IBM Hyper-Scale Manager
Version 5.5.2

REST API Specifications

SC27-6440-07

Note

Before using this information and the product it supports, read the information in “Notices” on page 99.

Edition Notice

Publication number: SC27-6440-07. This edition applies to IBM® Hyper-Scale Manager version 5.5.2 and to all subsequent releases and modifications, until otherwise indicated in a newer publication.

Contents

Tables

About this guide

The purpose of this guide is to provide the specifications of the IBM Hyper-Scale Manager REST application programming interface (API).

Who should use this guide

This guide is for software developers who are coding control applications for XIV, Spectrum Accelerate, FlashSystem A9000, and FlashSystem A9000R systems.

Conventions used in this guide

These notices are used to highlight key information.

Note: These notices provide important tips, guidance, or advice.

Important: These notices provide information or advice that might help you avoid inconvenient or difficult situations.

Attention: These notices indicate possible damage to programs, systems, or data. An attention notice appears before the instruction or situation in which damage can occur.

Related information and publications

Additional information and publications related to IBM Hyper-Scale Manager can be found on the following information sources.

- IBM FlashSystem® on IBM Knowledge Center (http://www.ibm.com/support/knowledgecenter/SSUMNQ)
- IBM Flash Storage and Solutions marketing website (ibm.com®/systems/storage/flash)
- IBM FlashSystem A9000 on IBM Knowledge Center (http://www.ibm.com/support/knowledgecenter/STJKMM)
- IBM FlashSystem A9000R on IBM Knowledge Center (http://www.ibm.com/support/knowledgecenter/STJKN5)
- IBM XIV® Storage System marketing website (ibm.com/systems/storage/disk/xiv)
- IBM XIV Storage System on IBM Knowledge Center (ibm.com/support/knowledgecenter/STJTAG)
- IBM Spectrum Accelerate marketing website (ibm.com/systems/storage/spectrum/accelerate)
- IBM Spectrum Accelerate on IBM Knowledge Center (ibm.com/support/knowledgecenter/STZSWD)
- IBM Knowledge Center (ibm.com/support/knowledgecenter)
- IBM Storage Redbooks® website (redbooks.ibm.com/portals/storage)

IBM Hyper-Scale Manager documentation set

IBM Hyper-Scale Manager 5.x currently has the following documentation set.

IBM Hyper-Scale Manager publications	Description
Release Notes	Describes what's new, requirements, compatibility, change log, and known issues information in the latest version.
User Guide	Describes how to install, configure, and operate IBM Hyper-Scale Manager.
Quick-Start Guide	Walks you through installing the IBM Hyper-Scale Manager software and initial configuration of the GUI.
REST API Specifications	Provides specifications of the IBM Hyper-Scale Manager Representational State Transfer (REST) application programming interface (API).

Getting information, help, and service

If you need help, service, technical assistance, or want more information about IBM products, you can find various sources to assist you. You can view the following websites to get information about IBM products and services and to find the latest technical information and support.

- IBM website (ibm.com)
- IBM Support Portal website (ibm.com/support/entry/portal/support?brandind=Hardware~System_Storage)
- IBM Directory of Worldwide Contacts website (ibm.com/planetwide)

  Use the Directory of Worldwide Contacts to find the appropriate phone number for initiating voice call support. Select the Software option, when using voice response system.

  When asked, provide your Internal Customer Number (ICN) and/or the serial number of the storage system that requires support. Your call will then be routed to the relevant support team, to whom you can provide the specifics of your problem.

IBM Publications Center

The IBM Publications Center is a worldwide central repository for IBM product publications and marketing material.

The IBM Publications Center website (ibm.com/shop/publications/order) offers customized search functions to help you find the publications that you need. You can view or download publications at no charge.

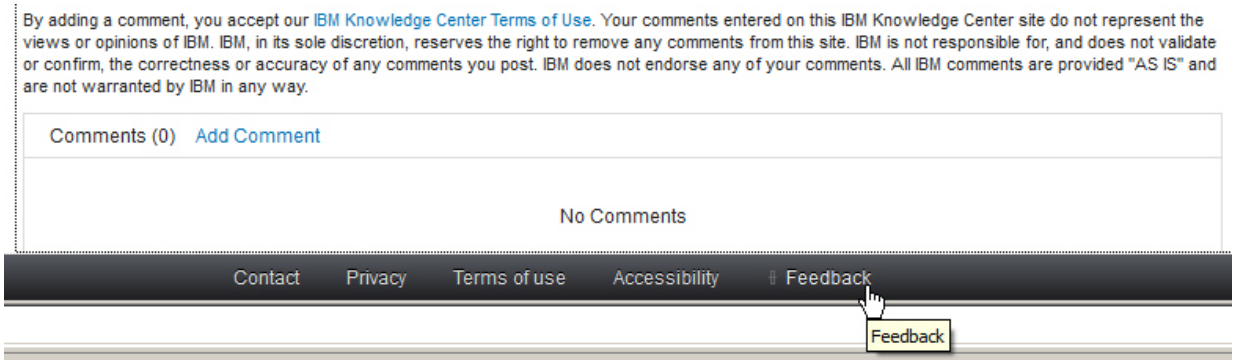
Sending or posting your comments

Your feedback is important in helping to provide the most accurate and highest quality information.

Procedure

To submit any comments about this guide:

- Go to IBM Knowledge Center (ibm.com/support/knowledgecenter), drill down to the relevant page, and then click the **Feedback** link that is located at the bottom of the page.



The feedback form is displayed and you can use it to enter and submit your comments privately.

- You can post a public comment on the IBM Knowledge Center page that you are viewing, by clicking **Add Comment**. For this option, you must first log in to IBM Knowledge Center with your IBM ID.
- You can send your comments by email to starpubs@us.ibm.com. Be sure to include the following information:
  - Exact publication title and product version
  - Publication form number (for example: SC01-0001-01)
  - Page, table, or illustration numbers that you are commenting on
  - A detailed description of any information that should be changed

---

Note: When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Chapter 1. Introduction

The Hyper-Scale Manager REST API allows the end user to process two main types of requests:

- **Query requests:** Requests that return states, for either a single resource or a list of resources. These are object entities managed by Hyper-Scale Manager (for example, all volumes of a specific supported storage system).
- **Update requests:** Requests that update states – that create, delete, or update the object entity state (for example, one that creates a new pool on a specific supported storage system).

The API uses HTTP as the transport protocol and relies on HTTP for some features, such as security.

There are two resource types that requests may call:

- **Single resource:** Volume, port, or host.
- **List of resources:** All pools of a specific supported storage system.

Protocol version

The current protocol version is v5. This is indicated in the URL by adding the `/v5` path to all of the URLs and all references to `xiv` have been updated to `hsm` (for example, `/hsm/v5/volumes`).

The current protocol version (v5) supports IBM FlashSystem A9000 and A9000R, IBM XIV Gen3, and IBM Spectrum Accelerate storage systems.

An example response format is displayed for both IBM XIV Gen3 and IBM FlashSystem A9000 and A9000R storage systems.

IBM REST API protocol versions **v1** and **v2** support IBM XIV Gen3 storage systems.

Important: The v3 protocol version has been deprecated and should no longer be used.

HTTP methods

The protocol relies on HTTP methods to support CRUD operations (Create/Read/Update/Delete):

- `HTTP POST`: Mapped to create/update/delete operations.
- `HTTP GET`: Mapped to query (read) operations.
- `HTTP DELETE`: Mapped to delete operations. The HTTP URL specifies the resource to which the request applies.

URL scope specifiers and resource names

Request URLs are built from scope specifiers and resource names, prefixed with `:`, for example:

```
/hsm/v5/:hsm2/ports/:10000000C9926DCA
```

In the above example:

- `/hsm` specifies the main API scope.
- v5 specifies the protocol version. (Should be `v1` for version one.)
- `:hsm2` specifies the system IP/DNS name.
- `/ports` defines the ports scope. For example, we are looking at ports in `system1`.
- `10000000C9926DCA` is the port name.

Note: The request and response format is JSON. All of the examples shown in the document are formatted as JSON.

Chapter 2. General requests

The following general query requests are supported.

Generate a capacity report for all of the systems in your inventory:

```
GET /hsm/v5/systems/capacityReport
```

Generate a capacity report for a specific system in your inventory:

```
GET /hsm/v5/systems/:<system-name>/capacityReport
```

A download bar appears at the bottom of your screen with a link to the containing folder of the zip file. Open the zip folder and click on the PDF to display the report.

Chapter 3. Query requests

A query is run by issuing the HTTP GET command on a URL, representing either:

- A collection of resources where a list of resources and their properties is returned (for example, `/hsm/v5/:system1/volumes`).
- A single resource where single resource properties are returned (for example, `/hsm/v5/:system1/volumes/:vol1`).

Query requests may contain additional URL parameters, for filtering. Requests that use filtering parameters return only resources that match the filtering criteria. For example, `/hsm/v5/:system1/volumes?pool=mypool` would return only volumes that exist in the **mypool** pool.

All queries listed in this document (excluding system queries), which return collections of resources, will support "by domain" filtering. For example: `/hsm/v5/:system1/volumes?domain=d1` would return all volumes on a given system (**system1**) that belong to the specified domain (**d1**).This functionality is optional and will be implemented on a"best effort basis."

The following topics are covered in this chapter:

Query paging support

A query request can return a huge list of resources. For example, a request for all volumes from all supported storage systems can contain tens of thousands of volume resources. A client may only be interested in requesting a subset/page from the query. The following two parameters support this requirement:

- **offset**: Defines the index of the first resource that should be returned in the query. (All resources are numbered, starting from 0.) If an **offset** parameter is omitted, an offset of 0 is assumed. If the offset is larger than the number of resources, a response with no resources is returned.

- **limit**: Defines the maximum count of resources that should be returned in the query. The actual count of resources in the response is less than or equal to the requested limit. If the **limit** parameter is omitted, the server returns up to `max_limit` amount of resources (defaults to 1000). If the limit parameter is higher than the amount of resources, the request will fail with a `400 Bad Request` error.

By default, the list of returned objects is sorted by name.

Note: System and event queries do not support paging.

Examples:

- `/hsm/v5/volumes`: Returns all of the volumes from all of the XIV systems (up to `max_limit`).
- `/hsm/v5/volumes?limit=400`: Returns the first 400 volumes from all of the XIV systems.
- `/hsm/v5/volumes?offset=10&limit=40`: Returns 40 volumes, starting from volume 11.
- `/hsm/v5/volumes?offset=101`: Returns all volumes (up to max_limit), starting from volume 102.

Query response format

The query response is defined as an object having the following fields:

- **status**: The response status.
  - **server**: Application level status and status message.
  - **failed_systems**: Array of machine errors that were encountered during the request. This field is used during multi-machine queries or updates to indicate that some machines had issues (such as `disconnected`, `not–authorized`, `invalid state`, etc.) during the query and that the returned data may not contain all resources, or the required operation had only partial success.
- **response**: The query response result.
  - **counts**: The response count and total count of objects.
    - **data_count**: The number of objects in this response data.
    - **total_count**: The number of total objects.
  - **data**: The response data. Can be either:
    - A list of objects for queries on object collections (for example: `/hsm/volumes`).
    - A single object for queries on a specific object (for example: `/hsm/:system1/volumes/:vol1`).
    - Null, if there was some error in query processing.

Notes regarding format of query response details:

- All of the storage capacity properties in the query responses are given as integers, in bytes. For example: capacity attributes of system resources, size attributes of volume resources, size attributes of pool resources, etc.
- Boolean attributes are given as `true`/`false`, such as for the mirrored attribute of volume resources, etc.
- Time stamps are displayed in milliseconds, since the Unix epoch (example: `1970-01-01 00:00:00`).

Example of a successful response:

```
Request - GET /hsm/v5/:system1/volumes/:vol1
…
Response - HTTP/1.1 200 OK
…
{
"status": {
                    "server": {"status": "0", "message": "OK" }
                    "failed_systems": []
           }
"response": {
"counts": { "data_count": "1", "total_count": "1"}
"data": {
                    "volume": {
                                        "name": "vol1"
                                         …
                                }
             }
             }
}
```

Example of a partially successful response:

```
Request - GET /hsm/v5/volumes
…
Response - HTTP/1.1 200 OK
…
{
"status": {
"server": {"status": "0", "message": "OK" }
"failed_systems": [ {"name": "mn44", "status": "1", "message": "The system is disconnected"}]
           }
"response": {
"counts": { "data_count": "10", "total_count": "1000"}
"data": {
                        "volume": [ {
                                "name": "vol1"
                                …
                        }
                        …
                        {
                                "name": "vol10"
                                 …
                   } ]
                }
                }
}
```

Example of a response to an invalid request:

```
Request - GET /hsm/v5/myvolumes
… Response -
HTTP/1.1 400 Bad Request
…
{
"status": {
     "server": {"status": "6", "message": "Invalid request – myvolumes path is undefined" }
     "failed_systems": []
          }
"response": {
     "counts": { "data_count": "0", "total_count": "0"}
     "data": {}
          }
}
```

System query requests

The following system query requests are supported:

List all of the systems on which the user is authorized:

```
GET /hsm/v5/systems
```

List specific system properties (system1 – system IP/DNS):

```
GET /hsm/v5/systems/:system1
```

System Fiber Channel (FC) port query requests

The following system Fiber Channel (FC) port query requests are supported:

List all of the FC ports from a specific machine:

```
GET /hsm/v5/:system1/fc_ports
```

List specific FC port properties: (where p1 is the port name)

```
GET /hsm/v5/:system1/fc_ports/:p1
```

System iSCSI port query requests

The following system iSCSI port query requests are supported:

List all of the system iSCSI ports from a specific machine:

```
GET /hsm/v5/:system1/iscsi_ports
```

List specific iSCSI port properties: (where p1 is the port name)

```
GET /hsm/v5/:system1/iscsi_ports/:p1
```

Pool query requests

The following pool query requests are supported:

List all of the pools from all of the systems on which the user is authorized:

```
GET /hsm/v5/pools
```

List all of the pools from a specific machine:

```
GET /hsm/v5/:system1/pools
```

List specific pool properties (where p1 is the pool name):

```
GET /hsm/v5/:system1/pools/:p1
```

List all of the pools from a specific machine that belong to a specific domain (where d1 is the domain name):

```
GET /hsm/v5/:system1/pools?domain=d1
```

List all of the pools from a specific machine that belong to a performance class (where q1 is the performance class name):

```
GET /hsm/v5/:system1/pools?perf_class=q1
```

Volume query requests

The following volume query requests are supported:

List all of the volumes from all of the systems on which the user is authorized:

```
GET /hsm/v5/volumes
```

List all of the volumes from specific machine:

```
GET /hsm/v5/:system1/volumes
```

List all of the volumes from specific machine and pool (where p1 is the pool name):

```
GET /hsm/v5/:system1/volumes?pool=p1
```

List all of the volumes from specific machine and domain (where d1 is the domain name):

```
GET /hsm/v5/:system1/volumes?domain=d1
```

List all of the volumes from specific machine and performance class on FlashSystems A9000 and A9000R systems only (where q1 is the performance class name):

```
GET /hsm/v5/:system1/volumes?perf_class=q1
```

List specific volume properties:

```
GET /hsm/v5/:system1/volumes/:vol1
```

Volume snapshot query requests

The following volume snapshot query requests are supported:

List all of the snapshots from all of the systems on which the user is authorized:

```
GET /hsm/v5/snapshots
```

List all of the snapshots from a specific machine:

```
GET /hsm/v5/:system1/snapshots
```

List all of the snapshots of the specified volume (where vol1 is the volume name):

```
GET /hsm/v5/:system1/snapshots?volume=vol1
```

List all of the snapshots associated with the specified domain (where dom1 is the domain name):

```
GET /hsm/v5/:system1/snapshots?domain=dom1
```

List specific snapshot properties (where sn1 is the snapshot name):

```
GET /hsm/v5/:system1/snapshots/:sn1
```

Consistency group query requests

The following consistency group query requests are supported:

List all of the consistency groups from all of the systems on which the user is authorized:

```
GET /hsm/v4/cgs
```

List all of the consistency groups from a specific machine:

```
GET /hsm/v4/:system1/cgs
```

List all of the consistency groups associated with the specified domain (where dom1 is the domain name):

```
GET /hsm/v4/:system1/cgs?domain=dom1
```

List specific consistency group properties (where cg1 is the consistency group name):

```
GET /hsm/v4/:system1/cgs/:cg1
```

Snapshot group query requests

The following snapshot group query requests are supported:

List all of the snapshot groups from all of the systems on which the user is authorized:

```
GET /hsm/v4/snap_groups
```

List all of the snapshot groups from a specific machine:

```
GET /hsm/v4/:system1/snap_groups
```

List all of the snapshot groups associated with the specified domain (where dom1 is the domain name):

```
GET /hsm/v4/:system1/snap_groups?domain=dom1
```

List all of the snapshot groups of the specified consistency group (where cg1 is the consistency group name):

```
GET /hsm/v4/:system1/snap_groups?cg=c1
```

List specific snapshot group properties (where sg1 is the snapshot group name):

```
GET /hsm/v4/:system1/snap_groups/:sg1
```

Mapping query requests

The following host, cluster, and volume mapping query requests are supported:

List all of the mappings of the host (where h1 is the host name):

```
GET /hsm/v5/:system1/vol_maps?host=h1
```

List all of the mappings of the cluster (where c1 is the cluster name):

```
GET /hsm/v5/:system1/vol_maps?cluster=c1
```

List all of the mappings of the volume (where `vol1` is the volume name):

```
GET /hsm/v5/:system1/vol_maps?volume=vol1
```

List specific host mapping properties (where h1 is the host name, and vol1 is the volume name):

```
GET /hsm/v5/:system1/vol_maps/:host:h1:vol1
```

List specific cluster mapping properties (where `c1` is the host name, and `vol1` is the volume name):

```
GET /hsm/v5/:system1/vol_maps/:cluster:c1:vol1
```

Mirror query requests

The following mirror query requests are supported:

List all of the mirrors from all of the systems on which the user is authorized:

```
GET /hsm/v4/mirrors
```

List all of the mirrors from a specific machine:

```
GET /hsm/v4/:system1/mirrors
```

List all of the mirrors of volumes and consistency groups associated with the specified domain (where dom1 is the domain name):

```
GET /hsm/v4/:system1/mirrors?domain=dom1
```

List the mirror properties of a specific volume mirror (where `vol1` is the volume name):

```
GET /hsm/v4/:system1/mirrors/:volume:vol1
```

List the mirror properties of a specific consistency group mirror (where `cg1` is the source consistency group name):

```
GET /hsm/v4/:system1/mirrors/:cg:cg1
```

Host query requests

The following host query requests are supported:

List all of the hosts from all of the systems on which the user is authorized:

```
GET /hsm/v5/hosts
```

List all of the hosts from a specific machine:

```
GET /hsm/v5/:system1/hosts
```

List all of the hosts from a specific machine that are associated with a specific domain (where dom1 is the domain name):

```
GET /hsm/v5/:system1/hosts?domain=dom1
```

List all of the hosts from a specific machine that belong to a specific cluster (where `cl1` is the cluster name):

```
GET /hsm/v5/:system1/hosts?cluster=cl1
```

List all of the hosts from a specific machine that belong to a specific performance class (where `q1` is the performance class name):

```
GET /hsm/v5/:system1/hosts?perf_class=q1
```

List specific host properties: (where `h1` is the host name)

```
GET /hsm/v5/:system1/hosts/:h1
```

Host port query requests

Host ports are ports that belong to hosts, not to systems. The following system host port query requests are supported:

List all of the ports of a specific host (where `h1` is the host name):

```
GET /hsm/v5/:system1/host_ports?host=h1
```

List the host to which the specific port is assigned (where `p1` is the port name):

```
GET /hsm/v5/:system1/host_ports?port=p1
```

List the specific FC port on a host (where `h1` is the host name, and `p1` is the port name):

```
GET /hsm/v5/:system1/host_ports/:fc:h1:p1
```

List the specific iSCSI port on a host (where `h1` is the host name, and `p1` is the port name):

```
GET /hsm/v5/:system1/host_ports/:iscsi:h1:p1
```

Cluster query requests

The following cluster query requests are supported:

List all of the clusters from all of the systems on which the user is authorized:

```
GET /hsm/v5/clusters
```

List all of the clusters from a specific machine:

```
GET /hsm/v5/:system1/clusters
```

List specific cluster properties: (where `c1` is the cluster name)

```
GET /hsm/v5/:system1/clusters/:c1
```

List all of the clusters from a specific machine that are associated to a specific domain (where `dom1` is the domain name):

```
GET /hsm/v5/:system1/clusters?domain=dom1
```

Performance class query requests

The following performance class query requests are supported:

List all of the performance classes from all of the systems on which the user is authorized:

```
GET /hsm/v5/perf_classes
```

List all of the performance classes from a specific machine:

```
GET /hsm/v5/:system1/perf_classes
```

List the properties of a specific performance class (where `silver` is the performance class name):

```
GET /hsm/v5/perf_classes/:silver
```

List all volumes that are a part of a specific performance class:

```
GET /hsm/v5/:system1/volumes?perf_class=silver
```

List all hosts that are part of a specific performance class

```
GET /hsm/v5/:system1/hosts?perf_class=silver
```

List all pools that are part of a specific performance class

```
GET /hsm/v5/:system1/pools?perf_class=silver
```

Data migration query requests

The following data migration query requests are supported:

List data migration for all of the systems on which the user is authorized:

```
GET /hsm/v5/data_migrations
```

List data migration for a specific machine:

```
GET /hsm/v5/:system1/data_migrations
```

List data migration for a specific volume on a specific machine (where v1 is the volume name):

```
GET /hsm/v5/:system1/data_migrations/:v1
```

List data migration for all of the machines that belong to a specific domain (where d1 is the domain name):

```
GET /hsm/v5/:system1/data_migrations?domain=d1
```

Domain query requests

The following domain query requests are supported:

List all of the domains from all of the systems on which the user is authorized:

```
GET /hsm/v5/domains
```

List all of the domains from a specific machine:

```
GET /hsm/v5/:system1/domains
```

List specific domain properties (where `dom1` is the domain name):

```
GET /hsm/v5/:system1/domains/:dom1
```

List all of the domains from a specific machine that belong to a performance class (where `q1` is the performance class name):

```
GET /hsm/v5/:system1/domains?perf_class=q1
```

Ethernet port query requests

The following Ethernet port query requests are supported:

List all of the Ethernet ports from all of the systems on which the user is authorized:

```
GET /hsm/v5/ethernet_ports
```

List all of the Ethernet ports from a specific machine:

```
GET /hsm/v5/:system1/ethernet_ports
```

List specific Ethernet port properties (where `eth1` is the port name):

```
GET /hsm/v5/:system1/ethernet_ports/:eth1
```

Host connectivity query requests

The following host connectivity query requests are supported:

List all of the host connectivities to all of the systems on which the user is authorized:

```
GET /hsm/v5/host_connectivity
```

List connectivity of all of the hosts to a specific machine:

```
GET /hsm/v5/:system1/host_connectivity
```

List connectivity properties for a host (where `host1` is the host name):

```
GET /hsm/v5/:system1/host_connectivity?host=host1
```

IP interface query requests

The following IP interface query requests are supported:

List all of the IP interfaces from all of the systems on which the user is authorized:

```
GET /hsm/v5/ip_interfaces
```

List all of the IP interfaces from a specific machine:

```
GET /hsm/v5/:system1/ip_interfaces
```

List specific IP interface properties (where `1.1` is the IP interface name):

```
GET /hsm/v5/:system1/ip_interfaces/:1.1
```

User query requests

The following user query requests are supported:

List all of the users from all of the systems on which the user is authorized:

```
GET /hsm/v5/users
```

List all of the users for a specific machine:

```
GET /hsm/v5/:system1/users
```

List specific user properties (where `user1` is the user name):

```
GET /hsm/v5/:system1/users/:user1
```

List all users for all of the machines that belong to a specific domain (where d1 is the domain name):

```
GET /hsm/v5/:system1/users?domain=d1
```

Event query requests

An event query request fetches events from a specific supported storage system. Event queries are different from the rest of the queries since events are currently not managed by the Hyper-Scale Manager.

The following event query request is supported.

List specific event properties: (where `100` is the event index)

```
GET /hsm/v5/:system1/events/:100
```

Since events are not managed by the Hyper-Scale Manager, paging is not supported for this query. The following query parameters can be used to filter the events:

- **min_severity**: Defines the minimum event severity. Possible values include `Informational`, `Minor`, `Major`, `Warning`, or `Critical`. If omitted, the minimum severity defaults to `Informational`.
- **after**: Only returns events that occur after the specified timestamp.
- **before**: Only returns events that occur before the specified timestamp.

The syntax for the before and after fields is as follows: `Y-M-D[.[h[:m[:s]]]]`, where the ranges are as follows:

- `Y`: year (four digits)
- `M`: month (01-12)
- `D`: day (01-31)
- `h`: hour (00-23, with 00 as the default)
- `m`: minute (00-59 with 00 as the default)
- `s`: second (00-59 with 00 as the default)

If filtering parameters are omitted, the last 300 events are returned.

Example event query:

```
GET /hsm/v5/:system1/events?min_severity=Warning&after=2013-03-07
```

Example event query response:

```
HTTP/1.1 200 OK
… {
"status": {
      "server": {"status": "0", "message": "OK" }
      "failed_systems": []
            }
"response": {
"counts": { "data_count": "300", "total_count": "300"}
"data": {
                              ….
                }
}
```

Note: Since events are not managed by the server and paging is not supported, the data count returned in response is always equal to the total count.

Chapter 4. Update requests

Update requests include creation, deletion, or modification of managed resources. Since the Hyper-Scale Manager only reflects the state of objects managed by the supported storage systems, all such requests should be passed by the Hyper-Scale Manager to the supported storage systems to handle, using CLI commands internally. Here a new generic protocol format is introduced, decoupled from the CLI syntax.

Since update requests require some data to be provided by clients (for example, a new volume size in the volume resize operation), how this data is transferred must be defined. One alternative is to provide this data as HTTP URL parameters; however, this means that the protocol is strictly coupled with the transport protocol (HTTP in this case) and cannot be reused easily with other transport protocols that do not support the URL parameters feature. The only other option is to deliver the required data inside an HTTP body in JSON format.

Define the update request object with the following generic fields:

- **action**: The required action. Possible values include `create` and `delete`. Additional actions are possible, based on the resource type.
- **params**: Contains the parameters required for the action.

Updates are done using the HTTP `POST` request on a URL representing a collection of resources (for example, `POST` on `/hsm/v5/:system1/volumes`), while the body contains the exact action to perform (`create`/`delete`/`update`). Each such request can carry a list of update requests to support bulk operations. Additionally, the following shortcuts are allowed:

- HTTP `DELETE` request on a full URL (for example, `/hsm/v5/:system1/volumes/:vol1`) with no body, in order to delete it.
- HTTP `POST` request on a full URL in order to update/delete it. The body contains a single request object with all of the required parameters, except **name** (since it appears in URL).

Note: This release does not support multiple requests in the body. Only a single update request in the body is supported.

The following topics are covered in this chapter:

Update response format

The update response is defined as an object that has a status field only, as defined in the query response format.

See Query response format.

Example of a successful response to an update request:

```
Request - GET /hsm/v5/:system1/volumes/:vol1
…
Response - HTTP/1.1 200 OK
…
{
"status": {
                    "server": {"status": "0", "message": "OK" }
                    "failed_systems": []
           }
}
```

Example of a failed request:

```
Request - DELETE/hsm/v5/:system1/volumes/:vol1
…
Response - HTTP/1.1 500 Internal Server Error
…
{
"status": {
                    "server": {"status": "5", "message": "Update request
                              failed with reason: Volume 'vol1' does not exist" }
                    "failed_systems": []
           }
}
```

Pool update requests

The following types of pool update requests are supported.

- "Pool creation requests" on page 18
- "Pool resize requests" on page 19
- "Pool resize snapshot size requests" on page 20
- "Pool rename requests" on page 20
- "Pool deletion requests" on page 21

Pool creation requests

Pool creation requests include the following components:

- An **action** parameter with a value of `create`.
- A **params** parameter, containing the following parameters:
  - **name**: The pool name.
  - (In IBM XIV Gen3 storage systems) **hard_size**: The total physical size of the storage pool, in GB.
  - (In IBM XIV Gen3 storage systems) **soft_size**: The virtual size of the storage pool, in GB.
  - (In IBM FlashSystems A9000 and A9000R storage systems) **size**: The size of the storage pool, in GB.
  - **snap_size**: The size, in GB, reserved for snapshots.
  - **domain** (optional): The name of the domain to associate. If this parameter is omitted, then the pool will not be associated with any domain.

Example:

The output for IBM XIV Gen3 storage systems is similar to the following:

```
POST /hsm/v5/:system1/pools
{
  "request" : [
              {
                "action" : "create",
                "params" :
                      {
                        "name" : "itay_demo_pool_02",
                        "soft_size" : "222",
                        "hard_size" : "111",
                        "snap_size" : "1",
                        "lock_behavior" : "no_io",
                        "compressed" : "false"
                      }
              }
            ]
}
```

The output for IBM FlashSystem A9000 and A9000R storage systems is similar to the following:

```
POST /hsm/v5/:system1/pools
{
  "request" : [
        {
          "action" : "create",
          "params" :
                {
                  "name" : "itay_demo_pool_03",
                  "size" : "222",
                  "snap_size" : "1"
                }
        }
  ]
}
```

Pool resize requests

Pool resize requests include the following components:

- An **action** parameter with a value of `resize_pool_size`.
- A **params** parameter, containing the following parameters:
  - **name**: The pool name.
  - (In IBM XIV Gen3 storage systems) **hard_size**: The total physical size of the storage pool, in GB.
  - (In IBM XIV Gen3 storage systems) **soft_size**: The virtual size of the storage pool, in GB.
  - (In IBM FlashSystems A9000 and A9000R storage systems) **size**: The size of the storage pool, in GB.
  - **size**: Virtual capacity of the storage pool, in GB.

Example:

The output for IBM XIV Gen3 storage systems is similar to the following:

```
POST /hsm/v5/:system1/pools
{
  "request" : [
              {
              "action": "resize_pool_size",
              "params": {
                "name": "EZPool",
                "hard_size": "1500",
                "soft_size" : "2000"
                }
              }
            ]
}
```

The output for IBM FlashSystems A9000 and A9000R storage systems is similar to the following:

```
POST /hsm/v5/:system1/pools
{
  "request" : [
              {
              "action": "resize_pool_size",
              "params": {
                "name": "alex_test",
                "size": "1000"
                }
              }
            ]
}
```

Alternatively, a pool can be resized using the following request:

```
POST /hsm/v5/:system1/pools/:p1
{
   "request": {
                  "action": "resize_pool_size",
                  "params": {
                                "size": "1700"
                            }
              }
}
```

Pool resize snapshot size requests

Pool resize snapshot size requests include the following components:

- An **action** parameter with a value of `resize_snapshot_size`.
- A **params** parameter, containing the following parameters:
  - **name**: The pool name.
  - **snap_size**: The new size, in GB, reserved for snapshots.

Example:

```
POST /hsm/v5/:system1/pools
{
   "request": [
              {
                  "action": "resize_snapshot_size",
                  "params": {
                                "name": "p1",
                                "snap_size": "170"
                            }
              }
   ]
}
```

Alternatively, the snapshot size can be resized using the following request:

```
POST /hsm/v5/:system1/pools/:p1
{
   "request": {
                  "action": "resize_snapshot_size",
                  "params": {
                                "snap_size": "170"
                            }
              }
}
```

Pool rename requests

Pool rename requests include the following components:

- An **action** parameter with a value of `rename`.
- A **params** parameter, containing the following parameters:
  - **name**: The pool name.

- **new_name**: The new name of the pool.

Example:

```
POST /hsm/v5/:system1/pools
{
   "request": [
               {
                   "action": "rename",
                   "params": {
                                       "name": "pool1",
                                       "new_name": "pool2"
                              }
               }
   ]
}
```

Alternatively, a pool can be renamed using the following request:

```
POST /hsm/v5/:system1/pools/:pool1
{
   "request": {
                   "action": "rename",
                   "params": {
                                       "new_name": "pool2"
                              }
               }
}
```

Pool deletion requests

Pool deletion requests include the following components:

- An **action** parameter with a value of `delete`.
- A **params** parameter, containing the following parameter:
  - **name**: The pool name.

Example:

```
POST /hsm/v5/:system1/pools
{
   "request": [
               {
                   "action": "delete",
                   "params": {
                                       "name": "pool1"
                              }
               }
   ]
}
```

Alternatively, a pool can be deleted using either of the the following requests, where p1 is the pool name:

```
POST /hsm/v5/:system1/pools/:p1
{
   "request": [
               {
                   "action": "delete"
               }
               ]
}
```

```
DELETE /hsm/v5/:system1/pools/:p1
```

Volume update requests

The following types of volume update requests are supported.

- Volume creation requests
- Volume resize requests
- Volume lock requests
- Volume unlock requests
- Volume rename requests
- Volume copy requests
- Volume move to a pool
- Volume delete requests
- “Volume restore from snapshot requests” on page 27

Volume creation requests

Volume creation requests include the following components:

- An **action** parameter with a value of `create`.
- A **params** parameter, containing the following parameters:
  - **name**: The volume name.
  - **pool**: The pool name where volume should be created.
  - **size**: The volume size.
  - **size_units**: The size units. Possible values include GB and BLOCKS.
  - **compressed**: The volume can be created as compressed in IBM XIV Gen3 storage systems that support compression. Possible values are `true` or `false`.

Example:

The output for IBM XIV Gen3 storage systems is similar to the following:

```
POST /hsm/v5/:system1/volumes
{
"request" : [
              {
              "action" : "create",
              "params" :
                      { "name" : "demo_volume_03",
                        "pool" : "pool_1",
                        "size": "17",
                        "size_units" : "GB" ,
                        "compressed" : "false"
                      }

              }
            ]
}
```

The output for IBM FlashSystems A9000 and A9000R storage systems is similar to the following:

```
POST /hsm/v5/:system1/volumes
{
"request" : [
              {
              "action" : "create",
              "params" :
                      { "name" : "itay_demo_volume_04",
                        "pool" : "igorg",
                        "size": "17",
                        "size_units" : "GB"
                      }

              }
            ]
}
```

Volume resize requests

Volume resize requests (to increase or decrease the volume size) include the following components:

- An **action** parameter with a value of `resize`.
- A **params** parameter, containing the following parameters:
  - **name**: The volume name.
  - **size**: The new volume size.
  - **size_units**: The size units. Possible values include GB and BLOCK.

Example:

```
POST /hsm/v5/:system1/volumes
{
   "request": [
               {
                   "action": "resize",
                   "params": {
                                   "name": "vol1",
                                   "size": "170",
                                   "size_units": "GB"
                             }
               }
   ]
}
```

Alternatively, a volume can be resized with the following request:

```
POST /hsm/v5/:system1/volumes/:vol1
{
   "request": [
               {
                   "action": "resize",
                   "params": {
                                   "size": "170",
                                   "size_units": "GB"
                             }
               }
   ]
}
```

Volume lock requests

Volume lock requests include the following components:

- An **action** parameter with a value of `lock`.
- A **params** parameter, containing the following parameter:
  - **name**: The volume name.

Example:

```
POST /hsm/v5/:system1/volumes
{
   "request": [
               {
                   "action": "lock",
                   "params": {
                                       "name": "vol1"
                              }
               }
   ]
}
```

Alternatively, a volume can be locked with the following request:

```
POST /hsm/v5/:system1/volumes/:vol1
{
   "request": {
                   "action": "lock"
               }
}
```

Volume unlock requests

Volume unlock requests include the following components:

- An **action** parameter with a value of `unlock`.
- A **params** parameter, containing the following parameter:
  - **name**: The volume name.

Example:

```
POST /hsm/v5/:system1/volumes
{
   "request": [
               {
                   "action": "unlock",
                   "params": {
                                       "name": "vol1"
                              }
               }
   ]
}
```

Alternatively, a volume can be unlocked with the following request:

```
POST /hsm/v5/:system1/volumes/:vol1
{
   "request": {
                   "action": "unlock"
               }
}
```

Volume rename requests

Volume rename requests include the following components:

- An **action** parameter with a value of `rename`.
- A **params** parameter, containing the following parameters:
  - **name**: The volume name.
  - **new_name**: The new name of the volume.

Example:

```
POST /hsm/v5/:system1/volumes
{
   "request": [
               {
                   "action": "rename",
                   "params": {
                                       "name": "vol1",
                                       "new_name":"vol2"
                                }
               }
   ]
}
```

Alternatively, a volume can be renamed with the following request:

```
POST /hsm/v5/:system1/volumes/:vol1
{
   "request": {
                   "action": "rename",
                   "params":{
                                       "new_name":"vol2"
                             }
               }
}
```

Volume copy requests

Volume copy requests include the following components:

- An **action** parameter with a value of `copy`.
- A **params** parameter, containing the following parameters:
  - **name**: The volume name.
  - **target**: The name of the target volume.

Example:

```
POST /hsm/v5/:system1/volumes
{
   "request": [
               {
                   "action": "copy",
                   "params": {
                                       "name": "vol1",
                                       "target":"vol2"
                                }
               }
   ]
}
```

Alternatively, a volume can be copied with the following request:

```
POST /hsm/v5/:system1/volumes/:vol1
{
   "request": {
                   "action": "copy",
                   "params":{
                                       "target":"vol2"
                             }
               }
}
```

Volume move requests

A request to move a volume to a new pool includes the following components:

- An **action** parameter with a value of `move`.
- A **params** parameter, containing the following parameters:
  - **name**: The volume name.

– **pool**: The name of the target pool.

Example:

```
POST /hsm/v5/:system1/volumes
{
   "request": [
               {
                   "action": "move",
                   "params": {
                                       "name": "vol1",
                                       "pool":"pool1"
                              }
               }
   ]
}
```

Alternatively, a volume can be moved to a new pool with the following request:

```
POST /hsm/v5/:system1/volumes/:vol1
{
   "request": {
                   "action": "move",
                   "params":{
                                      "pool":"pool1"
                             }
               }
}
```

Volume deletion requests

Volume deletion requests include the following components:

- An **action** parameter with a value of `delete`.
- A **params** parameter, containing the following parameters:
  – **name**: The volume name.

Example:

```
POST /hsm/v5/:system1/volumes
{
   "request": [
               {
                   "action": "delete",
                   "params": {
                                       "name": "vol1"
                              }
               }
   ]
}
```

Alternatively, a volume can be deleted using either of the following requests, where vol1 is the volume name:

```
POST /hsm/v5/:system1/volumes/:vol1
{
   "request": [
           {
                  "action": "delete"
              }
               ]
}
```

```
DELETE /hsm/v4/:system1/volumes/:vol1
```

Volume restore from snapshot requests

Volume restore from snapshot requests include the following components:

- An **action** parameter with a value of `restore_from_snapshot`.
- A **params** parameter, containing the following parameters:
  - **snapshot**: The source snapshot name.

Example:

```
POST /hsm/v5/:system1/volumes
{
   "request": [
               {
                   "action": "restore_from_snapshot",
                   "params": {
                                       "snapshot": "snap1"
                              }
               }
   ]
}
```

Volume snapshot update requests

The following types of volume snapshot update requests are supported.

Volume snapshot creation requests

Volume snapshot creation requests include the following components:

- An **action** parameter with a value of `create`.
- A **params** parameter, containing the following parameters:
  - **source**: The name of the volume.
  - **target** (optional): The snapshot name.
  - **delete_priority** (optional): The priority. Possible values are integers 0–4, and the default value is 1.

Example:

```
POST /hsm/v5/:system1/snapshots
{
   "request": [
               {
                   "action": "create",
                   "params": {
                                       "source": "vol1",
                                       "target": "snapshot1",
                                       "delete_priority": "2"
                              }
               }
   ]
}
```

Volume snapshot overwrite requests

Volume snapshot overwrite requests include the following components:

- An **action** parameter with a value of `overwrite`.
- A **params** parameter, containing the following parameters:
  - **source**: The name of the volume.
  - **target**: The snapshot name.

Example:

```
POST /hsm/v5/:system1/snapshots
{
   "request": [
               {
                   "action": "overwrite",
                   "params": {
                                   "source": "vol1",
                                   "target": "snapshot1"
               }
               }
   ]
}
```

Alternatively, the volume snapshot may be overwritten using the following update request, where the snapshot name is used as a part of the URL:

```
POST /hsm/v5/:system1/snapshots/snapshot1
{
   "request": [
               {
                   "action": "overwrite",
                   "params": {
                                   "source": "vol1"
               }
               }
   ]
}
```

Volume snapshot lock requests

Volume snapshot lock requests include the following components:

- An **action** parameter with a value of `lock`.
- A **params** parameter, containing the following parameter:
  - **name**: The snapshot name.

Example:

```
POST /hsm/v5/:system1/snapshots
{
   "request": [
               {
                   "action": "lock",
                   "params": {
                                   "name": "snapshot1"
                              }
               }
   ]
}
```

Alternatively, a snapshot may be locked using the following update request, where snapshot1 is the volume snapshot name:

```
POST /hsm/v5/:system1/snapshots/snapshot1
{
   "request": {
                "action": "lock"                }
             }
}
```

Volume snapshot unlock requests

Volume snapshot unlock requests include the following components:

- An **action** parameter with a value of `unlock`.
- A **params** parameter, containing the following parameter:
  - **name**: The snapshot name.

Example:

```
POST /hsm/v5/:system1/snapshots
{
   "request": [
             {
                "action": "unlock",
                "params": {
                                "name": "snapshot1"
                           }
             }
   ]
}
```

Alternatively, a snapshot may be un locked using the following update request, where snapshot1 is the volume snapshot name:

```
POST /hsm/v5/:system1/snapshots/snapshot1
{
   "request": {
                "action": "unlock"                }
             }
}
```

Volume snapshot duplication requests

Volume snapshot duplication requests include the following components:

- An **action** parameter with a value of `duplicate`.
- A **params** parameter, containing the following parameters:
  - **source**: The name of the volume.
  - **target**: The snapshot name.

Example:

```
POST /hsm/v4/:system1/snapshots
{
   "request": [
             {
                "action": "duplicate",
                "params": {
                                "source": "src",
                                "target": "target",
                           }
             }
   ]
}
```

Volume snapshot deletion requests

Volume snapshot deletion requests include the following components:

- An **action** parameter with a value of `delete`.
- A **params** parameter, containing the following parameter:
  - **name**: The snapshot name.

Example:

```
POST /hsm/v5/:system1/snapshots
{
   "request": [
               {
                   "action": "delete",
                   "params": {
                                       "name": "snapshot1"
                              }
               }
   ]
}
```

Alternatively, a snapshot can also be deleted using either of the following update requests, where snapshot1 is the volume snapshot name:

```
POST /hsm/v5/:system1/snapshots/snapshot1
{
   "request": [
               {
                   "action": "delete"
               }
               ]
}
```

```
DELETE /hsm/v5/:system1/snapshots/:snapshot1
```

Volume snapshot format requests

Volume snapshot format requests include the following components:

- An **action** parameter with a value of `format`.
- A **params** parameter, containing the following parameter:
  - **name**: The snapshot name.

Example:

```
POST /hsm/v5/:system1/snapshots
{
   "request": [
               {
                   "action": "format",
                   "params": {
                                       "name": "snapshot1"
                              }
               }
   ]
}
```

Volume snapshot restore volume requests

Volume snapshot requests for volume restore include the following components:

- An **action** parameter with a value of `restore_volume`.
- A **params** parameter, containing the following parameter:
  - **name**: Name of the snapshot with which to restore its master volume.

Example:

```
POST /hsm/v5/:system1/snapshots
{
   "request": [
               {
                   "action": "restore_volume",
                   "params": {
                                       "name": "snapshot1"
                              }
               }
   ]
}
```

Volume snapshot restore snapshot requests

Volume snapshot requests for snapshot restore include the following components:

- An **action** parameter with a value of `restore_snapshot`.
- A **params** parameter, containing the following parameter:
  - **source**: Name of the source snapshot.
  - **target**: Name of the target snapshot.

Example:

```
POST /hsm/v5/:system1/snapshots
{
   "request": [
               {
                   "action": "restore_snapshot",
                   "params": {
                                       "source": "snapshot1"
                                       "target": "snapshot2"
                              }
               }
   ]
}
```

Consistency group update requests

The following types of consistency group update requests are supported.

- Consistency group creation requests
- Consistency group add volume requests
- Consistency group remove volume requests
- Consistency group delete requests
- “Consistency group rename requests” on page 33
- “Consistency group restore from snapshot group requests” on page 34

Consistency group creation requests

Consistency group creation requests include the following components:

- An **action** parameter with a value of `create`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the consistency group.
  - **pool**: The name of the pool.

Example:

```
POST /hsm/v5/:system1/cgs
{
   "request": [
               {
                   "action": "create",
                   "params": {
                                       "name": "cg1",
                                       "pool": "pool1"
                               }
               }
   ]
}
```

Consistency group add volume requests

Consistency group add volume requests include the following components:

- An **action** parameter with a value of `add_volume`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the consistency group to which the volume will be added.
  - **volume**: The name of the volume to add.

Example:

```
POST /hsm/v5/:system1/cgs
{
   "request": [
               {
                   "action": "add_volume",
                   "params": {
                                       "name": "cg1",
                                       "volume": "vol1"
                               }
               }
   ]
}
```

Alternatively, adding a volume to a consistency group may be done using the following request:

```
POST /hsm/v5/:system1/cgs/:cg1
{
   "request": [
               {
                   "action": "add_volume",
                   "params": {
                                       "volume": "vol1"
                               }
               }
   ]
}
```

Consistency group remove volume requests

Consistency group remove volume requests include the following components:

- An **action** parameter with a value of `remove_volume`.
- A **params** parameter, containing the following parameter:
  - **volume**: The name of the volume to remove.

Example:

```
POST /hsm/v5/:system1/cgs
{
   "request": [
               {
                   "action": "remove_volume",
                   "params": {
                                       "volume": "vol1"
                              }
               }
   ]
}
```

Consistency group deletion requests

Consistency group deletion requests include the following components:

- An **action** parameter with a value of `delete`.
- A **params** parameter, containing the following parameter:
  - **name**: The name of the consistency group to delete.

Example:

```
POST /hsm/v5/:system1/cgs
{
   "request": [
               {
                   "action": "delete",
                   "params": {
                                       "name": "cg1"
                              }
               }
   ]
}
```

Alternatively, a consistency group may be deleted using either of the following requests, where `cg1` is the name of the consistency group:

```
POST /hsm/v5/:system1/cgs/:cg1
{
   "request": [
               {
                   "action": "delete",
                              }
   ]
}
```

```
DELETE /hsm/v5/:system1/cgs/:cg1
```

Consistency group rename requests

Consistency group rename requests include the following components:

- An **action** parameter with a value of `rename`.
- A **params** parameter, containing the following parameter:
  - **name**: The name of the consistency group to rename.
  - **new_name**: The new name of the consistency group.

Example:

```
POST /hsm/v5/:system1/cgs
{
   "request": [
               {
                   "action": "rename",
                   "params": {
                                       "name": "cg1",
                                       "new_name": "cg2"
                               }
               }
   ]
}
```

Consistency group restore from snapshot group requests

Consistency group restore from snapshot group requests include the following components:

- An **action** parameter with a value of `restore_from_snapshot_group`.
- A **params** parameter, containing the following parameter:
  - **snapshot_group**: The name of the source snapshot group.

Example:

```
POST /hsm/v5/:system1/cgs
{
   "request": [
               {
                   "action": "restore_from_snapshot_group",
                   "params": {
                                       "snapshot_group": "sg1",
                               }
               }
   ]
}
```

Snapshot group update requests

The following types of snapshot group update requests are supported.

- Snapshot group creation requests
- Snapshot group overwrite requests
- Snapshot group lock requests
- Snapshot group unlock requests
- Snapshot group delete requests
- “Snapshot group rename requests” on page 37
- “Snapshot group format requests” on page 37
- “Snapshot group restore requests” on page 38
- “Snapshot group disband requests” on page 38
- “Snapshot group duplicate requests” on page 38

Snapshot group creation requests

Snapshot group creation requests include the following components:

- An **action** parameter with a value of `create`.
- A **params** parameter, containing the following parameters:
  - **source**: The name of the consistency group.
  - **target** (optional): The name of the new snapshot group.
  - **delete_priority** (optional): The priority. Possible values are integers 0–4, and the default value is 1.

Example:

```
POST /hsm/v5/:system1/snap_groups
{
   "request": [
               {
                   "action": "create",
                   "params": {
                                       "source": "cg1",
                                       "target": "sg1"
                                       "delete_priority": "2"
                              }
               }
   ]
}
```

Snapshot group overwrite requests

Snapshot group overwrite requests include the following components:

- An **action** parameter with a value of `overwrite`.
- A **params** parameter, containing the following parameters:
  - **source**: The name of the consistency group.
  - **target**: The name of the snapshot group to be overwritten.

Example:

```
POST /hsm/v5/:system1/snap_groups
{
   "request": [
               {
                   "action": "overwrite",
                   "params": {
                                       "source": "cg1",
                                       "target": "sg1"
                              }
               }
   ]
}
```

Alternatively, the snapshot group may be overwritten using the following update request, with the snapshot group name (sg1 in this example) being used as a part of the URL:

```
POST /hsm/v5/:system1/snap_groups/:sg1
{
   "request": [
               {
                   "action": "overwrite",
                   "params": {
                                       "source": "cg1",
                              }
               }
   ]
}
```

Snapshot group lock requests

Snapshot group lock requests include the following components:

- An **action** parameter with a value of `lock`.
- A **params** parameter, containing the following parameter:
  - **name**: The snapshot group name.

Example:

```
POST /hsm/v5/:system1/snap_groups
{
   "request": [
               {
                   "action": "lock",
                   "params": {
                                       "name": "sg1"
                              }
               }
   ]
}
```

Alternatively, a snapshot group can be locked using the following update request, with sg1 as the snapshot group name in this example:

```
POST /hsm/v5/:system1/snap_groups/:sg1
{
   "request": {
                   "action": "lock",
               }
}
```

Snapshot group unlock requests

Snapshot group unlock requests include the following components:

- An **action** parameter with a value of `unlock`.
- A **params** parameter, containing the following parameters:
  - **name**: The snapshot group name.

Example:

```
POST /hsm/v5/:system1/snap_groups
{
   "request": [
               {
                   "action": "unlock",
                   "params": {
                                       "name": "sg1"
                              }
               }
   ]
}
```

Alternatively, a snapshot group can be locked using the following update request, with `sg1` as the snapshot group name in this example:

```
POST /hsm/v5/:system1/snap_groups/:sg1
{
   "request": {
                   "action": "lock",
               }
}
```

Snapshot group deletion requests

Snapshot group deletion requests include the following components:

- An **action** parameter with a value of `deletion`.
- A **params** parameter, containing the following parameter:
  - **name**: The snapshot group name.

Example:

```
POST /hsm/v5/:system1/snap_groups
{
   "request": [
               {
                   "action": "deletion",
                   "params": {
                                       "name": "sg1"
                              }
               }
   ]
}
```

Alternatively, a snapshot group can be deleted using either of the following update requests, where `sg1` is the snapshot group name:

```
POST /hsm/v5/:system1/snap_groups/:sg1
{
   "request": [
                 {
                   "action": "deletion",
                  }
               ]
}
```

```
DELETE /hsm/v5/:system1/snap_groups/:sg1
```

Snapshot group rename requests

Snapshot group rename requests include the following components:

- An **action** parameter with a value of `rename`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the snapshot group.
  - **new_name**: The new name of the snapshot group.

Example:

```
POST /hsm/v5/:system1/snap_groups
{
   "request": [
               {
                   "action": "rename",
                   "params": {
                                       "name": "sg1",
                                       "new_name": "new_sg1"
                              }
               }
   ]
}
```

Snapshot group format requests

Snapshot group format requests include the following components:

- An **action** parameter with a value of `format`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the snapshot group.

Example:

```
POST /hsm/v5/:system1/snap_groups
{
   "request": [
               {
                   "action": "format",
                   "params": {
                                       "name": "sg1",
                               }
               }
   ]
}
```

Snapshot group restore requests

Snapshot group restore requests include the following components:

- An **action** parameter with a value of `restore_snapshot_group`.
- A **params** parameter, containing the following parameters:
  - **source**: Name of the snapshot group from which to restore its master volumes.
  - **target**: The name of the snapshot group to be restored.

Example:

```
POST /hsm/v5/:system1/snap_groups
{
   "request": [
               {
                   "action": "restore_snapshot_group",
                   "params": {
                                       "source": "source_snap_groups",
                                       "target": "target_snap_groups"
                               }
               }
   ]
}
```

Snapshot group disband requests

Snapshot group disband requests include the following components:

- An **action** parameter with a value of `disband`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the snapshot group to be disbanded.

Example:

```
POST /hsm/v5/:system1/snap_groups
{
   "request": [
               {
                   "action": "disband",
                   "params": {
                                       "name": "sg1",
                               }
               }
   ]
}
```

Snapshot group duplicate requests

Snapshot group duplicate requests include the following components:

- An **action** parameter with a value of `duplicate`.
- A **params** parameter, containing the following parameters:
  - **source**: Name of the snapshot group to be duplicated.
  - **target** (optional): Name of the newly generated snapshot group.

Example:

```
POST /hsm/v5/:system1/snap_groups
{
   "request": [
               {
                   "action": "duplicate",
                   "params": {
                                       "source": "source_snap_groups",
                                       "target": "target_snap_groups"
                                   }
               }
   ]
}
```

Volume mapping update requests

Volume mapping is an artificial/virtual object; it represents relations/mappings between hosts (or clusters) and volumes. A single mapping name is created by concatenating the type (cluster or host) with the cluster/host name and volume name. For example, `host:h1:v1` represents host volume mapping, while `cluster:c1:v1` represents cluster volume mapping.

The following types of volume mapping update requests are supported.

- "Volume mapping creation requests" on page 39
- "Volume mapping deletion requests" on page 40

Volume mapping creation requests

Volume mapping creation requests include the following components:

- An **action** parameter with a value of `create`.
- A **params** parameter, containing the following parameters:
  - **volume**: The volume name.
  - **host_cluster_name**: The host name for host mappings or cluster name for cluster mappings.
  - **map_type**: The mapping type. Possible values include `host` (for host mappings) and `cluster` (for cluster mappings).
  - **lun**: The LUN ID.

Example of host mapping creation:

```
POST /hsm/v5/:system1/vol_maps
{
   "request": [
               {
                   "action": "create",
                   "params": {
                                       "volume": "vol1",
                                       "host_cluster_name": "h1",
                                       "map_type": "host",
                                       "lun": "3"
                                   }
               }
   ]
}
```

Example of cluster mapping creation:

```
POST /hsm/v5/:system1/vol_maps
{
   "request": [
               {
                   "action": "create",
                   "params": {
                                       "volume": "vol1",
                                       "host_cluster_name": "c1",
                                       "map_type": "cluster",
                                       "lun": "3"
                                   }
               }
   ]
}
```

Volume mapping deletion requests

Volume mapping deletion requests include the following components:

- An **action** parameter with a value of `delete`.
- A **params** parameter, containing the following parameters:
  - **volume**: The volume name.
  - **host_cluster_name**: The host name for host mappings or cluster name for cluster mappings.
  - **map_type**: The mapping type. Possible values include `host` (for host mappings) and `cluster` (for cluster mappings).

Example of host mapping deletion:

```
POST /hsm/v5/:system1/vol_maps
{
   "request": [
               {
                   "action": "delete",
                   "params": {
                                       "volume": "vol1",
                                       "host_cluster_name": "h1",
                                       "map_type": "host"
                                   }
               }
   ]
}
```

Example of cluster mapping deletion:

```
POST /hsm/v5/:system1/vol_maps
{
   "request": [
               {
                   "action": "delete",
                   "params": {
                                       "volume": "vol1",
                                       "host_cluster_name": "c1",
                                       "map_type": "cluster"
                                   }
               }
               ]
}
```

Mirror update requests

The following types of mirror update requests are supported.

Asynchronous mirror creation requests

Requests to create an asynchronous mirror of a volume or consistency group include the following components:

- An **action** parameter with a value of `create_asnyc_mirror`.
- A **params** parameter, containing the following parameters:
  - **name**: Name of the focal volume or consistency group (CG) to be mirrored (the master).
  - **mirror_type**: The type of mirror to create. Possible values include `cg` and `volume`.
  - **target**: Name of the target (remote) system.
  - **remote_name**: Name of the remote peer (volume or CG) on the remote storage system.
  - **schedule_type**: The scheduling type. Possible values include `internal` and `external`.
  - **rpo**: The mirror recovery point objective, in the format **hh:mm:ss**, where **hh** is hours, **mm** is minutes, and **ss** is seconds). Possible values range from 30 seconds to 24 hours.
  - **activation_type**: The activation type. Possible values include `offline` and `online`.
  - **part_of_xmirror** (optional): Indicates whether or not the created mirror will be a part of an x-mirror. Possible values include `true` or `false`. If this parameter is omitted, the value defaults to false.

Note: The **part_of_xmirror** parameter is used for three-way mirror creation.

Example:

```
POST /hsm/v5/:system1/mirrors
{
   "request": [
               {
                   "action": "create_async_mirror",
                   "params": {
                                    "name": "mirror1",
                                    "mirror_type": "volume",
                                    "target": "system1",
                                    "remote_name": "remote_vol",
                                    "schedule_type":"internal",
                                    "rpo":"23:00:01",
                                    "activation_type":"online"
                                }
               }
   ]
}
```

Synchronous mirror creation requests

Requests to create a synchronous mirror of a volume or consistency group include the following components:

- An **action** parameter with a value of `create_snyc_mirror`.
- A **params** parameter, containing the following parameters:
  - **name**: Name of the focal volume or consistency group (CG) to be mirrored (the master).
  - **mirror_type**: The type of mirror to create. Possible values include `cg` and `volume`
  - **target**: Name of the target (remote) system.

- **remote_name**: Name of the remote peer (volume or CG) on the remote storage system.

Example:

```
POST /hsm/v5/:system1/mirrors
{
   "request": [
               {
                   "action": "create_sync_mirror",
                   "params": {
                                   "name": "mirror1",
                                   "mirror_type": "volume",
                                   "target": "system1",
                                   "remote_name": "remote_vol"
                               }
               }
   ]
}
```

Mirror activation requests

Requests to activate a volume or consistency group mirror include the following components:

- An **action** parameter with a value of `activate_mirror`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the master volume or consistency group (CG) to activate mirroring for.
  - **mirror_type**: The type of mirror to activate. Possible values include `cg` and `volume`.
  - **target** (optional): Name of the target (remote) system. Used if the source has more than one destination.

Example:

```
POST /hsm/v5/:system1/mirrors
{
   "request": [
               {
                   "action": "activate_mirror",
                   "params": {
                                   "name": "vol1",
                                   "mirror_type": "volume",
                               }
               }
   ]
}
```

Alternatively, the mirror may be activated using the following requests, where **vol1** is the name of the source volume (in the first example), and **cg1** is the name of the source consistency group (in the second example):

```
POST /hsm/v5/:system1/mirrors/:volume:vol1
{
   "request": [
               {
                   "action": "activate_mirror",
               }
   ]
}
```

```
POST /hsm/v5/:system1/mirrors/:cg:cg1
{
   "request": [
               {
                   "action": "activate_mirror",
               }
   ]
}
```

Mirror deactivation requests

Requests to deactivate a volume or consistency group mirror include the following components:

- An **action** parameter with a value of `deactivate_mirror`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the master volume or consistency group (CG) to deactivate mirroring for.
  - **mirror_type**: The type of mirror to deactivate. Possible values include `cg` and `volume`.
  - **target** (optional): Name of the target (remote) system. Used if the source has more than one destination.

Example:

```
POST /hsm/v5/:system1/mirrors
{
   "request": [
               {
                   "action": "deactivate_mirror",
                   "params": {
                                       "name": "vol1",
                                       "mirror_type": "volume",
                                       "target":"system1"
                              }
               }
   ]
}
```

Alternatively, the mirror may be deactivated using the following requests, where **vol1** is the name of the source volume (in the first example), and **cg1** is the name of the source consistency group (in the second example):

```
POST /hsm/v5/:system1/mirrors/:volume:vol1
{
   "request": [
               {
                   "action": "deactivate_mirror",
                   "params": {
                                       "target":"system1"
                              }
                }
   ]
}
```

```
POST /hsm/v5/:system1/mirrors/:cg:cg1
{
   "request": [
               {
                   "action": "deactivate_mirror",

                   "params": {
                                       "target":"system1"
                              }
               }
   ]
}
```

Mirror deletion requests

Requests to delete a volume or consistency group mirror include the following components:

- An **action** parameter with a value of `delete`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the focal volume or consistency group (CG) to delete.
  - **mirror_type**: The type of mirror to delete. Possible values include `cg` and `volume`.
  - **force_on_slave** (optional): Possible values include `true` and `false`. If omitted, the value defaults to `false`

– **target** (optional): Name of the target (remote) system. Used if the source has more than one destination.

Example:

```
POST /hsm/v5/:system1/mirrors
{
   "request": [
               {
                   "action": "delete",
                   "params": {
                                       "name": "vol1",
                                       "mirror_type": "volume",
                                       "force_on_slave":"true"
                                   }
               }
   ]
}
```

Alternatively, the mirror may be deleted using the following requests, where **vol1** is the name of the source volume (in the first example), and **cg1** is the name of the source consistency group (in the second example):

```
POST /hsm/v5/:system1/mirrors/:volume:vol1
{
   "request": [
               {
                   "action": "deactivate_mirror",
                   "params": {
                                       "force_on_slave":"true"
                                   }
                }
   ]
}
```

```
POST /hsm/v5/:system1/mirrors/:cg:cg1
{
   "request": [
               {
                   "action": "delete_mirror",

                   "params": {
                                       "target":"system1"
                                   }
               }
   ]
}
```

In addition, the mirror may be deleted using the following requests, where **vol1** is the name of the source volume (in the first example), and **cg1** is the name of the source consistency group (in the second example):

```
DELETE /hsm/v5/:<system name>/mirrors/:volume:vol1
```

```
DELETE /hsm/v5/:<system name>/mirrors/:cg:cg1
```

Both of the delete requests above assume that the value for the **force_on_slave** parameter is `false`.

Note: The two requests listed above will only work with sources that have only one associated mirror.

Change RPO requests

Requests to change mirroring RPO include the following components:

- An **action** parameter with a value of `modify_rpo`.
- A **params** parameter, containing the following parameters:

- **name**: The name of the master volume or consistency group (CG) to activate mirroring for.
- **mirror_type**: The type of mirror to activate. Possible values include `cg` and `volume`.
- **rpo**: A mirror recovery point objective value.

Example:

```
POST /hsm/v5/:system1/mirrors
{
   "request": [
               {
                   "action": "change_rpo",
                   "params": {
                                   "name": "vol1",
                                   "mirror_type": "volume",
                                   "rpo": "222",
                               }
               }
   ]
}
```

Mirrored snapshot creation requests

Requests to create a mirrored snapshot of a volume include the following components:

- An **action** parameter with a value of `create_mirrored_snapshot`.
- A **params** parameter, containing the following parameters:
  - **name**: Name of the volume.
  - **primary_snapshot**: Name of the local snapshot.
  - **secondary_snapshot**: Name of the remote snapshot.
  - **target** (optional): Name of the target (remote) system.

Example:

```
POST /hsm/v5/:system1/mirrors
{
   "request": [
               {
                   "action": "create_mirrored_snapshot",
                   "params": {
                                   "name": "mirror1",
                                   "primary_snapshot": "local_snap",
                                   "secondary_snapshot": "remote_snap",
                                   "target": "system1"
                               }
               }
   ]
}
```

Mirrored snapshot cancellation requests

Requests to cancel a mirrored snapshot of a volume include the following components:

- An **action** parameter with a value of `cancel_mirrored_snapshot`.
- A **params** parameter, containing the following parameters:
  - **name**: Name of the volume.
  - **target** (optional): Name of the target (remote) system.

Example:

```
POST /hsm/v5/:system1/mirrors
{
   "request": [
               {
                   "action": "cancel_mirrored_snapshot",
                   "params": {
                                       "name": "mirror1",
                                       "target": "system1"
                              }
               }
   ]
}
```

Mirrored snapshot group creation requests

Requests to create a mirrored snapshot group include the following components:

- An **action** parameter with a value of `create_mirrored_snapshot_group`.
- A **params** parameter, containing the following parameters:
  - **name**: Name of the CG.
  - **primary_snapshot_group**: Name of the local snapshot group.
  - **secondary_snapshot_group**: Name of the remote snapshot group.
  - **target** (optional): Name of the target (remote) system.

Example:

```
POST /hsm/v5/:system1/mirrors
{
   "request": [
               {
                   "action": "create_mirrored_snapshot_group",
                   "params": {
                                       "name": "mirror1",
                                       "primary_snapshot_group": "local_snap_gr",
                                       "secondary_snapshot_group": "remote_snap_gr",
                                       "target": "system1"
                              }
               }
   ]
}
```

Mirrored snapshot group cancellation requests

Requests to cancel a mirrored snapshot group include the following components:

- An **action** parameter with a value of `cancel_mirrored_snapshot_group`.
- A **params** parameter, containing the following parameters:
  - **name**: Name of the CG.
  - **target** (optional): Name of the target (remote) system.

Example:

```
POST /hsm/v5/:system1/mirrors
{
   "request": [
               {
                   "action": "cancel_mirrored_snapshot_group",
                   "params": {
                                       "name": "mirror1",
                                       "target": "system1"
                              }
               }
   ]
}
```

Host update requests

The following types of host update requests are supported.

Host creation requests

Host creation requests include the following components:

- An **action** parameter with a value of `create`.
- A **params** parameter, containing the following parameters:
  - **name**: The host name.
  - **iscsi_chap_name**: The host's CHAP name identifier.
  - **iscsi_chap_secret**: The password of the initiator used to authenticate to the system when CHAP is enabled.
  - **cluster** (optional): The cluster name. If not provided, the stand-alone host is defined.
  - **type** (optional): The host type. Possible values include `default`, `hpux`, `zvm`, `windows2008`, `allothers`, `linux`, `esxi`. If omitted, the default value is `default`.
  - **domains** (optional): The domain. If omitted, the host will not be associated with any domain. Use "*" to associate the host with all domains.

Note:

- The `type` and `cluster` parameters are exclusive; that is, only one of them can be provided in the request. The `type` parameter can only be specified for stand-alone hosts. If the `cluster` parameter is provided, the host type is derived from the cluster.
- If the parameter `cluster` is defined, and the target system supports domains, then the parameter `domains` should be defined.

Example:

```
POST /hsm/v5/:system1/hosts
{
   "request": [
               {
                   "action": "create",
                   "params": {
                                   "name" : "host1",
                                   "iscsi_chap_name" : "chap",
                                   "iscsi_chap_secret" : "1111111111111111"
                                   "type" : "hpux",
                                   "domains" : "domain1"
                               }
               }
   ]
}
```

Host rename requests

Host rename requests include the following components:

- An **action** parameter with a value of `rename`.
- A **params** parameter, containing the following parameters:

- **name**: The host name.
- **new_name**: The new name of the host.

Example:

```
POST /hsm/v5/:system1/hosts
{
   "request": [
               {
                   "action": "rename",
                   "params": {
                                      "name": "host1",
                                      "new_name": "host2"
                              }
               }
   ]
}
```

Alternatively, a host can be renamed with the following request:

```
POST /hsm/v5/:system1/hosts/:host1
{
   "request": {
                   "action": "rename",
                   "params": {
                                      "new_name": "host2"
                              }
               }
}
```

Host deletion requests

Host deletion requests include the following components:

- An **action** parameter with a value of `delete`.
- A **params** parameter, containing the following parameter:
  - **name**: The host name.

Example:

```
POST /hsm/v5/:system1/hosts
{
   "request": [
               {
                   "action": "delete",
                   "params": {
                                      "name": "host1"
                              }
               }
   ]
}
```

Alternatively, a host can be deleted with either of the following requests:

```
POST /hsm/v5/:system1/hosts/:h1
{
   "request": [
                {
                   "action": "delete"
                    }
               ]
}
```

```
DELETE /hsm/v5/:system1/hosts/:h1
```

Host type update requests

Host type update requests include the following components:

- An **action** parameter with a value of `change_type`.
- A **params** parameter, containing the following parameters:
  - **name**: The host name.
  - **type**: The host type. Possible values include `default`, `hpux`, `zvm`, `windows2008`, `allothers`, `linux`, `esxi`. If omitted, the default value is `default`.

Example:

```
POST /hsm/v5/:system1/hosts
{
   "request": [
               {
                   "action": "change_type",
                   "params": {
                                       "name" : "host1",
                                       "type" : "hpux"
                              }
               }
   ]
}
```

Host CHAP update requests

Host CHAP update requests include the following components:

- An **action** parameter with a value of `update_chap`.
- A **params** parameter, containing the following parameters:
  - **name**: The host name.
  - **iscsi_chap_name**: The host's CHAP name identifier.
  - **iscsi_chap_secret**: The password of the initiator used to authenticate to the system when CHAP is enabled.

Example:

```
POST /hsm/v5/:system1/hosts
{
   "request": [
               {
                   "action": "update_chap",
                   "params": {
                                       "name" : "host1",
                                       "iscsi_chap_name" : "chap",
                                       "iscsi_chap_secret" : "1111111111111111"
                              }
               }
   ]
}
```

Host port update requests

Host ports represent ports that are defined on the host/client side. A single host port name is created by concatenating the port **type** (`fc` or `iscsi`), host name (**host**), and port name (**port**), as follows:

- **For FC ports:** `fc:h1:p1`, where h1 is the host name, and p1 is the port name.
- **For iSCSI ports:** `iscsi:h1:p1`, where h1 is the host name, and p1 is the port name

The following types of host port update requests are supported.

- Host port creation requests
- Host port deletion requests

Host port creation requests

Host port creation requests include the following components:

- An **action** parameter with a value of `create`.
- A **params** parameter, containing the following parameters:
  - **port**: The port name.
  - **host**: The host name.
  - **type**: The port type. Possible values include `iscsi` and `fc`.

Example:

```
POST /hsm/v4/:system1/host_ports
{
   "request": [
               {
                   "action": "create",
                   "params": {
                                      "port": "10000000C9926DCA",
                                      "host": "h1",
                                      "type": "fc"
                                  }
               }
   ]
}
```

Host port deletion requests

Host port deletion requests include the following components:

- An **action** parameter with a value of `delete`.
- A **params** parameter, containing the following parameters:
  - **port**: The port name.
  - **host**: The host name.
  - **type**: The port type. Possible values include `iscsi` and `fc`.

Example:

```
POST /hsm/v4/:system1/host_ports
{
   "request": [
               {
                   "action": "delete",
                   "params": {
                                      "port": "10000000C9926DCA",
                                      "host": "h1",
                                      "type": "fc"
                                  }
               }
   ]
}
```

Alternatively, deleting FC port mappings can be done using either of the following requests (where h1 is the host name and p1 is the port name):

```
POST /hsm/v4/:system1/host_ports/:fc:h1:p1
{
   "request": [
                   {
                   "action": "delete"
                   }
               ]
}
```

```
DELETE /hsm/v2/:system1/host_ports/:fc:h1:p1
```

Likewise, deleting iSCSI port mappings can be done using either of the following requests (where h1 is the host name and p1 is the port name):

```
POST /hsm/v4/:system1/host_ports/:iscsi:h1:p1
{
   "request": [
               {
                   "action": "delete"
               }
   ]
}
```

```
DELETE /hsm/v4/:system1/host_ports/:iscsi:h1:p1
```

Cluster update requests

The following types of cluster update requests are supported.

- "Cluster creation requests" on page 51
- "Cluster add host requests" on page 51
- "Cluster remove host requests" on page 52
- "Cluster rename requests" on page 53
- "Cluster deletion requests" on page 53
- "Cluster change type requests" on page 54

Cluster creation requests

Cluster creation requests include the following components:

- An **action** parameter with a value of `create`.
- A **params** parameter, containing the following parameters:
  - **name**: The cluster name.
  - **type** (optional): The cluster type. Possible values include `default`, `hpux`, `zvm`, `windows2008`, `allothers`, `linux`, `esxi`. If omitted, the default value is `default`.
  - **domains** (optional): The domain. If omitted, the cluster will not be associated with any domain. Use "*" to associate the cluster with all domains.

Example:

```
POST /hsm/v5/:system1/clusters
{
   "request": [
               {
                   "action": "create",
                   "params": {
                                       "name": "c1",
                                       "type": "zvm",
                                       "domains": "domain1"
                                   }
               }
   ]
}
```

Cluster add host requests

Cluster add host requests include the following components:

- An **action** parameter with a value of `add_host`.
- A **params** parameter, containing the following parameters:
  - **name**: The cluster name.

- **host**: The name of the host.
- **map**: Indicates the type of mapping. Possible values include `host` and `cluster`.

Example:

```
POST /hsm/v5/:system1/clusters
{
   "request": [
               {
                   "action": "add_host",
                   "params": {
                                       "name": "cluster1",
                                       "host": "host1",
                                       "map": "cluster"
                                   }
               }
   ]
}
```

Alternatively, a host can be added to a cluster with the following request:

```
POST /hsm/v5/:system1/clusters/:cluster1
{
         "request": {
                        "action": "add_host",
                        "params": {
                                       "host":"host1",
                                       "map":"cluster"
                                    }
                    }
}
```

Cluster remove host requests

Cluster remove host requests include the following components:

- An **action** parameter with a value of `remove_host`.
- A **params** parameter, containing the following parameters:
  - **name**: The cluster name.
  - **host**: The name of the host.

Example:

```
POST /hsm/v5/:system1/clusters
{
   "request": [
               {
                   "action": "remove_host",
                   "params": {
                                       "name": "cluster1",
                                       "host": "host1"
                                   }
               }
   ]
}
```

Alternatively, a host can be removed from a cluster with the following request:

```
POST /hsm/v5/:system1/clusters/:cluster1
{
         "request": {
                        "action": "remove_host",
                        "params": {
                                       "host":"host1"
                                    }
                    }
}
```

Cluster rename requests

Cluster rename requests include the following components:

- An **action** parameter with a value of `rename`.
- A **params** parameter, containing the following parameters:
  - **name**: The cluster name.
  - **new_name**: The new name of the cluster.

Example:

```
POST /hsm/v5/:system1/clusters
{
   "request": [
               {
                   "action": "rename",
                   "params": {
                                       "name": "cluster1",
                                       "new_name": "cluster2"
                               }
               }
   ]
}
```

Alternatively, a cluster can be renamed with the following request:

```
POST /hsm/v5/:system1/clusters/:cluster1
{
         "request": {
                        "action": "rename",
                        "params": {
                                      "new_name":"cluster2"
                                   }
                    }
}
```

Cluster deletion requests

Cluster deletion requests include the following components:

- An **action** parameter with a value of `delete`.
- A **params** parameter, containing the following parameters:
  - **name**: The cluster name.

Example:

```
POST /hsm/v5/:system1/clusters
{
   "request": [
               {
                   "action": "delete",
                   "params": {
                                       "name": "c1"
                               }
               }
   ]
}
```

Alternatively, a cluster can be deleted with either of the following requests:

```
POST /hsm/v5/:system1/clusters/:cluster1
{
         "request": [
                        {
                        "action": "delete"
                        }
                    ]
}
```

```
DELETE /hsm/v5/:system1/clusters/:c1
```

Cluster change type requests

Cluster change type requests include the following components:

- An **action** parameter with a value of `change_type`.
- A **params** parameter, containing the following parameters:
  - **name**: The cluster name.
  - **type**: The cluster type. Possible values include `default`, `hpux`, `zvm`, `windows2008`, `allothers`, `linux`, `esxi`. If omitted, the default value is `default`.

Example:

```
POST /hsm/v5/:system1/clusters
{
   "request": [
               {
                   "action": "change_type",
                   "params": {
                                       "name": "c1",
                                       "type": "allothers"
                              }
               }
   ]
}
```

Performance class update requests

The following types of performance class update requests are supported.

- "Performance class creation requests" on page 54
- "Performance class update IOPS limit requests" on page 55
- "Performance class update bandwidth limit requests" on page 56
- "Performance class rename requests" on page 56
- Performance class add domain requests
- Performance class remove domain requests
- "Performance class add pool requests" on page 58
- "Performance class remove pool requests" on page 58
- "Performance class add volume requests" on page 59
- "Performance class remove volume requests" on page 59
- "Performance class add host requests" on page 60
- "Performance class remove host requests" on page 60
- "Performance class deletion requests" on page 61

Performance class creation requests

Performance class creation requests include the following components:

- An **action** parameter with a value of `create`.
- A **params** parameter, containing the following parameters:
  - **name**: The performance class name.
  - **interface_ops_limit**: The IOPS limit for this performance class
  - **interface_bw_limit**: The bandwidth limit for this performance class.
  - **type**: The performance class type. Possible values are `SHARED` and `INDEPENDENT`.

Selecting SHARED, enables the transparent and dynamic sharing of data between multiple entities. If INDEPENDENT is selected, multiple entities do not share data. For example, if **interface_ops_limit** is set to 100, and 2 volumes are attached to the performance class, selecting SHARED will split the IOPS of 100 between the 2 volumes. Selecting INDEPENDENT will allocate the IOPS of 100 to each volume.

```
POST /hsm/v5/:system1/perf_classes
{
    "request":  [
             {
            "action": "create",
            "params": {
                    "name": "gold",
                    "interface_iops_limit”: "1500",
                    "interface_bw_limit”: "5000",
                    “type”: “SHARED”
                             }
        }
]
}
```

Alternatively, the performance class can be created by using the following request (where *gold* is the name of the performance class). For example:

```
POST /hsm/v5/:system1/perf_classes/:gold
{
    "request":  [
             {
            "action": "create",
            "params": {
                    "interface_iops_limit”: "1500",
                    "interface_bw_limit”: "5000",
                    “type”: “SHARED”
                             }
        }
]
}
```

Performance class update IOPS limit requests

Performance class update IOPS limit requests include the following components:

- An **action** parameter with a value of `update_iops_limit`.
- A **params** parameter, containing the following parameters:
  - **name**: The performance class name.
  - **interface_iops_limit**: The new iops limit for this performance class.

Example:

```
POST /hsm/v5/:system1/perf_classes
{
    "request":  [
             {
            "action": "update_iops_limit",
            "params": {
                                "name": "gold",
                                "interface_iops_limit": "1700"
             }
        }
]
}
```

Alternatively, an iops limit can be updated using the following request:

```
POST /hsm/v5/:system1/perf_classes/:gold
{
"request":  {
            "action": "update_iops_limit",
            "params": {
                                "interface_iops_limit": "1700"
             }
        }
}
```

Performance class update bandwidth limit requests

Performance class update bandwidth limit requests include the following components:

- An **action** parameter with a value of `update_bw_limit`.
- A **params** parameter, containing the following parameters:
  - **name**: The performance class name.
  - **interface_bw_limit**: The new bandwidth limit for this performance class.

```
POST /hsm/v5/:system1/perf_classes
{
    "request":  [
             {
            "action": "update_bw_limit",
            "params": {

                                  "name": "gold",
                                  "interface_bw_limit": "5000"
             }
        }
]
}
```

Alternatively, a bandwidth limit can be updated by using the following request:

```
POST /hsm/v5/:system1/perf_classes/:gold
{
"request":  {
            "action": "update_bw_limit",
            "params": {
                                "interface_bw_limit": "1700"
             }
        }
}
```

Performance class rename requests

Performance class rename requests include the following components:

- An **action** parameter with a value of `rename`.
- A **params** parameter, containing the following parameters:
  - **name**: The performance class name.
  - **new_name**: The new name of the performance class.

```
POST /hsm/v5/:system1/perf_classes
{
    "request":  [
             {
            "action": "rename",
            "params": {

                                  "name": "gold",
                                  "new_name": "silver"
             }
        }
]
}
```

Alternatively, a performance class can be renamed by using the following request:

```
POST /hsm/v5/:system1/perf_classes/:gold
{
"request":  {
            "action": "rename",
            "params": {
                                "new_name": "silver"
             }
        }
}
```

Performance class add domain requests

Performance class add domain requests include the following components:

- An **action** parameter with a value of `add_domain`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the performance class to which the domain is added.
  - **domain**: The name of the domain to add.

```
POST /hsm/v5/:system1/perf_classes
{
    "request":  [
             {
            "action": "add_domain”,
            "params": {
                                "name": "gold",
                                "domain": "domain1"
             }
        }
]
}
```

Alternatively, adding a domain to a performance class can be done by using the following request:

```
POST /hsm/v5/:system1/perf_classes/:gold
{
"request":  {
            "action": "add_domain”,
            "params": {
                                "domain": "domain1"
             }
        }
}
```

Performance class remove domain requests

Performance class remove domain requests include the following components:

- An **action** parameter with a value of `remove_domain`.
- A **params** parameter, containing the following parameters:
  - **domain**: The name of the domain to remove.

Example:

```
POST /hsm/v5/:system1/perf_classes
{
    "request":  [
             {
            "action": "remove_domain",
            "params": {
                                "domain": "domain1"
             }
        }
]
}
```

Alternatively, removing a domain from a performance class can be done using the following request:

```
POST /hsm/v5/:system1/perf_classes/:gold
{
"request":  {
            "action": "remove_domain",
            "params": {
                                "domain": "domain1"
             }
        }
}
```

Performance class add pool requests

Performance class add pool requests include the following components:

- An **action** parameter with a value of `add_pool`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the performance class to which the pool is added.
  - **pool**: The name of the pool to add.

```
POST /hsm/v5/:system1/perf_classes
{
    "request":  [
             {
            "action": "add_pool",
            "params": {
                                  "name": "gold",
                                  "pool": "pool1"
             }
        }
]
}
```

Alternatively, adding a pool to a performance class can be done by using the following request:

```
POST /hsm/v5/:system1/perf_classes/:gold
{
"request":  {
            "action": "add_pool",
            "params": {
                                "pool": "pool1"
             }
        }
}
```

Performance class remove pool requests

Performance class remove pool requests include the following components:

- An **action** parameter with a value of `remove_pool`.
- A **params** parameter, containing the following parameters:
  - **pool**: The name of the pool to remove.

Example:

```
POST /hsm/v5/:system1/perf_classes
{
    "request":  [
             {
            "action": "remove_pool",
            "params": {
                                "pool": "pool1"
             }
        }
]
}
```

Alternatively, removing a pool from a performance class can be done using the following request:

```
POST /hsm/v5/:system1/perf_classes/:gold
{
"request":  {
            "action": "remove_pool",
            "params": {
                                "pool": "pool1"
             }
        }
}
```

Performance class add volume requests

Performance class add volume requests include the following components (for IBM XIV Gen3 storage systems only):

- An **action** parameter with a value of `add_volume`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the performance class to which the volume is added.
  - **volume**: The name of the volume to add.

```
POST /hsm/v5/:system1/perf_classes
{
    "request":  [
             {
            "action": "add_volume",
            "params": {

                                "name": "gold",
                                "volume": "vol1"
             }
        }
]
}
```

Alternatively, adding a volume to a performance class can be done by using the following request:

```
POST /hsm/v5/:system1/perf_classes/:gold
{
"request":  {
            "action": "add_volume",
            "params": {
                                "volume": "vol1"
             }
        }
}
```

Performance class remove volume requests

Performance class remove volume requests include the following components (for IBM XIV Gen3 storage systems only):

- An **action** parameter with a value of `remove_volume`.
- A **params** parameter, containing the following parameters:
  - **volume**: The name of the volume to remove.

```
POST /hsm/v5/:system1/perf_classes
{
    "request":  [
                {
            "action": "remove_volume",
            "params": {
                                    "volume": "vol1"
                }
        }
]
}
```

Alternatively, removing a volume to a performance class can be done by using the following request:

```
POST /hsm/v5/:system1/perf_classes/:gold
{
"request":  {
            "action": "remove_volume",
            "params": {
                                "volume": "vol1"
                }
        }
}
```

Performance class add host requests

Performance class add host requests include the following components:

- An **action** parameter with a value of `add_host`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the performance class to which the host is added.
  - **host**: The name of the host to add.

Example:

```
POST /hsm/v5/:system1/perf_classes
{
    "request":  [
                {
            "action": "add_host",
            "params": {
                                    "name": "gold",
                                    "host": "host1"
                }
        }
]
}
```

Alternatively, adding a host to a performance class can be done using the following request:

```
POST /hsm/v5/:system1/perf_classes/:gold
{
"request":  {
            "action": "add_host",
            "params": {
                                "host": "host1"
                }
        }
}
```

Performance class remove host requests

Performance class remove host requests include the following components:

- An **action** parameter with a value of `remove_host`.
- A **params** parameter, containing the following parameters:
  - **host**: The name of the host to remove.

Example:

```
POST /hsm/v5/:system1/perf_classes
{
    "request":  [
                {
               "action": "remove_host",
               "params": {
                                    "host": "host1"
                }
           }
]
}
```

Alternatively, removing a host from a performance class can be done using the following request:

```
POST /hsm/v5/:system1/perf_classes/:gold
{
"request":  {
               "action": "remove_host",
               "params": {
                                   "host": "host1"
                }
           }
}
```

Performance class deletion requests

Performance class creation requests include the following components:

- An **action** parameter with a value of `delete`.
- A **params** parameter, containing the following parameters:
  - **name**: The performance class name.

Example:

```
POST /hsm/v5/:system1/perf_classes
{
    "request":  [
                {
               "action": "delete",
               "params": {
                       "name": "gold"
                      }
           }
      ]
}
```

Alternatively, the performance class may be deleted using the following request (where **gold** is the name of the performance class). For example:

```
POST /hsm/v5/:system1/perf_classes/:gold
{
    "request":  [
                {
               "action": "delete"
                }
]
}
```

```
DELETE /hsm/v5/:system1/perf_classes/:gold
```

Data migration update requests

The following types of data migration update requests are supported.

- “Data migration creation requests” on page 62

Data migration creation requests

Data migration creation requests include the following components:

- An **action** parameter with a value of `create`.
- A **params** parameter, containing the following parameters:
  - **volume**: The target volume name.
  - **target**: The target storage system name.
  - **lun**: The target storage system LUN number.
  - **update_source**: Source system update (*true* or *false*) .
  - **pool** (optional): The name of the pool if the volume does not exist. When specified, this parameter creates a target volume with the same name as the one specified for the pool.

Example:

```
POST /hsm/v5/:system1/data_migrations
{
  "request" : [
        {
          "action" : "create",
          "params" :
                {
                  "volume": "test_1",
                  "pool" : "test_1",
                  "target" : "gen4c-83-upperc-mf",
                  "lun" : "2",
                  "update_source" : "false"
                }
        }
  ]
}
```

Data migration activation requests

Data migration activation requests include the following components:

- An **action** parameter with a value of `activate`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the data migration object.

Example:

```
POST /hsm/v5/:system1/data_migrations
{
  "request" : [
        {
          "action" : "activate",
          "params" :
                {
                  "name": "dm_1",
                }
        }
  ]
}
```

Data migration deactivation requests

Data migration deactivation requests include the following components:

- An **action** parameter with a value of `deactivate`.

- A **params** parameter, containing the following parameters:
  - **name**: The name of the data migration object.

Example:

```
POST /hsm/v5/:system1/data_migrations
{
  "request" : [
        {
          "action" : "deactivate",
          "params" :
                {
                  "name": "dm_1",
                }
        }
  ]
}
```

Data migration test requests

Data migration test requests include the following components:

- An **action** parameter with a value of `test`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the data migration object.

Example:

```
POST /hsm/v5/:system1/data_migrations
{
  "request" : [
        {
          "action" : "test",
          "params" :
                {
                  "name": "dm_1",
                }
        }
  ]
}
```

Data migration deletion requests

Data migration deletion requests include the following components:

- An **action** parameter with a value of `delete`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the data migration object.

Example:

```
POST /hsm/v5/:system1/data_migrations
{
  "request" : [
        {
          "action" : "delete",
          "params" :
                {
                  "name": "dm_1",
                }
        }
  ]
}
```

Domain update requests

The following types of domain update requests are supported.

Domain creation requests

Domain creation requests include the following components:

- An **action** parameter with a value of `create`.
- A **params** parameter, containing the following parameters:
  - **name**: The domain name.
  - (In IBM XIV Gen3 storage systems) **hard_size**: The total physical size of the domain, in GB.
  - (In IBM XIV Gen3 storage systems) **soft_size**: The virtual size of the domain, in GB.
  - (In IBM FlashSystem A9000 and A9000R storage systems) **size**: The size of the domain, in GB.
  - **ldap_id** (optional): User ID for LDAP.
  - **max_pools_count** (optional): The maximum number of pools allowed per domain.
  - **max_volumes_snapshots_count** (optional): The maximum number of volumes and snapshots allowed per domain.
  - **max_cgs_count** (optional): The maximum number of CGs allowed per domain.
  - **max_replications_count** (optional): The maximum number of replication instances allowed per domain.
  - **max_data_migrations_count** (optional): The maximum number of data migration instances allowed per domain.

Example:

The output for IBM XIV Gen3 storage systems is similar to the following:

```
POST /hsm/v5/:system1/domains
{
  "request" : [
               {
                 "action" : "create",
                 "params" :
                       {
                         "name" : "test_rest",
                         "hard_size" : "22222",
                         "soft_size" : "33333",
                         "ldap_id" : "ldap_id",
                         "max_pools_count" : "11",
                         "max_volumes_snapshots_count" : "12",
                         "max_cgs_count" : "13",
                         "max_replications_count" : "14" ,
                         "max_data_migrations_count" : "15"
                       }
               }
             ]
}
```

The output for IBM FlashSystem A9000 and A9000R storage systems is similar to the following:

```
POST /hsm/v5/:system1/domains
{
  "request" : [
         {
           "action" : "create",
           "params" :
                 {
                   "name" : "test_rest",
                   "size" : "22222",
                   "ldap_id" : "ldap_id",
                   "max_pools_count" : "11",
                   "max_volumes_snapshots_count" : "12",
                   "max_cgs_count" : "13",
                   "max_replications_count" : "14" ,
                   "max_data_migrations_count" : "15"
                 }
         }
  ]
}
```

Domain editing requests

Domain editing requests include the following components:

- An **action** parameter with a value of `update`.
- A **params** parameter, containing the following parameters:
  - **name**: The domain name.
  - (In IBM XIV Gen3 storage systems) **hard_size** (optional): The total physical size of the domain, in GB.
  - (In IBM XIV Gen3 storage systems) **soft_size** (optional): The virtual size of the domain, in GB.
  - (In IBM FlashSystem A9000 and A9000R storage systems) **size**: The size of the domain, in GB.
  - **ldap_id** (optional): User ID for LDAP.
  - **max_pools_count** (optional): The maximum number of pools allowed per domain.
  - **max_volumes_snapshots_count** (optional): The maximum number of volumes and snapshots allowed per domain.
  - **max_cgs_count** (optional): The maximum number of CGs allowed per domain.
  - **max_replications_count** (optional): The maximum number of replication instances allowed per domain.
  - **max_data_migrations_count** (optional): The maximum number of data migration instances allowed per domain..

Example:

The output for IBM XIV Gen3 storage systems is similar to the following:

```
POST /hsm/v5/:system1/domains
{
  "request" : [
               {
                 "action" : "update",
                 "params" :
                       {
                         "name" : "test_rest",
                         "hard_size" : "22222",
                         "soft_size" : "33333",
                         "ldap_id" : "ldap_id",
                         "max_pools_count" : "11",
                         "max_volumes_snapshots_count" : "12",
                         "max_cgs_count" : "13",
                         "max_replications_count" : "14" ,
                         "max_data_migrations_count" : "15"
                       }
               }
            ]
}
```

The output for IBM FlashSystem A9000 and A9000R storage systems is similar to the following:

```
POST /hsm/v5/:system1/domains
{
  "request" : [
         {
           "action" : "update",
           "params" :
                 {
                   "name" : "test_rest",
                   "size" : "22222",
                   "ldap_id" : "ldap_id",
                   "max_pools_count" : "11",
                   "max_volumes_snapshots_count" : "12",
                   "max_cgs_count" : "13",
                   "max_replications_count" : "14" ,
                   "max_data_migrations_count" : "15"
                 }
         }
  ]
}
```

Domain rename requests

Domain rename requests include the following components:

- An **action** parameter with a value of `rename`.
- A **params** parameter, containing the following parameters:
  - **name**: The domain name.
  - **new_name**: The new name of the domain.

Example:

```
POST /hsm/v5/:system1/domains
{
   "request": [
               {
                   "action" : "rename",
                   "params" : {
                                    "name" : "dom1",
                                    "new_name" : "dom2"
                              }
               }
   ]
}
```

Domain deletion requests

Domain deletion requests include the following components:

- An **action** parameter with a value of `delete`.

- A **params** parameter, containing the following parameter:
  - **name**: The domain name.

Example:

```
POST /hsm/v5/:system1/domains
{
   "request": [
               {
                   "action" : "delete",
                   "params" : {
                                       "name" : "dom1"
                              }
               }
   ]
}
```

Domain add pool requests

Domain add pool requests include the following components:

- An **action** parameter with a value of `add_pool`.
- A **params** parameter, containing the following parameters:
  - **name**: The domain name.
  - **pool**: The name of the pool to be added to the domain.

Example:

```
POST /hsm/v5/:system1/domains
{
   "request": [
               {
                   "action" : "add_pool",
                   "params" : {
                                       "name" : "dom1"
                                       "pool" : "pool1"
                              }
               }
   ]
}
```

Domain move pool requests

Domain move pool requests include the following components:

- An **action** parameter with a value of `move_pool_to_domain`.
- A **params** parameter, containing the following parameters:
  - **pool**: The name of the pool to be moved between the domains.
  - **source**: The name of the source domain.
  - **target**: The name of the target domain.

Example:

```
POST /hsm/v5/:system1/domains
{
   "request": [
               {
                   "action" : "move_pool_to_domain",
                   "params" : {
                                       "pool" : "pool1"
                                       "source" : "dom1",
                                       "target" : "dom2"
                              }
               }
   ]
}
```

Domain remove pool requests

Domain remove pool requests include the following components:

- An **action** parameter with a value of `remove_pool`.
- A **params** parameter, containing the following parameters:
  - **name**: The domain name.
  - **pool**: The name of the pool to be removed from the domain.

Example:

```
POST /hsm/v5/:system1/domains
{
   "request": [
               {
                   "action" : "remove_pool",
                   "params" : {
                                       "name" : "dom1"
                                       "pool" : "pool1"
                               }
               }
   ]
}
```

Domain associate host requests

Domain associate host requests include the following components:

- An **action** parameter with a value of `associate_host`.
- A **params** parameter, containing the following parameters:
  - **name**: The domain name.
  - **host**: The name of the host to be associated with the domain.

Example:

```
POST /hsm/v5/:system1/domains
{
   "request": [
               {
                   "action" : "associate_host",
                   "params" : {
                                       "name" : "dom1"
                                       "host" : "host1"
                               }
               }
   ]
}
```

Domain unassociate host requests

Domain unassociate host requests include the following components:

- An **action** parameter with a value of `unassociate_host`.
- A **params** parameter, containing the following parameters:
  - **name**: The domain name.
  - **host**: The name of the host to be unassociated from the domain.

Example:

```
POST /hsm/v5/:system1/domains
{
   "request": [
               {
                   "action" : "unassociate_host",
                   "params" : {
                                       "name" : "dom1"
                                       "host" : "host1"
                              }
               }
   ]
}
```

Domain associate host cluster requests

Domain associate host cluster requests include the following components:

- An **action** parameter with a value of `associate_cluster`.
- A **params** parameter, containing the following parameters:
  - **name**: The domain name.
  - **cluster**: The name of the host cluster to be associated with the domain.

Example:

```
POST /hsm/v5/:system1/domains
{
   "request": [
               {
                   "action" : "associate_cluster",
                   "params" : {
                                       "name" : "dom1"
                                       "cluster" : "cluster1"
                              }
               }
   ]
}
```

Domain unassociate host cluster requests

Domain unassociate host cluster requests include the following components:

- An **action** parameter with a value of `unassociate_cluster`.
- A **params** parameter, containing the following parameters:
  - **name**: The domain name.
  - **cluster**: The name of the host cluster to be unassociated from the domain.

Example:

```
POST /hsm/v5/:system1/domains
{
   "request": [
               {
                   "action" : "unassociate_cluster",
                   "params" : {
                                       "name" : "dom1"
                                       "cluster" : "cluster1"
                              }
               }
   ]
}
```

Domain associate user requests

Domain associate user requests include the following components:

- An **action** parameter with a value of `associate_user`.
- A **params** parameter, containing the following parameters:

- **name**: The domain name.
- **user**: The name of the user to be associated with the domain.
- **exclusively** (optional): Restricts the user to domain's objects (*yes* or *no*).

Example:

```
POST /hsm/v5/:system1/domains
{
   "request": [
               {
                   "action" : "associate_user",
                   "params" : {
                                       "name" : "dom1"
                                       "user" : "user1"
                               }
               }
   ]
}
```

Domain unassociate user requests

Domain unassociate user requests include the following components:

- An **action** parameter with a value of `unassociate_user`.
- A **params** parameter, containing the following parameters:
  - **name**: The domain name.
  - **user**: The name of the user to be unassociated from the domain.

Example:

```
POST /hsm/v5/:system1/domains
{
   "request": [
               {
                   "action" : "unassociate_user",
                   "params" : {
                                       "name" : "dom1"
                                       "user" : "user1"
                               }
               }
   ]
}
```

Domain associate user group requests

Domain associate user group requests include the following components:

- An **action** parameter with a value of `associate_user_group`.
- A **params** parameter, containing the following parameters:
  - **name**: The domain name.
  - **user_group**: The name of the user group to be associated with the domain.

Example:

```
POST /hsm/v5/:system1/domains
{
   "request": [
               {
                   "action" : "associate_user_group",
                   "params" : {
                                       "name" : "dom1"
                                       "user_group" : "user_group1"
                               }
               }
   ]
}
```

Domain unassociate user group requests

Domain unassociate user group requests include the following components:

- An **action** parameter with a value of `unassociate_user_group`.
- A **params** parameter, containing the following parameters:
  - **name**: The domain name.
  - **user_group**: The name of the user group to be unassociated from the domain.

Example:

```
POST /hsm/v5/:system1/domains
{
   "request": [
               {
                   "action" : "unassociate_user_group",
                   "params" : {
                                       "name" : "dom1"
                                       "user_group" : "user_group1"
                              }
               }
   ]
}
```

Domain associate target requests

Domain associate target requests include the following components:

- An **action** parameter with a value of `associate_target`.
- A **params** parameter, containing the following parameters:
  - **name**: The domain name.
  - **target**: The name of the target storage system to be associated with the domain.

Example:

```
POST /hsm/v5/:system1/domains
{
   "request": [
               {
                   "action" : "associate_target",
                   "params" : {
                                       "name" : "dom1"
                                       "target" : "gen4c-24-upperd"
                              }
               }
   ]
}
```

Domain unassociate target requests

Domain unassociate target requests include the following components:

- An **action** parameter with a value of `unassociate_target`.
- A **params** parameter, containing the following parameters:
  - **name**: The domain name.
  - **target**: The name of the target storage system to be unassociated from the domain.

Example:

```
POST /hsm/v5/:system1/domains
{
   "request": [
               {
                   "action" : "unassociate_target",
                   "params" : {
                                       "name" : "dom1"
                                       "target" : "gen4c-24-upperd"
                               }
               }
   ]
}
```

Ethernet port update requests

The following types of Ethernet port update requests are supported.

- “Ethernet port MTU change requests” on page 72

Ethernet port MTU change requests

Ethernet port MTU change requests include the following components:

- An **action** parameter with a value of `change_mtu`.
- A **params** parameter, containing the following parameters:
  - **name**: The Ethernet port name.
  - **mtu**: MTU size.

Example:

```
POST /hsm/v5/:system1/ethernet_ports/change_mtu
{
  "request" : [
        {
          "action" : "change_mtu",
          "params" :
                {
                  "name" : "eth_port_14_6",
                  "mtu" : "5000"
                }
        }
  ]
}
```

IP interface update requests

The following types of IP interface update requests are supported.

- “IP interface creation requests” on page 72
- “IP interface editing requests” on page 73
- “IP interface deletion requests” on page 74
- “IP interface traceroute requests” on page 74
- “IP interface traceroute for IPv6 requests” on page 75
- “IP interface VLAN update requests” on page 75

IP interface creation requests

IP interface creation requests include the following components:

- An **action** parameter with a value of `create`.
- A **params** parameter, containing the following parameters:

- **name**: The IP interface name.
- **address**: IP address of the interface.
- **netmask**: Network mask of the interface.
- **gateway**: IP address of the default gateway for this interface.
- **module** (optional for IBM FlashSystem A9000 and A9000R): Component identifier (rack and module) of the module containing Ethernet ports.
- **port** (optional for IBM FlashSystem A9000 and A9000R): Port number.
- **ethernet_port_name** (optional for IBM FlashSystem A9000 and A9000R ver. 12.3.2 and above): Ethernet port name. Can be used instead of **module** and **port** combination.
- **vlan_id** (optional for IBM FlashSystem A9000 and A9000R ver. 12.3.2 and above): VLAN ID number.

Example:

The output for IBM XIV Gen3 storage systems is similar to the following:

```
POST /hsm/v5/:system1/ip_interface
{
  "request" : [
              {
                "action" : "create",
                "params" :
                      {
                        "name": "11.1",
                        "address" : "173.18.17.223",
                        "netmask" : "255.240.0.0",
                        "gateway" : "173.20.252.258",
                        "port" : "2",
                        "module" : "1:module:1"
                      }
              }
            ]
}
```

The output for IBM FlashSystem A9000 and A9000R storage systems is similar to the following:

```
POST /hsm/v5/:system1/ip_interface
{
  "request" : [
        {
          "action" : "create",
          "params" :
                {
                  "name": "11.1",
                  "address" : "173.18.17.223",
                  "netmask" : "255.240.0.0",
                  "gateway" : "173.20.252.258",
                  "ethernet_port_name" : "port_name"
                }
        }
  ]
}
```

IP interface editing requests

IP interface editing requests include the following components:

- An **action** parameter with a value of `update`.
- A **params** parameter, containing the following parameters:
  - **name**: The IP interface name.
  - **ip4_address** (optional): IPv4 address of the interface.
  - **ip6_address** (optional): IPv6 address of the interface.
  - **netmask** (optional): Network mask of the interface.
  - **ip4_gateway** (optional): IPv4 address of the default gateway for this interface.

- **ip6_gateway** (optional): IPv6 address of the default gateway for this interface.
- **module** (optional): Component identifier (rack and module) of the module containing Ethernet ports.
- **port** (optional): Port number.
- **access_group** (optional): The name of the IP access group used for IP filtering.

Example:

```
POST /hsm/v5/:system1/ip_interface
{
  "request" : [
                {
                  "action" : "update",
                  "params" :
                        {
                          "name": "11.1",
                          "ip4_address" : "172.16.11.247",
                          "netmask" : "255.240.0.0",
                          "ip4_gateway" : "172.16.250.254"
                        }
                }
              ]
}
```

IP interface deletion requests

IP interface deletion requests include the following components:

- An **action** parameter with a value of `delete`.
- A **params** parameter, containing the following parameters:
  - **name**: The IP interface name.

Example:

```
POST /hsm/v5/:system1/ip_interface
{
  "request" : [
                {
                  "action" : "delete",
                  "params" :
                        {
                          "name" : "11.1"
                        }
                }
              ]
}
```

IP interface traceroute requests

IP interface traceroute requests include the following components:

- An **action** parameter with a value of `traceroute`.
- A **params** parameter, containing the following parameters:
  - **local_ip_address**: IP address of the IP interface for which the traceroute command is run.
  - **remote_host_ip_address**: IP address or DNS for the traceroute test.
  - **vlan_id** (optional): VLAN ID for the traceroute test.

Example:

```
POST /hsm/v5/:system1/ip_interface
{
  "request" : [
               {
                 "action" : "traceroute",
                 "params" :
                        {
                          "local_ip_address" : "11.1.2.2",
                          "remote_host_ip_address" : "2"
                        }
               }
             ]
}
```

IP interface traceroute for IPv6 requests

IP interface traceroute for IPv6 network requests include the following components:

- An **action** parameter with a value of `traceroute_ip6`.
- A **params** parameter, containing the following parameters:
  - **local_ip_address**: IP address of the IP interface for which the traceroute command is run.
  - **remote_host_ip_address**: IP address or DNS for the traceroute test.

Example:

```
POST /hsm/v5/:system1/ip_interface
{
  "request" : [
               {
                 "action" : "traceroute_ip6",
                 "params" :
                        {
                          "local_ip_address" : "11.1.2.2",
                          "remote_host_ip_address" : "2"
                        }
               }
             ]
}
```

IP interface VLAN update requests

IP interface VLAN update requests include the following components:

- An **action** parameter with a value of `update_vlan`.
- A **params** parameter, containing the following parameters:
  - **vlan_pcp**: VLAN priority.

Example:

```
POST /hsm/v5/:system1/ip_interface
{
  "request" : [
               {
                 "action" : "update_vlan",
                 "params" :
                        {
                          "vlan_pcp" : "111"
                        }
               }
             ]
}
```

User update requests

The following types of user update requests are supported.

- “User creation requests” on page 76

User creation requests

User creation requests include the following components:

- An **action** parameter with a value of `create`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the user.
  - **password**: Password of the user to be created. The passwords are case-sensitive and must have between 6 and 12 characters.
  - **category** : The role of the user to be created. Available roles: *storageadmin*, *applicationadmin*, *readonly*, *securityadmin*, *storageintegrationadmin*.
  - **user_group** (optional): User group into which the user is to be added.
  - **area_code** (optional): Area code of the cellular phone number of the user.
  - **phone** (optional): Cellular phone number of the user for event notification via SMS, excluding the area code.
  - **email** (optional): Email address of the user.
  - **domains** (optional): Domain that the user will be attached to. To specify more than one domain, separate them with a comma.
  - **exclusive** (optional): Restricts the user to domain's objects (*yes* or *no*).

Example:

```
POST /hsm/v5/:system1/users/create
{
  "request" : [
        {
          "action" : "create",
          "params" :
                {
                  "name" : "user1",
                  "password" : "pwd1*",
                  "category" : "applicationadmin",
                  "area_code" : "03",
                  "phone" : "5229",
                  "email" : "a@b.c",
                  "user_group" : "user1_ug",
                  "exclusive" : "yes",
                  "domains" : "domain1"
                }
        }
  ]
}
```

User password change requests

User password change requests include the following components:

- An **action** parameter with a value of `change_password`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the user.
  - **password**: New user password. The passwords are case-sensitive and must have between 6 and 12 characters.

Example:

```
POST /hsm/v5/:system1/users
{
  "request" : [
        {
          "action" : "change_password",
          "params" :
                {
                  "name" : "user1",
                  "password" : "new_password",
                }
        }
  ]
}
```

User rename requests

Username change requests include the following components:

- An **action** parameter with a value of `rename`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the user.
  - **new_name**: New username.

Example:

```
POST /hsm/v5/:system1/users
{
  "request" : [
        {
          "action" : "rename",
          "params" :
                {
                  "name" : "user1",
                  "new_name" : "new_username",
                }
        }
  ]
}
```

User deletion requests

User deletion requests include the following components:

- An **action** parameter with a value of `delete`.
- A **params** parameter, containing the following parameters:
  - **name**: The name of the user.

Example:

```
POST /hsm/v5/:system1/users
{
  "request" : [
        {
          "action" : "delete",
          "params" :
                {
                  "name" : "user1",
                }
        }
  ]
}
```

Chapter 5. Request URL definitions

The following groups of request URLs are defined:

System request URLs

The following system request URLs are defined:

- `GET /hsm/v5/systems`: Lists all system properties managed by the Hyper-Scale Manager.
- `GET /hsm/v5/systems/:system1`: Lists all specific system properties.
- `GET /hsm/v5/systems/capacityReport`: Generates capacity report for all systems in your inventory.
- `GET /hsm/v5/systems/:<system-name>/capacityReport`: Generates capacity report for a specific system in your inventory.

System FC port request URLs

The following system FC port request URLs are defined:

- `GET /hsm/v5/:system1/fc_ports`: Lists all FC ports on a specific machine.
- `GET /hsm/v5/:system1/fc_ports/:p1`: Lists properties for a specific FC port.

System iSCSI port request URLs

The following system iSCSI port request URLs are defined:

- `GET /hsm/v5/:system1/iscsi_ports`: Lists all iSCSI ports on a specific machine.
- `GET /hsm/v5/:system1/iscsi_ports/:p1`: Lists properties for a specific iSCSI port.

Pool request URLs

The following pool request URLs are defined:

- `GET /hsm/v5/pools`: Lists all pools on all authorized systems.
- `GET/POST /hsm/v5/:system1/pools`: Lists all pools on a specific system; creates / updates / deletes pools.
- `GET/POST/DELETE /hsm/v5/:system1/hosts/:h1`: Lists properties of a specific pool; deletes a pool.

Volume request URLs

The following volume request URLs are defined:

- `GET /hsm/v5/volumes`: Lists all volumes on all authorized systems.
- `GET/POST /hsm/v5/:system1/volumes`: Lists all volumes on a specific system; creates / updates / deletes volumes.
- `GET /hsm/v5/:system1/volumes?pool=x`: Lists all volumes in a specific pool.
- `GET/POST/DELETE /hsm/v5/:system1/volumes/:vol1`: Lists specific volume properties; updates / deletes a volume.

Volume snapshot request URLs

The following volume snapshot request URLs are defined:

- `GET /hsm/v5/snapshots`: Lists all snapshots on all authorized systems.
- `GET /hsm/v5/:system1/snapshots?volume=vol1`: Lists all snapshots of a specific volume.
- `GET/POST /hsm/v5/:system1/snapshots`: Lists all snapshots on a specific system; creates / updates / deletes snapshots.
- `GET/POST/DELETE /hsm/v5/:system1/snapshots/:sn1`: Lists properties of a specific snapshot; deletes a snapshot.

Volume mapping request URLs

The following volume mapping request URLs are defined:

- `POST /hsm/v5/:system1/vol_maps`: Creates / deletes mappings.
- `GET /hsm/v5/:system1/vol_maps?host=x`: Lists all mappings of a specific host.
- `GET /hsm/v5/:system1/vol_maps?cluster=x`: Lists all mappings of a specific cluster.
- `GET /hsm/v5/:system1/vol_maps?volume=x`: Lists all mappings of a specific volume.
- `GET/DELETE /hsm/v5/:system1/vol_maps/:cluster:c1:vol1`: Lists properties of a specific mapping; deletes the mapping.
- `GET/DELETE /hsm/v5/:system1/vol_maps/:host:h1:vol1`: Lists properties of a specific mapping; deletes the mapping.

Consistency group request URLs

The following consistency group request URLs are defined:

- `GET /hsm/v4/cgs`: Lists all consistency groups on all authorized systems.
- `GET/POST /hsm/v4/:system1/cgs`: Lists all consistency groups on a specific system; creates / updates / deletes consistency groups.
- `GET/POST/DELETE /hsm/v4/:system1/cgs/:cg1`: Lists properties of a specific consistency group; deletes a consistency group.

Snapshot group request URLs

The following snapshot group request URLs are defined:

- `GET /hsm/v4/snap_groups`: Lists all snapshot groups on all authorized systems.

- `GET /hsm/v4/:system1/snap_groups?cg=cg1`: Lists all snapshot groups of a specific consistency group.
- `GET/POST /hsm/v4/:system1/snap_groups`: Lists all snapshot groups on a specific system; creates / updates / deletes snapshot groups.
- `GET/POST/DELETE /hsm/v4/:system1/snap_groups/:sg1`: Lists properties of a specific snapshot group; deletes a snapshot group.

Mirror request URLs

The following mirror request URLs are defined:

- `GET /hsm/v4/mirrors`: Lists all mirrors on all authorized systems.
- `GET/POST /hsm/v4/:system1/mirrors`: Lists all mirrors on a specific system; updates / deletes mirrors.
- `GET/POST /hsm/v4/:system1/mirrors/:cg:cg1`: Lists properties of a specific consistency group mirror; deletes the mirror.
- `GET/POST /hsm/v4/:system1/mirrors/:volume:vol1`: Lists properties of a specific volume mirror; deletes the mirror.

Host request URLs

The following host request URLs are defined:

- `GET /hsm/v5/hosts`: Lists all hosts' properties on all authorized systems.
- `GET/POST /hsm/v5/:system1/hosts`: Lists all hosts on a specific system; creates / updates / deletes hosts.
- `GET/POST/DELETE /hsm/v5/:system1/hosts/:h1`: Lists specific host properties; deletes a host.

Host port request URLs

The following host port request URLs are defined:

- `POST /hsm/v5/:system1/host_ports`: Add/remove ports to/from a specific host.
- `GET /hsm/v5/:system1/host_ports?host=x`: Lists all ports of a specific host.
- `GET /hsm/v5/:system1/host_ports?port=x`: Lists the host of a specific port.
- `GET/DELETE /hsm/v5/:system1/host_ports/:h1:p1`: Lists properties of a specific port; removes port from a host.

Cluster request URLs

The following cluster request URLs are defined:

- `GET /hsm/v5/clusters`: Lists all clusters on all authorized systems.
- `GET/POST /hsm/v5/:system1/clusters`: Lists all hosts on a specific system; creates / updates / deletes clusters.
- `GET/POST/DELETE /hsm/v5/:system1/clusters/:cl1`: Lists specific cluster properties; updates / deletes a cluster.

Performance class request URLs

The following performance class request URLs are defined:

- `GET /hsm/v5/perf_classes` : Lists all performance classes on all authorized systems.

- `GET/POST /hsm/v5/:system1/perf_classes`: Lists all performance classes on a specific system; creates / updates / deletes performance classes.
- `GET/POST/DELETE /hsm/v5/:system1/perf_classes/:gold`: Lists properties of a specific performance class; updates a performance class or deletes a performance class.
- `GET /hsm/v5/:system1/volumes?perf_class=gold`: Lists all of the volumes that are a part of a specific performance class.
- `GET /hsm/v5/:system1/hosts?perf_class=gold`: Lists all of the hosts that are a part of a specific performance class.
- `GET /hsm/v5/:system1/pools?perf_class=gold`: Lists all of the pools that are a part of a specific performance class.

Event request URLs

The following event request URL is defined:

- `GET /hsm/v5/:system1/events/`: Lists events from a specific system.

Chapter 6. Resource definitions

The resource formats that are returned by the REST API are defined in the following sections:

System resource definitions

The system resource formats that are returned by the REST API for XIV Gen3 and Spectrum Accelerate are as follows::

```
{
       "system": {
                 "allocation_limit": "82564000000000"
                 "ip_addresses":

                    {
                        "ip": "gen3p1-25b"
                    }
                 "safe_mode": "false"
                 "compression_savings": "0"
                 "system_state": "on"
                 "version": "11.6.2.c"
                 "compression_state": "compression_state_enabled"
                 "reclaimable_capacity": "258000000000"
                 "written_by_host": "412000000000"
                 "ref": "/hsm/v5/systems/:gen3p1-25b"
                 "allocated_volumes_and_snapshots": "2340000000000"
                 "free_to_allocate": "80224000000000"
                 "stored": "412000000000"
                 "target_state": "on"
                 "potential_compression_savings": "0"
                 "physical_free": "82151000000000"
                 "projected_system_capacity": "82564000000000"
                 "name": "gen3p1-25b"
                 "redundancy_status": "Full Redundancy"
                 "total_reduction_savings": "0"
                 "id": "XIV Gen3P1-25b:1310025:2810:114"
                 "physical_size": "82564000000000"
              }
}
```

The system resource formats that are returned by the REST API for FlashSystem A9000 and A9000R are as follows:

```
{
       "system": {
                 "allocation_limit": "1400071000000000"
                 "ip_addresses":

                    {
                       "ip": "gen4d-27c"
                    }
                 "safe_mode": "false"
                 "compression_savings": "8000000000"
                 "system_state": "on"
                 "compression_state": "compression_state_enabled"
                 "reclaimable_capacity": "1982000000000"
                 "deduplication_savings": "22000000000"
                 "written_by_host": "34000000000"
                 "ref": "/hsm/v5/systems/:gen4d-27c"
                 "allocated_volumes_and_snapshots": "10137000000000"
                 "free_to_allocate": "1389933000000000"
                 "stored": "3000000000"
                 "target_state": "on"
                 "physical_free": "11056000000000"
                 "projected_system_capacity": "55336000000000"
                 "name": "gen4d-27c"
                 "redundancy_status": "Fully Protected"
                 "total_reduction_savings": "30000000000"
                 "id": "XIV Gen4D-27c:6013766:9835:415"
                 "physical_size": "11059000000000"
                 }
}
```


System FC port resource definitions

The system FC port resource formats that are returned by the REST API are as follows:


```
{
       "fc_port": {
                  "ref": "/hsm/v5/:mn44/ fc_ports/:1234567890123001",
                  "id": "101280",
                  "wwpn": "1234567890123001",
                  "module": "6",
                  "port_number": "1",
                  "status": "OK",
                  "role": "Target",
                  "state": "Online",
                  "user_enabled": "true",
                  "configured_rate": "Auto",
                  "current_rate": "4",
                  "error_count": "0",
                  "system": "mn44"
                  }
}
```


System iSCSI port resource definitions

The system iSCSI port resource formats that are returned by the REST API are as follows:


```
{
        "iscsi_port": {
                      "ref": "/hsm/v5/:mn44/iscsi_ports/:myport",
                      "id": "101280",
                      "name": "myport",
                      "address": "1.1.1.13",
                      "netmask": "255.255.255.0",
                      "gateway": "1.1.1.1",
                      "mtu": "4500",
                      "module": "7",
                      "port_number": "1",
                      "system": "mn44"
                      }
}
```

Pool resource definitions

The pool resource formats that are returned by the REST API for XIV Gen3 and Spectrum Accelerate are as follows:

```
{
      "pool":
         {
              "lock_behavior": "read_only"
              "reduction_savings": "0"
              "volumes_space": "2804000000000"
              "perf_class": " "
              "soft_size": "3011000000000"
              "hard_size": "2512000000000"
              "reclaimable_capacity": "0"
              "free_to_allocate_volumes": "2202000000000"
              "volumes_physical_space": "2305000000000"
              "ref": "/hsm/v5/:gen3p1-25b/pools/:ErezPool"
              "system": "gen3p1-25b"
              "allocated_volumes": "602000000000"
              "snapshots_reserved": "206000000000"
              "snapshots_used": "0"
              "domain": "ErezDomain"
              "potential_compression_savings": "0"
              "volumes_written_by_host": "129000000000"
              "name": "ErezPool"
              "volumes_stored": "223000000000"
              "id": "62c14800001"
              "locked": "false"
              "pool_type": "thin"
         }
}
```

The pool resource formats that are returned by the REST API for FlashSystem A9000 and A9000R are as follows:

```
{
      "pool":
         {
              "reduction_savings": "N/A"
              "volumes_space": "620000000000"
              "perf_class": " "
              "free_to_allocate_volumes": "31000000000"
              "reclaimable_capacity": "567000000000"
              "volumes_physical_space": "N/A"
              "ref": "/hsm/v5/:gen4d-27c/pools/:alex_test"
              "system": "gen4d-27c"
              "size": "723000000000"
              "snapshots_reserved": "103000000000"
              "allocated_volumes": "588000000000"
              "snapshots_used": "0"
              "domain": "/Global Space/"
              "name": "alex_test"
              "volumes_written_by_host": "0"
              "id": "4e418c00002"
              "volumes_stored": "N/A"
              "locked": "false"
         }
}
```

Note: The **perf_class_ref** parameter is applicable only to objects limited by QOS.

Volume resource definitions

The volume resource formats that are returned by the REST API for XIV Gen3 and Spectrum Accelerate are as follows:


```
{
                "volume": {
                          "reduction_savings": "0"
                          "size_on_disk": "120000000000"
                          "cg": " "
                          "perf_class": "N/A"
                          "written_by_host_capacity": "26000000000"
                          "pool": "ExamplePool"
                          "snapshots_used_capacity": "0"
                          "mirrored": "true"
                          "wwn": "0017380027290003"
                          "pool_ref": "/hsm/v5/:gen3p1-25b/pools/:ExamplePool"
                          "ref": "/hsm/v5/:gen3p1-25b/volumes/:example_vol_001"
                          "system": "gen3p1-25b"
                          "unique_stored_data": "26000000000"
                          "size": "120000000000"
                          "stored_capacity": "26000000000"
                          "reduction_status": "Uncompressed"
                          "domain": "ExampleDomain"
                          "potential_compression_savings": "0"
                          "name": "example_vol_001"
                          "id": "62d14d00003"
                          "locked": "false"
                          "free_size": "93000000000"
                          }
}
```


The volume resource formats that are returned by the REST API for FlashSystem A9000 and A9000R are as follows:


```
{
                "volume": {
                          "reduction_ratio":"2.16",
                          "reduction_savings":"9307674781",
                          "stored_attributed_size":"8003455770",
                          "perf_class":"",
                          "written_by_host_capacity":"17310941184",
                          "compression_savings":"9307674781",
                          "data_reduction_iq_valid_on":"2018-12-20T13:14:10Z",
                          "snapshots_used_capacity":"N/A",
                          "mirrored":"false",
                          "wwn":"6001738CFC9035D30000000000EC3090",
                          "pool_ref":"/hsm/v5/:gen4d-60b/pools/:example_pool",
                          "deduplication_savings":"0",
                          "ref":"/hsm/v5/:gen4d-60b/volumes/:example_volume",
                          "stored_reclaimable_margin_of_error_size":"3841244265",
                          "reduction_savings_margin_of_error_size":"3841244265",
                          "stored_attributed_margin_of_error_size":"3841244265",
                          "id":"59b1aa00054",
                          "stored_reclaimable_size":"5337810207",
                          "locked":"false",
                          "free_size":"32689058816",
                          "size_on_disk":"50147098624",
                          "cg":"",
                          "pool":"example_pool",
                          "system":"gen4d-60b",
                          "unique_stored_data":"5000000000",
                          "size":"50000000000",
                          "domain":"/Global Space/",
                          "name":"example_volume",
                          "domain_ref":"/hsm/v5/:gen4d-60b/domains/:/Global Space/"
}
```


Note: The **perf_class_ref** parameter is applicable only to objects limited by QoS.

Volume snapshot resource definitions

The volume snapshot resource formats that are returned by the REST API for XIV Gen3 and Spectrum Accelerate are as follows:

```
{
       "snapshot": {
                   "creator": " "
                   "size_on_disk": "154000000000"
                   "delete_priority": "1"
                   "cg": " "
                   "pool": "testt"
                   "source_ref": "/hsm/v5/:gen3p1-25b/volumes/:vol_om"
                   "source": "vol_om"
                   "wwn": "0017380027290036"
                   "pool_ref": "/hsm/v5/:gen3p1-25b/pools/:testt"
                   "ref": "/hsm/v5/:gen3p1-25b/snapshots/:last-replicated-vol_om"
                   "system": "gen3p1-25b"
                   "size": "150000000000"
                   "sg": " "
                   "domain": "/Global Space/"
                   "name": "last-replicated-vol_om"
                   "id": "71c14d00036"
                   "locked": "true",
                   }
}
```

The volume snapshot resource formats that are returned by the REST API for FlashSystem A9000 and A9000R are as follows:

```
{
       "snapshot": {
                   "creator": "xiv_msms"
                   "size_on_disk": "10000000000"
                   "delete_priority": "1"
                   "cg": " "
                   "pool": "alex_test"
                   "source_ref": "/hsm/v5/:gen4d-27c/volumes/:vol_alex_03"
                   "source": "vol_alex_03"
                   "wwn": "6001738CFC9035C600000000007971E1"
                   "pool_ref": "/hsm/v5/:gen4d-27c/pools/:alex_test"
                   "ref": "/hsm/v5/:gen4d-27c/
snapshots/:vol_alex_03.automated_snapshot_1_20170822_0000"
                   "system": "gen4d-27c"
                   "size": "10000000000"
                   "sg": " "
                   "domain": "/Global Space/"
                   "name": "vol_alex_03.automated_snapshot_1_20170822_0000"
                   "id": "3e02819202340"
                   "locked": "true"
                   }
)
```

Note: The **perf_class_ref** parameter is applicable only to objects limited by QoS.

Volume mapping resource definitions

The volume mapping resource formats that are returned by the REST API are as follows:

```
{
  "vol_map": {
          "ref": "/hsm/v5/:mn44/vol_maps/:host:host_rest-host-sa-def:rest-vol-rpnoio_001",
          "lun": "2",
          "host": "rest-host-sa-def",
          "cluster": "",
          "volume": "rest-vol-rpnoio_001",
          "id": "101215",
          "system": "mn44"
          }
}
```

Consistency group resource definitions

The consistency group resource formats that are returned by the REST API are as follows:

```
{
       "cg": {
             "ref": "/hsm/:mn44/cgs/:test-sg ",
             "id": "111111", "name": "test-cg",
             "pool": "pool1",
             "pool_ref": "/hsm/:mn44/pools/:pool1 ",
             "mirrored": "false" ,
             "used_capacity": "0",
             "volumes": [
                        {
                         "name": "rest-vol-cd-def-1",
                         "ref": "/hsm/v4/:mn44/volumes/:rest-vol-cd-def-1"
                        },
                        {
                         "name": "rest-vol-cd-def-2",
                         "ref": "/hsm/v4/:mn44/volumes/:rest-vol-cd-def-2"
                        }
                        ],
             "system": "mn44"
             }
}
```

Snapshot group resource definitions

The snapshot group resource formats that are returned by the REST API are as follows:

```
{
       "sg": {
             "id": "0",
             "ref": "/hsm/v4/:6010625a/snap_groups/:cg1.snap_group_00001",
             "system": "6010625a",
             "name": "cg1.snap_group_00001",
             "cg": "cg1",
             "cg_ref": "/hsm/v4/:6010625a/cgs/:cg1",
             "locked": "yes",
             "delete_priority": "1",
             "snapshots": [
                          {
                          "name": "cg1.snap_group_00001.vol2",
                          "ref": "/hsm/v4/:6010625a/snapshots/:cg1.snap_group_00001.vol2"
                           },
                          {
                          "name": "cg1.snap_group_00001.vol4",
                          "ref": "/hsm/v4/:6010625a/snapshots/:cg1.snap_group_00001.vol4"
                           },
                          {
                           "name": "cg1.snap_group_00001.vol3",
                           "ref": "/hsm/v4/:6010625a/snapshots/:cg1.snap_group_00001.vol3"
                           },
                          {
                           "name": "cg1.snap_group_00001.vol1",
                           "ref": "/hsm/v4/:6010625a/snapshots/:cg1.snap_group_00001.vol1"
                           }
                           ]
                }
}
```

Mirror resource definitions

The mirror resource formats that are returned by the REST API are as follows:

```
{
     "mirror": {
               "destination_ref": "/hsm/v4/:6010625a/cgs/:cg1-625",
               "sync_type": "sync_best_effort",
               "mirrored_object": "CG",
               "destination_system_ref": "/hsm/v4/systems/:6010625a",
               "connected": "yes",
               "source_ref": "/hsm/v4/:6010625b/cgs/:cg1-625b",
               "source_system_ref": "/hsm/v4/systems/:6010625b",
               "sync_state": "Consistent",
               "specified_rpo": "",
               "remote_rpo": "",
               "destination": "cg1-625",
               "id": "56a14e00000",
               "system": "6010625a",
               "source": " cg1-625b",
               "active": "yes",
               "destination_system": "6010625a",
               "source_system": "6010625b"
               }
}
```

Host resource definitions

The host resource formats that are returned by the REST API for XIV Gen3 and Spectrum Accelerate are as follows:

```
      "host": {

         {
             "cluster": " "
             "ref": "/hsm/v5/:gen3p1-25b/hosts/:host184"
             "system": "gen3p1-25b"
             "perf_class": " "
             "name": "host184"
             "domains":

                {
                    "domain": "ErezDomain"
                }

                {
                    "domain": "/Global Space/"
                }
             "id": "63714500002"
             "type": "default"
             "ports":

                 {
                     "type": "iscsi"
                     "name": "iqn.1994-05.com.redhat:7979112c186"
                     "ref": "/hsm/v5/:gen3p1-25b/host_ports
                     /:iscsi:host184:iqn.1994-05.com.redhat:7979112c186"
                 }
       }
```

The host resource formats that are returned by the REST API for FlashSystem A9000 and A9000R are as follows:

```
"host":

   {
       "cluster": " "
       "ref": "/hsm/v5/:gen4d-27c/hosts/:123"
       "system": "gen4d-27c"
       "perf_class": " "
       "name": "123"
       "domains":

          {
              "domain": "/Global Space/"
          }
       "id": "3dd6e18900015"
       "type": "default"
       "ports":

           {
               "type": "iscsi"
               "name": "172.16.4.179"
               "ref": "/hsm/v5/:gen4d-27c/host_ports
               /:iscsi:123:172.16.4.179"
           }

           {
               "type": "iscsi"
               "name": "9.151.154.237"
               "ref": "/hsm/v5/:gen4d-27c/host_ports
               /:iscsi:123:9.151.154.237"
           }
   }
```

Note: The **perf_class_ref** parameter is applicable only to objects limited by QoS.

Host port resource definitions

The host port resource formats that are returned by the REST API are as follows:

```
{
        "host_port": {
                     "ref": "/hsm/v5/:mn44/host_ports/:rest-host-sa-def:1234567890123001",
                     "host": "rest-host-sa-def",
                     "type": "fc",
                     "id": "101280",
                     "name": "1234567890123001",
                     "system": "mn44"
                     }
}
```

Cluster resource definitions

The cluster resource formats that are returned by the REST API are as follows:

```
{
           "cluster": {
                      "ref": "/hsm/v5/:mn44/clusters/:rest-cluster-def",
                      "type": "standard",
                      "id": "101239",
                      "name": "rest-cluster-def",
                      "hosts": [
                               {
                                "name": "rest-host-cd-def-1",
                                "ref": "/hsm/v5/:mn44/hosts/:rest-host-cd-def-1"
                                },
                               {
                                "name": "rest-host-cd-def-2",
                                "ref": "/hsm/v5/:mn44/hosts/:rest-host-cd-def-2"
                                }
                                ],
                     "system": "mn44",
                     "domains": [
                                {
                                 "domain": "domain1"
                                },
                                {
                                 "domain": "domain2"
                                }
                                ]
                     }
}
```

Performance class resource definitions

The performance class resource formats that are returned by the REST API are as follows:

```
{
     "performanceClass": {
             "ref":"/hsm/v4/:Gen4D64a/perf_classes/:ClassQoS",
             "system": "gen4d-54c",
             "name": "AnotherQoS",
             "interface_iops_limit": 0,
             "interface_bw_limit": 500,
             "id": "10d61720000b",
          "type": SHARED
            }
}
```

Event resource definitions

The event resource formats that are returned by the REST API are as follows:

```
{
"event": {
      "ref": "/hsm/:mn44/events/:111111 ",
      "id": "111111", "code": "USER_LOGIN_HAS_SUCCEEDED",
      "severity": "Informational",
      "timestamp": "1234567890",
      "description": "User ‘admin’ from IP ‘X.X.X.X’ successfully logged into the system.",
      "system": "mn44"
         }
}
```

Chapter 7. Error handling

Each request to the REST API receives a response with two types of status codes:

- **HTTP (transport) status code:** A quick indication of whether or not the request succeeded. Some automation scripts may not parse the full response in case of error, so it is helpful to have a quick indication at the HTTP level.
- **server status code:** Provides a more detailed application level status to allow troubleshooting in case of errors.

The following topics are covered in this chapter:

HTTP status codes

The following tables lists the HTTP status codes and messages returned by the REST API:

Table 1. HTTP status codes and messages

HTTP status code	HTTP message	Description
200	OK	The request succeeded. The response contains either a list of resources (in response to a query) or is empty.
400	Bad Request	The request contains invalid scope keywords or invalid URL parameters. Examples of invalid URL parameters are: non-numeric values, redundant parameters, etc.
401	Unauthorized	The user credentials provided are not valid.
404	Not Found	The request addresses non-existing resource(s). The server status code contains more details about what resource was not found.
500	Internal Server Error	The request is valid, but some application level error occurred. See server status for more information.

Server status codes

The following tables lists the server status codes and messages returned by the REST API:

Table 2. Server status codes, correlated to HTTP status codes

HTTP status code	Server status code	Server message	Description
200	0	OK	The request succeeded. The response contains either a list of resources (in response to a query) or is empty.
500	1	Robot authentication failed	The server cannot connect to a supported storage system with robot user credentials. Most likely, the robot user password is not set on the server.
401	2	Unauthorized	Invalid user credentials were supplied for a query or update operation.
500	3	No systems connected	All supported storage systems that are addressed in the request are disconnected.
404	4	Resource named X of type Y not found	The query request addressed a non-existent resource. The server message contains the missing resource name and type. This is useful for troubleshooting complex requests, such as mapping of the system's host.
500	5	Request failed	The request execution failed. The **failed_systems** part contains the reason.
400	6	Invalid Request. <Reason>	The request is invalid. The message contains the exact reason.
500	8	No managed systems configured	No supported storage systems are managed by the server.
500	9	Server general error	Some general error occurred during processing.

Failed system status codes

When performing multi-system queries, some supported storage systems may be in an error state (such as `disconnected`, `not authenticated`, etc.). This indicates to the user that the query response does not contain resources from these systems. To address this situation, there is the **failed_systems** field in the response. The field is a list that contains systems that are in error, according to the following table:

Table 3. Failed system status codes

Failed system status code	Message	Description
1	Robot user authentication failed	Robot user credentials are not defined on the system, or the robot password is not set on the server.
2	Unauthorized	Invalid user credentials were supplied.
3	System is disconnected	The server cannot connect to the specific supported storage system.
4	The system is not managed by the server	The specific supported storage system is not managed by the server.
5	Request failed. <Reason>	The request failed. Reason contains CLI command response + error code.
6	System is suspended	The specific storage system is suspended.

If there is a multi-system request when all of the systems are inaccessible, the following rules apply:

- If all systems are in an `Unauthorized` state, the response is `HTTP 401 , server status 2`.
- If all systems are in a `Robot user authentication failed state`, the response is `HTTP 500, server status 1`.
- For all other cases, the response is `HTTP 500, server status 3`.

Chapter 8. Security

The REST API protocol does not have any built-in security features, so it strongly relies on the transport protocol (HTTP) to provide security. Specifically, the following HTTP features are used:

- **HTTP/S:** Provides an encrypted HTTP channel between client and server.
- **HTTP basic authentication:** Used for carrying client provided credentials to the server / supported storage systems.

Notices

This information was developed for products and services offered in the U.S.A. This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Almaden Research
650 Harry Road
Bldg 80, D3-304, Department 277

San Jose, CA 95120-6099
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Copyright and trademark information website (www.ibm.com/legal/us/en/copytrade.shtml).

Other product and service names might be trademarks of IBM or other companies.

Printed in USA

SC27-6440-07