message

Object ID

None.

Query parameters

With each request, you can obtain information for a maximum of 1000 snapshots. To obtain information for more than 1000 snapshots, execute the API request several times by using a combination of the `count` and `startId` parameters. By specifying the `count` parameter, you can also filter the snapshot information.

Parameter	Type	Filter Condition
masterVolumeId	int	(Optional) ID of the master volume from which the snapshots were created
snapshotDateFrom	string	(Optional) Specify the date and time from which snapshots will be obtained, in YYYY-MM-DDThh:mm:ssZ format. Information about snapshots created on and after the date and time specified for this parameter will be obtained.
snapshotDateTo	string	(Optional) Specify the date and time until which snapshots will be obtained, in YYYY-MM-DDThh:mm:ssZ format. Information about snapshots created on and before the date and time specified for this parameter will be obtained.
snapshotGroupName	string	(Optional) Snapshot group name Information about the snapshot that is a perfect match with the specified value is obtained.
startId	string	(Optional) Specify the first snapshot information to be obtained, by specifying the volume ID of the master volume from which the snapshot was created and the snapshot ID, linked by a comma. *masterVolumeId*,*snapshotId* If this parameter is omitted, "0,0" is assumed.
count	int	(Optional) Specify the number of snapshots by using a value in the range from 1 through 1000.

Parameter	Type	Filter Condition
		If this parameter is omitted, `1000` is considered.

Body

None.

Response message

Body

```
{
  "data": [
    {
      "id": "100,3",
      "masterVolumeId": 100,
      "snapshotId": 3,
      "status": "Completed",
      "snapshotDate": "2015-03-20T09:27:35Z",
      "snapshotGroupName": "snapshotGroup",
      "rootVolumeId": 100,
      "poolId": 10,
      "usedCapacityPerRootVolume": 126,
    },
    {
      "id": "100,4",
      "masterVolumeId": 100,
      "snapshotId": 4,
      "status": "Completed",
      "snapshotDate": "2015-03-20T09:27:35Z",
      "snapshotGroupName": "snapshotGroup",
      "rootVolumeId": 100,
      "poolId": 10,
      "usedCapacityPerRootVolume": 126,
    },
  ],
  "count": 2,
  "totalCount": 2,
  "hasNext": false
}
```

Attribute	Type	Description
id	string	ID of the master volume from which the snapshots were created and the snapshot ID, linked by a comma

Attribute	Type	Description
masterVolumeId	int	ID of the master volume from which the snapshots were created
snapshotId	int	Snapshot ID The mirror unit number is obtained.
status	string	Status of the snapshot: ▪ `Creating`: Snapshot creation is in progress ▪ `In Sync`: Snapshot synchronization is complete ▪ `Completed`: Snapshot creation is complete ▪ `Deleting`:Snapshot deletion is in progress ▪ `Restoring`: Snapshot restoration is in progress ▪ `Preparing`: Preparation for cloning is in progress ▪ `Clone Ready`: The snapshot is ready to be cloned ▪ `Cloning`: Cloning is in progress ▪ `Removing`: Removal of the cloning settings is in progress ▪ `Error`: A failure has occurred
snapshotDate	string	Date and time when the snapshot was created
snapshotGroupName	string	Snapshot group name
mappedVolumeId	int	ID of the volume to be created from the snapshots This attribute is obtained only if the ID of the volume for which the snapshot was created is defined.
rootVolumeId	int	ID of the root volume of the snapshots
poolId	int	ID of the pool in which the differential data of the snapshots is stored

Attribute	Type	Description
usedCapacityPerRootVolume	long	The amount of disk space (MiB) occupied by differential information and control information used by the snapshot group, within the pool where the differential data for the snapshots is stored.

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/snapshots
```

Getting information about a specific snapshot

The following request gets information about the snapshot for the specified ID of the master volume when the snapshot was created and the specified snapshot ID.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/snapshots/object-ID
```

Request message

Object ID

Specify the `id` value obtained by requesting snapshot information. The value of the `id` attribute is obtained in the following format, where the ID of the master volume for which the snapshots were created and the snapshot ID are linked by a comma. You can specify these values (obtained by getting snapshot information) in the command by using the same format: `masterVolumeId` followed by `snapshotId`, linked by a comma.

```
masterVolumeId,snapshotId
```

Attribute	Type	Description
masterVolumeId	int	(Required) ID of the master volume from which the snapshots were created
snapshotId	int	(Required) Snapshot ID

Query parameters

None.

Body

None.

Response message

Body

The following is an example of the output when getting information about a specific snapshot (the ID of the master volume from which the snapshot was created: `100`, snapshot ID: `3`).

```
{
  "id": "100,3",
  "masterVolumeId": 100,
  "snapshotId": 3,
  "status": "Completed",
  "snapshotDate": "2015-03-20T09:27:35Z",
  "snapshotGroupName": "snapshotGroup",
  "rootVolumeId": 100,
  "poolId": 10,
  "usedCapacityPerRootVolume": 126
}
```

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/snapshots/100,3
```

Creating a snapshot

The following request creates a snapshot for backing up data or for re-purposing the backed up data. This request also allows you to clone a snapshot.

Execution permission

Storage Administrator (Local Copy)

Request line

```
POST base-URL/simple/v1/objects/snapshots
```

Request message

Object ID

None.

Query parameters

None.

Body

The following is an example of cloning, in the pool with ID `13`, a snapshot from the master volume (ID `100`). In this example, a snapshot group (snapshot name: `snapshotGroup`) is also created at the same time.

```
{
  "params": [
    {
      "masterVolumeId": 100,
      "poolId": 13,
      "snapshotGroupName": "snapshotGroup",
      "type": "Snap Clone"
    }
  ]
}
```

The following is an example of creating, in the pool with ID `13`, a snapshot from the master volume (ID `100`, snapshot group name: `snapshotGroup`):

```
{
  "params": [
    {
      "masterVolumeId": 100,
      "poolId": 13,
      "snapshotGroupName": "snapshotGroup",
      "type": "Snapshot"
    }
  ]
}
```

The following is an example of creating, in the pool with ID `13`, a snapshot from the master volume (ID `100`, snapshot group name: `snapshotGroup`). In this

example, the volume ID is also defined for the volume to be created from the snapshot.

```
{
  "params": [
    {
      "masterVolumeId": 100,
      "poolId": 13,
      "snapshotGroupName": "snapshotGroup",
      "type": "Mapped Snapshot"
    }
  ]
}
```

Attribute	Type	Description
params	object[]	Information about a snapshot to be created ▪ `masterVolumeId` (int) (Required) Specify the ID of the master volume from which snapshots are to be created, as an integer from `0` to `65279`. ▪ `poolId` (int) (Required) Specify the ID of the pool in which the differential data of the snapshot is to be stored, as an integer in the range from `0` to `127`. If the usage rate of the pool exceeds the depletion threshold value, snapshots including the created snapshot might become unusable. Make sure that there is enough free space in the specified pool. Tiered pools cannot be specified. To check the usage rate of the pool and whether the pool has a tiered structure, execute the API function for getting information about a specific pool by specifying a pool ID.

Attribute	Type	Description
		▪ `snapshotGroupName` (string) (Required) Specify a snapshot group name of up to 32 characters. You can use alphanumeric characters (`0` through `9`, `A` through `Z`, `a` through `z`), space characters, and the following symbols: Hyphen (–), period (`.`), forward slash (`/`), colon (`:`), at sign (`@`), back slash (`\`), underscore (_) This attribute is case-sensitive. You can use a space character between characters, but cannot use it at the beginning or end of the snapshot group name. You cannot use a hyphen (–) at the beginning of the snapshot group name. If you specify a new snapshot group name, a snapshot group of that name will be created when the request is executed.

Attribute	Type	Description
		▪ `type` (string) (Required) Snapshot type: • `Snap Clone`: Clone a snapshot. • `Snapshot`: Create a snapshot. • `Mapped Snapshot`: Create a snapshot, and define the volume ID of the volume to be created from the snapshot. If `Snap Clone` or `Mapped Snapshot` is specified for this attribute, a nickname is automatically assigned, in the following applicable format, to the volume that is created from the snapshot. **If `Snap Clone` is specified:** "`Clone of ID:`*ID-of-master-volume-created-from-snapshot*" **If `Mapped Snapshot` is specified:** "`Snapshot of ID:`*ID-of-master-volume-created-from-snapshot*" ▪ `destinationSize` (int) (Optional) Specify the capacity of the volume for which the snapshot is to be created, as an integer from `47` through `268435456`. Specify a value greater than the capacity of the master volume from which the snapshot is to be created. You can specify this attribute only when `Snap Clone` is specified for the `type` attribute. If this attribute is omitted but `Snap Clone` is specified for the `type` attribute, a value equal to the capacity of the master volume from which snapshot is to be created is considered.

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the create snapshot request

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97) .

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/snapshots
```

Mapping a snapshot

The following request maps snapshots based on the specified ID of the master volume from which the snapshots were created and the specified snapshot ID. Mapping a snapshot enables you to use the volume created from the specified snapshot.

Execution permission

Storage Administrator (Local Copy)

Request line

```
POST base-URL/simple/v1/objects/snapshots/object-ID/actions/map/invoke
```

Request message

Object ID

Specify the `id` value obtained by getting snapshot information. The value of the `id` attribute is obtained in the following format, where the ID of the master volume for which the snapshots were created and the snapshot ID are linked by a comma. You can specify these values (obtained by getting snapshot information) in the

command by using the same format: `masterVolumeId` followed by `snapshotId`, linked by a comma.

```
masterVolumeId,snapshotId
```

Attribute	Type	Description
masterVolumeId	int	(Required) ID of the master volume from which the snapshots were created
snapshotId	int	(Required) Snapshot ID

Query parameters

None.

Body

```
{
  "poolId": 13
}
```

Attribute	Type	Description
poolId	int	(Required) Specify the ID of the pool in which the volume to which snapshots are to be mapped was created, as an integer in the range from `0` through `127`.

Note:

When snapshots are mapped, nicknames in the following format are automatically assigned to the volumes created from the snapshots:

"`Snapshot of ID:`*`ID-of-master-volume-created-from-snapshot`*"

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the map a snapshot request

> **Note:**
>
> Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/snapshots/100,3/actions/mount/invoke
```

Restoring a snapshot

The following request restores snapshots based on the specified ID of the master volume from which the snapshots were created and the specified snapshot ID. This request overwrites the snapshot data of the specified master volume from which the snapshots were created.

Execution permission

Storage Administrator (Local Copy)

Request line

```
POST base-URL/simple/v1/objects/snapshots/object-ID/actions/restore/invoke
```

Request message

Object ID

Specify the `id` value obtained by getting snapshot information. The value of the `id` attribute is obtained in the following format, where the ID of the master volume for which the snapshots were created and the snapshot ID are linked by a comma. You can specify these values (obtained by getting snapshot information) in the command by using the same format: `masterVolumeId` followed by `snapshotId`, linked by a comma.

```
masterVolumeId,snapshotId
```

Attribute	Type	Description
masterVolumeId	int	(Required) ID of the master volume from which the snapshots were created
snapshotId	int	(Required) Snapshot ID

Query parameters

None.

Body

None.

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the restore a snapshot request

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST https://
192.0.2.100/ConfigurationManager/simple/v1/objects/snapshots/100,3/actions/
restore/invoke -d ""
```

Deleting a snapshot group

The following request deletes a snapshot group and all snapshots in the snapshot group for the specified snapshot group name.

Execution permission

Storage Administrator (Local Copy)

Request line

```
DELETE base-URL/simple/v1/objects/snapshot-groups/object-ID
```

Request message

Object ID

Specify a value for the `name` that was obtained by getting the list of snapshot group names.

Attribute	Type	Description
name	string	(Required) Snapshot group name

Note:

Do not specify a snapshot group name that contains a slash (/) or a backslash (\).

To delete a snapshot group whose name contains a slash (/) or a backslash (\), execute the API request for deleting a snapshot.

If all snapshots in a snapshot group are deleted, the snapshot group is also deleted automatically.

Query parameters

None.

Body

None.

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the delete the snapshot group request

> **Note:**
>
> Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE
https://192.0.2.100/ConfigurationManager/simple/v1/objects/snapshot-groups/
snapshotGroup
```

Deleting a snapshot

The following request deletes a snapshot for the specified ID of the master volume when the snapshot was created and the specified snapshot ID.

Execution permission

Storage Administrator (Local Copy)

Request line

```
DELETE base-URL/simple/v1/objects/snapshots/object-ID
```

Request message

Object ID

Specify the `id` value obtained by getting snapshot information. The value of the `id` attribute is obtained in the following format, where the ID of the master volume for which the snapshots were created and the snapshot ID are linked by a comma. You can specify these values (obtained by getting snapshot information) in the command by using the same format: `masterVolumeId` followed by `snapshotId`, linked by a comma.

```
masterVolumeId,snapshotId
```

Attribute	Type	Description
masterVolumeId	int	(Required) ID of the master volume from which the snapshots were created

Attribute	Type	Description
snapshotId	int	(Required) Snapshot ID

Query parameters

None.

Body

None.

Response message

Body

Attribute	Type	Description
statusResource	string	URL used to obtain the execution results of the delete a snapshot request

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE
https://192.0.2.100/ConfigurationManager/simple/v1/objects/snapshots/100,3
```

Chapter 16: Configuring a global-active device environment (REST API)

When configuring an environment for using global-active device, you can use the REST API to create external volumes and register quorum disks.

For functions related to global-active device, see the *Global-Active Device User Guide*.

Getting external volume information

The following request obtains information about external volumes. You can also specify filter conditions.

Note:

- External parity group: A parity group created by mapping volumes of an externally connected storage system. External parity groups are used to manage external volumes.
- External volume: A volume created from an external parity group
- External path group: A group consisting of multiple external volumes that use the same external path, which is a route that connects an external connection port to the port of an external storage system.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/external-volumes
```

Request message

Object ID

None.

Query parameters

With each request, you can obtain information about a maximum of 400 external volumes. To obtain information about additional external volumes, execute the API request multiple times by using a combination of the `count` and `startVolumeId` parameters. By specifying the `count` parameter, you can also filter the external volume information you require.

Parameter	Type	Filter Condition
nickname	string	(Optional) Nickname of an external volume External volume information will also be obtained if part of the specified value matches the nickname of an external volume.
minTotalCapacity	long	(Optional) Minimum capacity of the external volumes (MiB) Specify the minimum capacity of the external volumes for which you want to obtain information as an integer in the range from `47` through `268435456`.
maxTotalCapacity	long	(Optional) Maximum capacity of the external volumes (MiB) Specify the maximum capacity of the external volumes for which you want to obtain information as an integer in the range from `47` through `268435456`.
status	string	(Optional) Specify one of the following values to obtain information about volumes in a particular state. ▪ `Normal`: Volumes in a normal state ▪ `Blockade`: Volumes that are blocked
volumeType	string	(Optional) Specify `Quorum Disk` (the quorum disk used on the global-active device) as the volume type of the external volume about which information is to be obtained.
externalParityGroupId	string	(Optional) ID of the external parity group Specify the ID in the format *X*-*Y*. For `X`, you can specify an integer in the range from `1` through `16384`. For `Y`, you can specify an integer in the range from `1` through `4096`.
externalPathGroupId	int	(Optional) Specify the ID of the external path group by using an integer in the range from `0` through `63231`.

Parameter	Type	Filter Condition
startVolumeId	int	(Optional) Specify the ID of the external volume for which you want to start acquiring information, by using a value in the range from `0` through `65279`.
count	int	(Optional) Specify the number of external volumes for which information is to be obtained, by using a value in the range from `1` through `400`. If this parameter is omitted, `400` is assumed.

Body

None.

Response message

Body

The following is an example of the output when information is obtained about an external volume whose external volume type is quorum disk (if `Quorum Disk` is specified for the query parameter `volumeType`):

```
{
    "data": [
        {
            "id": 6400,
            "nickname": "externalVolumeQuorum",
            "totalCapacity": 20480,
            "status": "Normal",
            "externalParityGroupId": "9-500",
            "externalPathGroupId": 600,
            "externalPathOfVolume": [
                {
                    "portId": "CL5-A",
                    "portProtocol": "FC",
                    "externalPortWwn": "50060e8012000c60",
                    "lun": 0
                }
            ],
            "volumeTypes": [
                "Quorum Disk"
            ],
            "externalStorage": {
                "model": "VSP Gx00",
                "serial": "400012",
                "vendor": "HITACHI"
            },
```

```
            "quorumSetting": {
                "quorumId": 0,
                "pairedStorageModel": "M8",
                "pairedStorageSerial": "400011"
            }
        }
    ],
    "count": 1,
    "totalCount": 1,
    "hasNext": false
}
```


Attribute	Type	Description
id	int	External volume ID
nickname	string	Nickname of the external volume This information appears if a nickname is defined.
totalCapacity	long	Total capacity of the external volume (MiB)
status	string	Status of the external volume ▪ `Normal`: Volumes in a normal state ▪ `Blockade`: Volumes that are blocked
externalParityGroupId	string	ID of the external parity group
externalPathGroupId	int	ID of the external path group
externalPathOfVolume	object[]	Information about each external path of the external volume appears. ▪ `portId` (string) ID of the external connection port ▪ `portProtocol` (string) Protocol of the port • `FC` • `iSCSI`

Attribute	Type	Description
		▪ `externalPortWwn` (string) WWN of the port of the externally connected storage system If an iSCSI port is specified, this attribute will not appear. ▪ `lun` (int) LUN assigned to the port of the externally connected storage system
volumeTypes	object[]	Volume type of an external volume ▪ `Quorum Disk`: The volume that is the quorum disk used by global-active device
externalStorage	object	Information about the externally connected storage system appears. ▪ `model` (string) Model of the externally connected storage system For details about the notation that is used when information about the externally connected storage system is output, see the *Hitachi Universal Volume Manager User Guide*. If information about the model cannot be obtained, `OTHER` appears. ▪ `serial` (string) Serial number of the externally connected storage system If a valid value cannot be obtained, this attribute does not appear. ▪ `vendor` (string) Vendor ID of the externally connected storage system If a valid value cannot be obtained, this attribute does not appear.

Attribute	Type	Description
quorumSetting	object	If information about a quorum disk is set, the information will appear. ▪ `quorumId` (int) Quorum disk ID ▪ `pairedStorageModel` (string) Model of the paired storage systems that share the quorum disk • `R8`: VSP F1500, VSP G1000, or VSP G1500 • `M8`: VSP Fx00 models or VSP Gx00 models If information about the model cannot be obtained, `Unknown` will appear. ▪ `pairedStorageSerial` (string) Serial number of the paired storage system that shares the quorum disk

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/external-volumes?
volumeType="Quorum Disk"
```

Getting information about a specific external volume

The following request gets information about the external volume corresponding to the specified external volume ID.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/external-volumes/object-ID
```

Request message

Object ID

Specify the `id` value obtained by getting information about external volumes.

Attribute	Type	Description
id	int	(Required) External volume ID

Query parameters

None.

Body

None.

Response message

Body

The following is an example of getting information about an external volume with a volume ID of `3`:

```
{
    "id": 3,
    "nickname": "externalVolumeForQuorum",
    "totalCapacity": 20480,
    "status": "Normal",
    "volumeIdentifier": "HITACHI 5040000C1965",
    "externalParityGroupId": "8-501",
    "externalPathGroupId": 501,
    "externalPathOfVolume": [
        {
            "portId": "CL5-A",
            "portProtocol": "FC",
            "externalPathGroupId": 501,
            "externalPortWwn": "50060e8012000c60",
            "lun": 501
        }
    ],
    "volumeTypes": [],
    "externalStorage": {
        "model": "M8",
        "serial": "400012",
        "vendor": "HITACHI"
    }
}
```

The obtained attributes include the following attributes, in addition to the attributes obtained when external volume information is obtained:

Attribute	Type	Description
volumeIdentifier	string	An identifier that is used to identify an external volume Example: `HITACHI 5040000C1965`

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/external-volumes/3
```

Creating an external volume

The following request creates an external parity group in a storage system, and then creates (maps) volumes for the storage system that is externally connected to the newly created external parity group. An external path group and external paths are also created at the same time. Volumes created by using this request are called external volumes.

Note:

If the capacity of an external volume exceeds 4,194,304 MiB (4 TB), an external volume with a capacity of 4,194,304 MiB (4TB) is created.

Execution permission

Storage Administrator (Provisioning)

Request line

```
POST base-URL/simple/v1/objects/external-volumes
```

Request message

Object ID

None.

Query parameters

None.

Body


```
{
    "data": [
        {
            "id": "9-400",
            "status": "Normal",
            "externalPathGroupId": 400,
            "externalPath": [
                {
                    "portId": "CL5-A",
                    "portProtocol": "FC",
                    "externalPortWwn": "50060e8012000c60",
                    "lun": 5,
                    "status": "Normal"
                },
                {
                    "portId": "CL6-A",
                    "portProtocol": "FC",
                    "externalPortWwn": "50060e8012000c70",
                    "lun": 6,
                    "status": "Normal"
                }
            ],
            "index": 2050
        }
    ],
    "count": 1,
    "totalCount": 1,
    "hasNext": false
}
```


Attribute	Type	Description
externalParityGroupId	string	(Required) ID of the external parity group Specify a unique (unused) ID that is not being used for the external parity group. Specify the ID in the format *X*-*Y*. For X, you can specify an integer in the range from 1 through 16384. For Y, you can specify an integer in the range from 1 through 4096. Example: 1-1
externalPathGroupId	int	(Required) ID of the external path group

Attribute	Type	Description
		Specify a unique (unused) ID for the external path group. You can specify an integer in the range from `0` to `63231`.
portId	string	(Required) Port ID for connecting with the external system You cannot specify an iSCSI port.
externalPortWwn	string	(Required) WWN of the port of the externally connected storage system
lun	int	(Required) LUN assigned to the port of the externally connected storage system

Response message

Body

Attribute	Type	Description
statusResource	string	URL to be used to obtain the execution results of the newly-created external volumes

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/external-volumes
```

Changing the nickname of an external volume

You can specify the ID of an external volume and then change the nickname of the external volume.

Execution permission

Storage Administrator (Provisioning)

Request line

```
PATCH base-URL/simple/v1/objects/external-volumes/object-ID
```

Request message

Object ID

Specify the `id` value obtained by getting information about external volumes.

Attribute	Type	Description
id	int	(Required) External volume ID

Query parameters

None.

Body

```
{
  "nickname": "externalVolumeForQuorum2"
}
```

Attribute	Type	Description
nickname	string	(Optional) Specify a new nickname consisting of 1 through 32 characters. You can use alphanumeric characters (`0` through `9`, `A` through `Z`, `a` through `z`), space characters, and the following symbols: Comma (`,`), Hyphen (–), period (`.`), forward slash (`/`), colon (`:`), at sign (`@`), back slash (`\`), underscore (_) This attribute is case-sensitive. You can use a space character between usable characters, but cannot use it at the beginning or end of the nickname.

Attribute	Type	Description
		You cannot use a hyphen (–) at the beginning of the name.

Response message

Body

Attribute	Type	Type
affectedResources	string[]	List of URLs for referencing information about the changed external volume
operationDetails	object[]	Details about the changed External volume For details, see the description of the `operationDetails` attribute of the `commandStatus` object.

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X PATCH --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/external-volumes/3
```

Getting information about external parity groups

You can obtain information about external parity groups.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/external-parity-groups
```

Request message

Object ID

None.

Query parameters

With each request, you can obtain information about a maximum of 500 external parity groups. To obtain information about additional external parity groups, execute the API request multiple times by using a combination of the `count` and `startIndex` parameters. By specifying the `count` parameter, you can also filter the external parity group information you require.

Parameter	Type	Filter Condition
startIndex	int	(Optional) Specify the index number of the external parity group from which to start obtaining information. Use a value in the range from `0` through `65535`.
count	int	(Optional) Specify the number of external parity groups for which information is to be obtained. Use a value in the range from `1` through `500`. If this parameter is omitted, `500` is assumed.

Body

None.

Response message

Body

```
{
    "data": [
        {
            "id": "9-400",
            "status": "Normal",
            "externalPathGroupId": 400,
            "externalPath": [
                {
                    "portId": "CL5-A",
                    "portProtocol": "FC",
                    "externalPortWwn": "50060e8012000c60",
                    "lun": 5,
                    "status": "Normal"
                },
                {
                    "portId": "CL6-A",
                    "portProtocol": "FC",
                    "externalPortWwn": "50060e8012000c70",
                    "lun": 6,
                    "status": "Normal"
                }
            ],
```

```
            "index": 2050
        }
    ],
    "count": 1,
    "totalCount": 1,
    "hasNext": false
}
```


Attribute	Type	Description
id	string	External parity group ID
status	string	Status of the external parity group ▪ `Normal`: Normal ▪ `Checking`: Currently checking the status of the external path ▪ `Cache Destaging`: Currently writing data from the cache to the volume ▪ `Disconnect`: External paths are not connected ▪ `Blockade`: External paths are blocked ▪ `Warning`: A problem has occurred related to one or more external paths ▪ `Unknown`: Unknown
externalPathGroupId	int	External path group ID
externalPath	object[]	Information about each external path in the external parity group appears. ▪ `portId` (string) ID of the external connection port ▪ `portProtocol` (string) Protocol of the port • `FC` • `iSCSI` ▪ `externalPortWwn` (string) WWN of the port of the externally connected storage system If an iSCSI port is specified, this attribute will not appear.

Attribute	Type	Description
		▪ `lun` (int) LUN assigned to the port of the externally connected storage system If `Unknown` is displayed for the status of the external path, this attribute will not appear. ▪ `status` (string) Status of the external path • `Normal`: Normal • `Disconnect`: Not connected • `Temporary Blockade`: The port is blocked (temporarily blocked) • `Blockade`: Blocked • `Unknown`: Unknown
index	int	Index number of the external parity group

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/external-parity-groups
```

Getting information about a specific external parity group

You can specify the ID of an external parity group and obtain information about the external parity group.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/external-parity-groups/object-ID
```

Request message

Object ID

Specify the value of `id` that was obtained by using the API function for getting information about external parity groups.

Attribute	Type	Description
id	string	(Required) External parity group ID

Query parameters

None.

Body

None.

Response message

Body

The following is an example of output when information is obtained about the external parity group whose ID is `9-500`.

```
{
    "id": "9-500",
    "status": "Normal",
    "externalPathGroupId": 500,
    "externalPath": [
        {
            "portId": "CL5-A",
            "portProtocol": "FC",
            "externalPortWwn": "50060e8012000c60",
            "lun": 3,
            "status": "Normal"
        }
    ],
    "index": 2051
}
```

For details on attributes to be obtained, see the description of the API function for getting information about external parity groups.

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/external-parity-
groups/9-500
```

Getting information about external path groups

You can obtain a list of information about external path groups and about related external paths.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/external-path-groups
```

Request message

Object ID

None.

Query parameters

With each request, you can obtain information about a maximum of 500 external path groups. To obtain information about additional external path groups, execute the API request multiple times by using a combination of the `count` and `startId` parameters. By specifying the `count` parameter, you can also filter the external path group information you require.

Parameter	Type	Filter Condition
startId	int	(Optional) Specify the ID of the external path group from which to start obtaining information, by using a value in the range from `0` through `63231`.
count	int	(Optional) Specify the number of external path groups for which information is to be obtained, by using a value in the range from `1` through `500`. If this parameter is omitted, `500` is assumed.

Body

None.

Response message

Body

```
{
    "data": [
        {
            "id": 500,
            "externalPaths": [
                {
                    "portId": "CL5-A",
                    "portProtocol": "FC",
                    "externalPortWwn": "50060e8012000c60"
                }
            ]
        },
        {
            "id": 600,
            "externalPaths": [
                {
                    "portId": "CL5-A",
                    "portProtocol": "FC",
                    "externalPortWwn": "50060e8012000c60"
                },
                {
                    "portId": "CL6-A",
                    "portProtocol": "FC",
                    "externalPortWwn": "50060e8012000c70"
                }
            ]
        }
    ],
    "count": 2,
    "totalCount": 2,
    "hasNext": false
}
```

Attribute	Type	Description
id	int	External path group ID

Attribute	Type	Description
externalPaths	object[]	Information about each external path related to the external path group appears. ▪ `portId` (string) ID of the external connection port ▪ `portProtocol` (string) Protocol of the port • `FC` • `iSCSI` If port information cannot be obtained, this attribute will not appear. ▪ `externalPortWwn` (string) WWN of the port of the externally connected storage system If port information cannot be obtained or an iSCSI port is specified, this attribute will not appear.

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/external-path-groups
```

Getting information about a specific external path group

You can specify the ID of an external path group and obtain information about the specific external path group.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/external-path-groups/object-ID
```

Request message

Object ID

Specify the value of `id` that was obtained by using the API function for getting information about external path groups.

Attribute	Type	Description
id	int	(Required) External path group ID

Query parameters

None.

Body

None.

Response message

Body

The following is an example of output when information is obtained about the external path group whose ID is `600`.

```
{
    "id": 600,
    "externalPaths": [
        {
            "portId": "CL5-A",
            "portProtocol": "FC",
            "externalPortWwn": "50060e8012000c60"
        },
        {
            "portId": "CL6-A",
            "portProtocol": "FC",
            "externalPortWwn": "50060e8012000c70"
        }
    ]
}
```

For details on attributes to be obtained, see the description of the API function for getting information about external path groups.

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/external-path-groups/600
```

Getting information about a specific quorum disk

The following request gets information about a quorum disk by using the specified quorum disk ID.

Execution permission

Storage Administrator (View Only)

Request line

```
GET base-URL/simple/v1/objects/quorum-disks/object-ID
```

Request message

Object ID

Specify the `quorumId` value that was specified when information about the quorum disk was registered.

Attribute	Type	Description
quorumId	int	(Required) Quorum disk ID

Query parameters

None.

Body

None.

Response message

Body

The following is an example of obtaining information about the quorum disk whose quorum disk ID is `7`:

```
{
    "quorumId": 7,
    "externalVolumeId": 6400,
    "pairedStorageModel": "M8",
```

```
    "pairedStorageSerial": "401026"
}
```

Attribute	Type	Description
quorumId	int	Quorum disk ID
externalVolumeId	int	ID of the external volume for which information about the quorum disk is registered
pairedStorageModel	string	Model of the paired storage systems that share the quorum disk ▪ `R8`: VSP F1500, VSP G1000, or VSP G1500 ▪ `M8`: VSP Fx00 models or VSP Gx00 models If information about the model cannot be obtained, `Unknown` appears.
pairedStorageSerial	string	Serial number of the paired storage system that shares the quorum disk

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X GET https://
192.0.2.100/ConfigurationManager/simple/v1/objects/quorum-disks/7
```

Registering information about a quorum disk

In the storage system, you can register information about the quorum disk to be used for global-active device.

Execution permission

Storage Administrator (Provisioning)

Request line

```
POST base-URL/simple/v1/objects/quorum-disks
```

Request message

Object ID

None.

Query parameters

None.

Body

The following is an example of registering information about a quorum disk by specifying the following settings:

- Quorum disk ID: `7`
- External volume ID: `6400`
- Model of the paired storage system: `M8`
- Serial number of the paired storage system: `401026`

```
{
  "quorumId": 7,
  "externalVolumeId": 6400,
  "pairedStorageModel": "M8",
  "pairedStorageSerial": "401026"
}
```

Attribute	Type	Description
quorumId	int	(Required) Quorum disk ID Specify an integer in the range from `0` through `31` as the ID of the quorum disk to be registered. Specify an unused quorum disk ID.
externalVolumeId	int	(Required) External volume ID Specify an integer in the range from `0` through `65279` as the ID of the external volume to be registered.
pairedStorageModel	string	(Required) Model of the paired storage system that are to share the quorum disk You can specify any of the following values: ▪ `R8`: VSP F1500, VSP G1000, or VSP G1500 ▪ `M8`: VSP Fx00 models or VSP Gx00 models

Attribute	Type	Description
pairedStorageSerial	string	(Required) Serial number of the paired storage system that are to share the quorum disk

Response message

Body

Attribute	Type	Description
statusResource	string	URL to be used to obtain the execution results of the registered quorum disk

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97) .

Status codes

See HTTP status codes (on page 85) .

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X POST --data-
binary @./InputParameters.json https://192.0.2.100/ConfigurationManager/
simple/v1/objects/quorum-disks
```

Deleting information about a quorum disk

You can delete information about a quorum disk from the storage system by specifying the Quorum disk ID.

Note:

To also delete external volumes, you must execute the API request that unmapping an external volume before you delete those volumes. For details about how to perform this operation, see the *REST API Reference Guide*.

Execution permission

Storage Administrator (Provisioning)

Request line

```
DELETE base-URL/simple/v1/objects/quorum-disks/object-ID
```

Request message

Object ID

Specify the `id` value obtained by getting information about quorum disks.

Attribute	Type	Description
quorumId	int	(Required) Quorum disk ID

Query parameters

None.

Body

None.

Response message

Body

Attribute	Type	Description
statusResource	string	URL to be used to obtain the execution results of deleted quorum disk information

Note:

Execute the API function for obtaining information about the status of the API function that performs asynchronous processing. For details, see Getting status information about an API function that performs asynchronous processing (on page 97).

Status codes

See HTTP status codes (on page 85).

Coding example

```
curl -v -H "Accept:application/json" -H "Content-Type:application/json" -H
"Authorization:Session b74777a3f9f04ea8bd8f09847fac48d3" -X DELETE
https://192.0.2.100/ConfigurationManager/simple/v1/objects/quorum-disks/7
```

Index

H

L

M

N

O

P

Q

R

S

U

V

W

Hitachi Vantara Corporation

Corporate Headquarters
2845 Lafayette Street
Santa Clara, CA 95050-2639 USA
www.HitachiVantara.com | community.HitachiVantara.com

Regional Contact Information
Americas: +1 866 374 5822 or info@hitachivantara.com
Europe, Middle East, and Africa: +44 (0) 1753 618000 or info@emea@hitachivantara.com
Asia Pacific: + 852 3189 7900 or info.marketing.apac@hitachivantara.com