

# Hitachi Ops Center Protector

7.10

---

## Rest API User Guide

This guide is intended for those who wish to develop a script or application which utilizes Hitachi Ops Center Protector

© 2016, 2024 Hitachi Vantara LLC. All rights reserved

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including copying and recording, or stored in a database or retrieval system for commercial purposes without the express written permission of Hitachi, Ltd., Hitachi Vantara, Ltd., or Hitachi Vantara Corporation (collectively "Hitachi"). Licensee may make copies of the Materials provided that any such copy is: (i) created as an essential step in utilization of the Software as licensed and is used in no other manner; or (ii) used for archival purposes. Licensee may not make any other copies of the Materials. "Materials" mean text, data, photographs, graphics, audio, video and documents.

Hitachi reserves the right to make changes to this Material at any time without notice and assumes no responsibility for its use. The Materials contain the most current information available at the time of publication.

Some of the features described in the Materials might not be currently available. Refer to the most recent product announcement for information about feature and product availability, or contact Hitachi Vantara LLC at <https://support.hitachivantara.com/en-us/contact-us.html>.

**Notice:** Hitachi products and services can be ordered only under the terms and conditions of the applicable Hitachi agreements. The use of Hitachi products is governed by the terms of your agreements with Hitachi Vantara LLC.

By using this software, you agree that you are responsible for:

1. Acquiring the relevant consents as may be required under local privacy laws or otherwise from authorized employees and other individuals; and
2. Verifying that your data continues to be held, retrieved, deleted, or otherwise processed in accordance with relevant laws.

**Notice on Export Controls.** The technical data and technology inherent in this Document may be subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Reader agrees to comply strictly with all such regulations and acknowledges that Reader has the responsibility to obtain licenses to export, re-export, or import the Document and any Compliant Products.

Hitachi and Lumada are trademarks or registered trademarks of Hitachi, Ltd., in the United States and other countries.

AlX, DB2, DS6000, DS8000, Enterprise Storage Server, eServer, FICON, FlashCopy, GDPS, HyperSwap, IBM, OS/390, PowerHA, PowerPC, S/390, System z9, System z10, Tivoli, z/OS, z9, z10, z13, z14, z15, z16, z/VM, and z/VSE are registered trademarks or trademarks of International Business Machines Corporation.

Active Directory, ActiveX, Bing, Excel, Hyper-V, Internet Explorer, the Internet Explorer logo, Microsoft, Microsoft Edge, the Microsoft corporate logo, the Microsoft Edge logo, MS-DOS, Outlook, PowerPoint, SharePoint, Silverlight, SmartScreen, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, the Windows logo, Windows Azure, Windows PowerShell, Windows Server, the Windows start button, and Windows Vista are registered trademarks or trademarks of Microsoft Corporation. Microsoft product screen shots are reprinted with permission from Microsoft Corporation.

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

Copyright and license information for third-party and open source software used in Hitachi Vantara products can be found in the product documentation, at <https://www.hitachivantara.com/en-us/company/legal.html>.

---

# Contents

Preface.....	5
Software version.....	5
Intended audience.....	5
Related documents.....	5
Document conventions.....	6
Conventions for storage capacity values.....	7
Accessing product documentation.....	8
Getting help.....	8
Comments.....	8
Chapter 1: Introduction to the REST API.....	9
Chapter 2: REST URL root.....	10
Chapter 3: HTTP methods and responses.....	11
Chapter 4: Request and Response formats.....	12
Chapter 5: User Session and Cookies.....	13
Chapter 6: API Concepts.....	14
Resources.....	14
Services.....	14
Handlers.....	15
Actions.....	15
Collections.....	15
Filtering.....	15
Sorting.....	16
Fields.....	16
Count and Offset.....	16
Dates and Times.....	17
URL Encoding.....	17
Job responses.....	17
Error responses.....	17
Templating concept.....	17

Chapter 7: API Reference Guide.....	19
Chapter 8: Examples.....	20
Log In.....	21
List all Nodes.....	21
Create Repository Node on the Master.....	23
Create Policy.....	25
Edit Policy.....	29
Create Data Flow.....	33
Compile and Activate Data Flow.....	37
Trigger Backups.....	38
List Jobs.....	40
Restore a backup to an alternate location.....	42
View RPO Report.....	45
Delete Backup.....	47
Log Out.....	47
Checking the Application version.....	47
Glossary.....	49
Conventions for storage capacity values.....	53

---

# Preface

This document describes how to develop scripts and applications for Hitachi Ops Center Protector.

## Software version

This document revision applies to Ops Center Protector version 7.10. Please refer to the accompanying Release Notes for information on what's changed in this release.

## Intended audience

This guide is intended for those who wish to develop a script or application which utilizes Hitachi Ops Center Protector

## Related documents

Main product guides:

- *Hitachi Ops Center Protector Software Release Notes.*
- *Hitachi Ops Center Protector Quick Start Guide.*
- *Hitachi Ops Center Protector User's Guide.*
- *Hitachi Ops Center Protector Oracle Application Guide.*
- *Hitachi Ops Center Protector VMware Application Guide.*
- *Hitachi Ops Center Protector Hyper-V Application Guide.*
- *Hitachi Ops Center Protector Microsoft SQL Application Guide*

Programming guides:







- *Hitachi Ops Center Protector REST API User Guide.*
- *Hitachi Ops Center Protector REST API Reference Guide.*
- *Hitachi Ops Center Protector REST API Change Log.*

## Document conventions

This document uses the following typographic conventions:

Convention	Description
<b>Bold</b>	<ul style="list-style-type: none"> <li>Indicates text in a window, including window titles, menus, menu options, buttons, fields, and labels. Example: Click <b>OK</b>.</li> <li>Indicates emphasized words in list items.</li> </ul>
<i>Italic</i>	<ul style="list-style-type: none"> <li>Indicates a document title or emphasized words in text.</li> <li>Indicates a variable, which is a placeholder for actual text provided by the user or for output by the system. Example: <pre>pairedisplay -g group</pre> (For exceptions to this convention for variables, see the entry for angle brackets.)</li> </ul>
Monospace	Indicates text that is displayed on screen or entered by the user. Example: <code>pairedisplay -g oradb</code>
< > angle brackets	<p>Indicates variables in the following scenarios:</p> <ul style="list-style-type: none"> <li>Variables are not clearly separated from the surrounding text or from other variables. Example: <pre>Status-&lt;report-name&gt;&lt;file-version&gt;.csv</pre></li> <li>Variables in headings.</li> </ul>
[ ] square brackets	Indicates optional values. Example: [ a   b ] indicates that you can choose a, b, or nothing.
{ } braces	Indicates required or expected values. Example: { a   b } indicates that you must choose either a or b.
vertical bar	<p>Indicates that you have a choice between two or more options or arguments. Examples:</p> <p>[ a   b ] indicates that you can choose a, b, or nothing.</p> <p>{ a   b } indicates that you must choose either a or b.</p>

This document uses the following icons to draw attention to information:

Icon	Label	Description
	Note	Calls attention to additional information.
	Tip	Provides helpful information, guidelines, or suggestions for performing tasks more effectively.
	Important	Highlights information that is essential to the completion of a task.
	Caution	Warns the user of adverse conditions and/or consequences (for example, disruptive operations, data loss, or a system crash).
	CAUTION	Warns the user of a hazardous situation that, if not avoided, could result in major or minor injury.
	WARNING	Warns the user of a hazardous situation which, if not avoided, could result in death or serious injury.

## Conventions for storage capacity values

Physical storage capacity values (for example, disk drive capacity) are calculated based on the following values:

Physical capacity unit	Value
1 kilobyte (KB)	1,000 ( $10^3$ ) bytes
1 megabyte (MB)	1,000 KB or $1,000^2$ bytes
1 gigabyte (GB)	1,000 MB or $1,000^3$ bytes
1 terabyte (TB)	1,000 GB or $1,000^4$ bytes
1 petabyte (PB)	1,000 TB or $1,000^5$ bytes
1 exabyte (EB)	1,000 PB or $1,000^6$ bytes

Logical capacity values (for example, logical device capacity, cache memory capacity) are calculated based on the following values:

Logical capacity unit	Value
1 block	512 bytes

Logical capacity unit	Value
1 cylinder	Mainframe: 870 KB Open-systems: <ul style="list-style-type: none"> <li>OPEN-V: 960 KB</li> <li>Others: 720 KB</li> </ul>
1 KB	1,024 (2 <sup>10</sup> ) bytes
1 MB	1,024 KB or 1,024 <sup>2</sup> bytes
1 GB	1,024 MB or 1,024 <sup>3</sup> bytes
1 TB	1,024 GB or 1,024 <sup>4</sup> bytes
1 PB	1,024 TB or 1,024 <sup>5</sup> bytes
1 EB	1,024 PB or 1,024 <sup>6</sup> bytes

## Accessing product documentation

Product user documentation is available on: <https://docs.hitachivantara.com>. Check this site for the most current documentation, including important updates that may have been made after the release of the product.

## Getting help

The [Hitachi Vantara Support Website](https://support.hitachivantara.com/en_us/contact-us.html) is the destination for technical support of products and solutions sold by Hitachi Vantara. To contact technical support, log on to the Hitachi Vantara Support Website for contact information: [https://support.hitachivantara.com/en\\_us/contact-us.html](https://support.hitachivantara.com/en_us/contact-us.html).

[Hitachi Vantara Community](https://community.hitachivantara.com) is a global online community for Hitachi Vantara customers, partners, independent software vendors, employees, and prospects. It is the destination to get answers, discover insights, and make connections. **Join the conversation today!** Go to [community.hitachivantara.com](https://community.hitachivantara.com), register, and complete your profile.

## Comments

Please send comments to [doc.feedback@hitachivantara.com](mailto:doc.feedback@hitachivantara.com). Include the document title and number, including the revision level (for example, -07), and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Vantara LLC.

**Thank you!**

---

## Chapter 1: Introduction to the REST API

Hitachi Ops Center Protector exposes a Representational State Transfer (REST) application programming interface (API) solely over HTTPS which can be accessed using any scripting or programming language which supports making HTTPS requests.

The API requests are governed by Role Based Access Control (RBAC) which itself can be configured using the API (or via the application graphical user interface (GUI)).

Once the user is authenticated (by logging in with valid credentials) they are able to perform tasks in the roles for which they are authorized.

The user interface solely uses the REST API so 100% of the functionality available in the user interface is available through the REST API.

---

## Chapter 2: REST URL root

The Hitachi Ops Center Protector REST API is served from the master node and can either be accessed locally or remotely. The root URL is constructed from a number of parts as follows:

*https://<host>[:<port>]/API/<version>/*

- host - The IP or DNS resolvable name of the master machine. Or "localhost" if accessing the API from the master machine directly.
- port - The port on which the UI / API is being servers. By default this is 443 but can be changed during master installation.
- version - The application major and minor version, for example, 7.1. This provides the ability for scripts or applications to use a consistent version of the API and support for backwards compatibility with new releases.

---

## Chapter 3: HTTP methods and responses

Requests to the REST API use the common HTTP/1.1 methods. The methods allow users to read (GET), update (PUT), create (POST) and delete (DELETE) resources.

Responses also use the HTTP/1.1 status codes. Important status codes are as follows:

- 200 OK – The request has succeeded.
- 201 Created – The request has been fulfilled and the resource created.
- 202 Accepted – The request has been accepted but the processing is not complete. The response normally includes an ID for a Job which can be queried to get information about the progress of the requested task.
- 400 Bad Request – The request could not be understood the client should not repeat the request without modifications.
- 401 Unauthorized – The request requires authentication.
- 404 Not Found – The requested handler or resource was not found.
- 500 Internal Server Error – The server encountered an unexpected condition which has prevented it from fulfilling the request.

Further information regarding HTTP/1.1 and its supported methods and status codes can be found in the Hypertext Transfer Protocol specification <https://tools.ietf.org/html/rfc2616>.

---

## Chapter 4: Request and Response formats

Requests can be made with URL query data and/or JSON body data depending on the resource and method being used. Request data should be provided in JSON and response data is always returned as JSON.

---

## Chapter 5: User Session and Cookies

Most requests require a valid session to perform the request. To get a valid session the user must first log in using their credentials. The response to a valid log in request contains the “Set-Cookie” HTTP header which contains a Session cookie. This session cookie must be provided to all subsequent requests in the HTTP “Cookie” header.

The session is valid until the user logs out, or there has been two hours of inactivity (unless otherwise configured).

---

## Chapter 6: API Concepts

### Resources

Almost all interactions using the API are done against resources. Resources typically represent an object which can be retrieved, created, modified or deleted, as permitted by the resource handler and user permissions.

All resources are referenced by their pluralized type name, e.g., Nodes instead of Node, even when retrieving a single resource.

Resources can be retrieved as a list which can include filtering, sorting and field query data or as an individual object by appending the resource ID to the URL. When retrieving a list of resources the response will include a “pageInfo” object. This contains properties describing the current page count and offset, the total amount of resources which match the query and whether they have all been returned within the response, for example:

```
"pageInfo": {  
  "offset": 0,  
  "count": 10  
  "totalCount": 10,  
  "end": true  
}
```

To retrieve all Nodes (which the user has access to):

<https://localhost/API/:version/master/NodeManager/objects/Nodes>

Or to retrieve a single Node:

<https://localhost/API/:version/master/NodeManager/objects/Nodes/:ID>

### Services

Services allow actions to be performed on handlers where the action does not relate to an individual resource. Service endpoints only allow action requests and can either change the state of many resources or none at all.

An example of a service action is the one used for logging in:

<https://localhost/API/:version/master/UIController/services/Sessions/actions/login/invoke>

## Handlers

Resources are exposed by handlers. Each handler is responsible for the management of a number of resources. Most handlers are available exclusively on the master node, however some handlers are available only on client nodes and some handlers are available on all nodes.

To access a resource provided by a handler, the handler address (node) and name are required. The API does not handle requests which are not directed to a handler.

## Actions

Action requests can be performed on resources or services and are used to perform specific tasks.

Action requests often modify multiple resources or the entire system state. In a few situations no resources are modified but the response contains data otherwise unavailable for the associated resource.

Actions are useful in describing a task being performed against a resource. For example to authorize a client node the following action request would be made:

<https://localhost/API/:version/master/NodeManager/objects/Nodes/:id/actions/authorize/>  
[invoke](#)

This allows for a more explicit request and response as opposed to simply making an update request against the node's "authorized" property.

## Collections

Collections represent resources which are contained within other resources.

For example, Nodes within a Node Group:

[https://localhost/API/:version/master/NodeManager/objects/NodeGroups/:id/collections/](https://localhost/API/:version/master/NodeManager/objects/NodeGroups/:id/collections/Nodes)  
[Nodes](#)

Or Files and Directories within a FileSystemItem:

[https://localhost/API/:version/master/FileSystemBrowser/objects/FileSystemItems/:id/](https://localhost/API/:version/master/FileSystemBrowser/objects/FileSystemItems/:id/collections/FileSystemItems)  
[collections/FileSystemItems](#)

Some collections allow the user to add or remove items from within it. Others are fixed and cannot be changed. Depending on the resource the collection may be managed by adding and removing a list of resource IDs or by adding and removing complete objects.

## Filtering

When retrieving a list of resources it is possible to apply a filter using the "query" parameter within the URL.

It is possible to filter on properties deep within the resource structure by using ‘dot notation’. For example, filtering for authorized nodes can be achieved with the following query:

<https://localhost/API/:version/master/NodeManager/objects/Nodes/?query=stateInfo.authorized=true>

The query syntax supports AND, OR and grouping (using brackets) for complex queries.



**Note:** It is only possible to filter on supported properties. Please refer to the API Reference Guide for further information.

## Sorting

When retrieving a list of resources it is possible to apply a sort order using the “order-by” parameter within the URL. This will order the data by a single property either ascending (ASC) or descending (DESC). The order keyword is required and failure to provide this will result in an error stating:

“Invalid order-by parameter. Invalid format”.

For example, retrieving the list of nodes ordered alphabetically by name:

<https://localhost/API/:version/master/NodeManager/objects/Nodes/?order-by=name+ASC>



**Note:** It is only possible to sort on a single supported property within any one request. Please refer to the API Reference Guide for further information.

## Fields

When retrieving a resource or list of resources it is possible to request a subset of properties from the resources using the “fields” parameter within the URL. Multiple fields can be specified by comma separating the field names.

For example, retrieving only the name of all nodes:

<https://localhost/API/:version/master/NodeManager/objects/Nodes/?fields=name>



**Note:** Not all handlers allow retrieving specific resources properties from either a list or individual item requests. Please refer to the API Reference Guide for further information.

## Count and Offset

When retrieving a list of resources it is possible to limit the amount of resources returned by using the “count” and “offset” parameters within the URL. By default resources are returned in an arbitrary order so it is often useful to also provide an “order-by” parameter.

It is possible to request a list of resources with both count and offset of zero. This will return the pageInfo object containing the total number of resources which matched the query.

For example, retrieving the second page of fifty nodes:

<https://localhost:/API/version/master/NodeManager/objects/Nodes/?count=50&offset=50>

## Dates and Times

All dates and times are returned in ISO-8601 format with a zero UTC offset. In most cases the dates and times should end in a 'Z' to indicate this.

## URL Encoding

All values after the URL host name and port, but excluding URL parameter names should be percentencoded. This includes target node, and query parameters.

For example, the node ID "Master@BT-XRTQWX-RRWTRT-SSQWUW-RTRXRQ[0-1-1]" should be encoded into the URL as "Mater%40BT-XRTQWX-RRWTRT-SSQWUW-RTRXRQ[0-1-1]".

For more details can be found on the URL specification: <https://url.spec.whatwg.org/#percentencoded-bytes>

## Job responses

Asynchronous processes are exposed as Jobs. A job can be created automatically by the system or by a request made by the user.

When a request is made to the API which results in a job being created the HTTP response code will be 201 Accepted and the body of the response contains the ID of the created job.

In some situations the Job may not be immediately available from the JobStatusHandler so any attempt to retrieve it should allow the request to be retried.

## Error responses

Invalid requests will receive a JSON error response as well as the appropriate HTTP status code.

Error responses contain an error message detailing the error along with the handler on which the error was generated.

## Templating concept

Node, policy, data flow and schedule resources exposed by the API are built up of a number of smaller objects that result in a complex object with many required and optional attributes and variable depth.

Creating these resources manually can be difficult and prone to error. To aid in the creation of these resources it is advised that the user create a template within the GUI and use these templates to create specialized versions of the resources for the specific use cases.

The templates would then be used as the framework to creating new resources based on the template.

For example a policy could be created containing a path classification and backup operation with an eight hour RPO or a specific schedule. The path classification would simply contain a placeholder value. The API can then be used to retrieve the resource, modify the path classification values and save the policy as a new resource. This allows the operation settings to be copied and applied to a new policy without having to redefine all of its properties.

---

## Chapter 7: API Reference Guide

The API Reference guide can be used to view the descriptions, properties and URL details of resources. To access the command syntax for all the APIs, go to the Help menu in the UI, select Online Help, and then select API Reference Guide.

Resources consist of properties. Each property can be a basic data type (string, number, boolean), an enumeration, or an object. Each property can be an array of that data.

Enumerations are sent and received as a string. The API reference guide details all possible values of an enumeration and a relevant description.

Data classes represent basic objects which have properties, but cannot be retrieved, created, updated or deleted directly and only exist as a property of a parent resource.

---

## Chapter 8: Examples

In the following examples the master node is called “demomaster”.

The curl CLI tool is used for all examples to clearly indicate the minimum input required to make the request.

The curl argument ‘—insecure’ is used to skip SSL certificate checks. It is recommended that the master is configured with a valid SSL certificate.

The examples use the curl “cookie-jar” to maintain the user session and provide the session ID for each request.

In some examples JSON data must be sent with the request. To maintain clarity the examples use the curl file reference feature, thus data beginning with @ reference a file within the local working directory.

The examples assumes that you have installed Hitachi Ops Center Protector and that a valid license has been applied. Please refer to the application user guide for details on how to install and activate a license if required.

The examples form the minimum steps required to create, perform and validate a simple host based backup to a local Repository store. The process does not conform to the best practices or recommendations in doing host based backup only to provide a clear purpose to the collection of examples.

The following steps are included in this process:

- Log In
- List Nodes
- Create a Repository Node on the Master
- Create a Policy
- Edit a Policy
- Create a Data Flow
- Compile and Activate a Data Flow
- Trigger a Backup
- List Jobs
- Restore to an alternative location
- View the RPO Report
- Log Out

## Log In

Log in to create the user session. Log in can be done using data formatted as “application/x-www-form-urlencoded” or “application/json”. In this example form-urlencoded data is used.

### Log In

```
curl --insecure --cookie-jar demosection --data  
"username=user&password=password&space=master"  
"https://demomaster/API/7.1/master/UIController/services/Users/actions/log  
in/invoke" -silent
```

## List all Nodes

Nodes are a key resources within Ops Center Protector. Generally any data protection operation is made from and to Nodes. The Master node is the only node available at the beginning of the examples and can be received with a GET request to the NodeManager handler.

### List all Nodes

```
curl --insecure --cookie demosection  
https://demomaster/API/7.1/master/NodeManager/objects/Nodes
```

**List all Nodes Response**

```

{
  "node": [
    {
      "agentless": false,
      "filterAvailable": false,
      "id": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
      "master": true,
      "name": "demomaster",
      "networkDetails": {
        "ipAddresses": [
          "127.0.0.1"
        ]
      },
      "nodeAttributes": [
      ],
      "osDetails": {
        "architecture": "x64",
        "chassis": "Virtual",
        "os": "eOS_WINDOWS",
        "osLevel": "Server",
        "osVariant": "Windows 2016",
        "osVersion": "Standard"
      },
      "resourceId": "4294967297",
      "stateInfo": {
        "accessible": true,
        "activeRules": 0,
        "authorized": true,
        "connected": true,
        "hasRules": false,
        "hubState": "Connected",
        "upToDate": true
      },
      "type": "OSHost",
      "userTags": [
      ],
      "version": "7.1.0.84083-R7.1"
    }
  ],
  "pageInfo": {
    "end": true,
    "totalCount": 1
  }
}

```

## Create Repository Node on the Master

To create a Repository Node on the Master the Master Node ID is required. Building on the previous request, get the node list with a query and field parameters. The “query” ensures only the Master Node is returned and “fields” indicates that the response should only include the requested fields.

The response then contains the Master node object with its ID field and page information.

A Repository Node is created on an OS Host node and the host node ID is required for the creation request. Using the Master Node ID received in the previous example the Repository Node is now created using the Master Node as its host (known as the proxy node). Creation request typically use the HTTP POST method (curl automatically applies the POST method when data is added to the request).

Repository Node creation is done immediately and the response contains the created Node details.

### List only the Master node and request just the Node ID

```
curl --insecure --cookie demoseession
"https://demomaster/API/7.1/master/NodeManager/objects/Nodes?query=(master
=true)&fields=id"
```

### List only the Master node and request just the Node ID Response

```
{
  "node": [
    {
      "id": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]"
    }
  ],
  "pageInfo": {
    "end": true,
    "totalCount": 1
  }
}
```

### Create the Repository on the Master Node

```
curl --insecure --cookie demoseession
"https://demomaster/API/7.1/master/RepositoryNodeHandler/objects/Repositor
ies" --data @createrepository.json --header "Content-Type:
application/json"
```

**Create the Repository on the Master Node Response**

```
{
  "blockSize": "e16K",
  "capacityCriticalLevel": 95,
  "capacityWarningLevel": 85,
  "cloudOptimized": false,
  "deduplication": false,
  "encrypted": false,
  "generation": "eREPOSITORY_GENERATION_2",
  "id": "RepositoryHP@00-25FAAB-3B94CF-459595-78584C[1-1-8D]",
  "isMounted": false,
  "name": "RepositoryHP",
  "paths": {
    "checksums": "F:\\RepositoryHP\\repository\\streams",
    "data": "F:\\RepositoryHP\\repository\\streams",
    "metadata": "F:\\RepositoryHP\\repository\\metadata",
    "metadataStreams":
"F:\\RepositoryHP\\repository\\metadatastreams",
    "root": "F:\\RepositoryHP\\"
  },
  "proxyNodeId": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
  "timestamp": "2020-05-11T16:20:02",
  "userTags": [
  ]
}
```

**Contents of 'createrepository.json'**

```
{
  "blockSize": "e16K",
  "capacityCriticalLevel": 95,
  "capacityWarningLevel": 85,
  "cloudOptimized": false,
  "deduplication": false,
  "encrypted": false,
  "generation": "eREPOSITORY_GENERATION_2",
  "import": false,
  "name": "RepositoryHP",
  "nodeType": "Repository",
  "paths": {
    "checksums": "",
    "data": "",
    "metadata": "",
    "root": "F:"
  },
  "proxyNodeId": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
  "resourceGroup": null
}
```

## Create Policy

A Policy is used to define the data classification and the operation to apply to it. Here the “Path” classification is used to specify the path “C:\testdata” which using the Backup operation will be backed up with an RPO of eight hours.

Policy creation is done immediately and the response contains the created policy details.

### Create Policy

```
curl --insecure --cookie demoseession  
"https://demomaster/API/7.1/master/PolicyHandler/objects/Policies" --data  
@createpolicy.json --header "Content-Type: application/json"
```

**Create Policy Response**

```

{
  "classificationChains": [
    {
      "items": [
        {
          "id": 1,
          "path": {
            "exclude": "",
            "include": "C:\\\\testdata",
            "options": {
              "fastIncremental": true,
              "fineChangeDetection": false
            }
          },
          "propertyName": "path",
          "typeName": "path"
        }
      ]
    }
  ],
  "createdDate": "2020-05-11T15:20:02Z",
  "description": "",
  "id": "141eb1403015408aa3e6a77e2aelab22",
  "modifiedDate": "2020-05-11T15:20:02Z",
  "name": "demopolicy",
  "nextClassificationId": 2,
  "nextOperationId": 2,
  "operationChains": [
    {
      "items": [
        {
          "backup": {
            "label": "Backup",
            "recoveryPointObjective": {
              "period": 8,
              "units": "eRPO_HOURS"
            },
            "recoveryTimeObjective": {
              "incrementalTapeBackup": false
            },
            "retention": {
              "period": 6,
              "units": "eRETENTION_MONTHS"
            },
            "run": {
              "mode": "eRUN_ANYTIME"
            },
            "sourceOption": {
              "postScriptEnabled": false,

```

```

        "preScriptEnabled": false,
        "quiesceApplications": true
    },
    "type": {
        "mode": "eTRANSFER_ASYNCHRONOUS_JOURNALED"
    }
},
"id": 1,
"propertyName": "backup",
"typeName": "Backup",
"userTags": [
]
}
]
}
],
"permissions": [
{
    "access": "eA_READWRITE",
    "propertyType": "pt_user",
    "pt_user": {
        "upn": "Administrator@master"
    }
}
],
"userTags": [
],
"version": 1
}

```

## Contents of 'createpolicy.json'

```

{
  "classificationChains": [
    {
      "items": [
        {
          "category": "Data",
          "displayName": "Path",
          "normalizedTypeName": "path",
          "path": {
            "exclude": "",
            "include": "C:\\\\testdata",
            "options": {
              "fastIncremental": true,
              "fineChangeDetection": false
            }
          },
          "propertyName": "path",
          "typeName": "path"
        }
      ]
    }
  ],
  "description": "",
  "name": "demopolicy",
  "operationChains": [
    {
      "items": [
        {
          "backup": {
            "label": "Backup",
            "recoveryPointObjective": {
              "period": 8,
              "units": "eRPO_HOURS"
            },
            "recoveryTimeObjective": {
              "incrementalTapeBackup": false
            },
            "retention": {
              "period": 6,
              "units": "eRETENTION_MONTHS"
            },
            "run": {
              "mode": "eRUN_ANYTIME",
              "scheduleName": ""
            },
            "sourceOption": {
              "postScriptEnabled": false,
              "preScriptEnabled": false,
              "quiesceApplications": true
            }
          }
        }
      ]
    }
  ]
}

```

```

    },
    "type": {
      "mode": "eTRANSFER_ASYNCHRONOUS_JOURNALED"
    }
  },
  "category": "Standard",
  "displayName": "Backup",
  "normalizedTypeName": "backup",
  "propertyName": "backup",
  "typeName": "Backup"
}
]
}
]
}

```

## Edit Policy

Editing a resource, in this case a Policy, is very similar to creation. However the HTTP PUT method is used and the URL must point to a specific resource, rather than the resource list.

This example changes the Backup operation RPO from eight hours to one hour.

### Edit Policy

```

curl --insecure --cookie demosection
"https://demomaster/API/7.1/master/PolicyHandler/objects/Policies/141eb140
3015408aa3e6a77e2aelab22" --data @editpolicy.json --header "Content-Type:
application/json" --request PUT

```

**Edit Policy Response**

```

{
  "classificationChains": [
    {
      "items": [
        {
          "id": 1,
          "path": {
            "exclude": "",
            "include": "C:\\testdata",
            "options": {
              "fastIncremental": true,
              "fineChangeDetection": false
            }
          },
          "propertyName": "path",
          "typeName": "path"
        }
      ]
    }
  ],
  "createdDate": "2020-05-11T15:20:02Z",
  "description": "",
  "id": "141eb1403015408aa3e6a77e2ae1ab22",
  "modifiedDate": "2020-05-11T15:20:03Z",
  "name": "demopolicy",
  "nextClassificationId": 2,
  "nextOperationId": 2,
  "operationChains": [
    {
      "items": [
        {
          "backup": {
            "label": "Backup",
            "recoveryPointObjective": {
              "period": 1,
              "units": "eRPO_HOURS"
            },
            "recoveryTimeObjective": {
              "incrementalTapeBackup": false
            },
            "retention": {
              "period": 6,
              "units": "eRETENTION_MONTHS"
            },
            "run": {
              "mode": "eRUN_ANYTIME"
            },
            "sourceOption": {
              "postScriptEnabled": false,

```

```

        "preScriptEnabled": false,
        "quiesceApplications": true
    },
    "type": {
        "mode": "eTRANSFER_ASYNCHRONOUS_JOURNALED"
    }
},
"id": 1,
"propertyName": "backup",
"typeName": "Backup",
"userTags": [
]
}
]
}
],
"permissions": [
{
    "access": "eA_READWRITE",
    "propertyType": "pt_user",
    "pt_user": {
        "upn": "Administrator@master"
    }
}
],
"userTags": [
],
"version": 2
}

```

## Contents of 'editpolicy.json'

```
{
  "classificationChains": [
    {
      "items": [
        {
          "id": 1,
          "path": {
            "exclude": "",
            "include": "C:\\\\testdata",
            "options": {
              "fastIncremental": true,
              "fineChangeDetection": false
            }
          },
          "propertyName": "path",
          "typeName": "path"
        }
      ]
    },
    {
      "createdDate": "2020-05-11T15:20:02Z",
      "description": "",
      "id": "141eb1403015408aa3e6a77e2ae1ab22",
      "modifiedDate": "2020-05-11T15:20:02Z",
      "name": "demopolicy",
      "nextClassificationId": 2,
      "nextOperationId": 2,
      "operationChains": [
        {
          "items": [
            {
              "backup": {
                "label": "Backup",
                "recoveryPointObjective": {
                  "period": 1,
                  "units": "eRPO_HOURS"
                },
                "recoveryTimeObjective": {
                  "incrementalTapeBackup": false
                },
                "retention": {
                  "period": 6,
                  "units": "eRETENTION_MONTHS"
                },
                "run": {
                  "mode": "eRUN_ANYTIME"
                },
                "sourceOption": {
                  "postScriptEnabled": false,

```

```

        "preScriptEnabled": false,
        "quiesceApplications": true
    },
    "type": {
        "mode": "eTRANSFER_ASYNCHRONOUS_JOURNALED"
    }
},
"id": 1,
"propertyName": "backup",
"typeName": "Backup",
"userTags": []
}
]
}
],
"permissions": [
    {
        "access": "eA_READWRITE",
        "propertyType": "pt_user",
        "pt_user": {
            "upn": "Administrator@master"
        }
    }
]
],
"userTags": [],
"version": 1
}

```

## Create Data Flow

A Data Flow is used to define from where the data will originate and to where the data will be sent.

The Policy created earlier will be applied to the Master Node which will send the data to the Repository Node via a batch mover.

The Master Node ID, Repository Node ID and Policy ID are required within the Data Flow.

### Create Data Flow

```

curl --insecure --cookie demosection
"https://demomaster/API/7.1/master/DataFlowHandler/objects/DataFlows" --
data @createdataflow.json --header "Content-Type: application/json"

```

**Create Data Flow Response**

```

{
  "createdDate": "2020-05-11T15:20:03Z",
  "data": {
    "compilerDirectives": "",
    "connections": [
      {
        "bandwidthSettings": {
          "hoursOfWeek": [
          ],
          "throttleDefaultSpeed": false
        },
        "destination": "166",
        "id": "168",
        "label": "",
        "netCompression": false,
        "routedPolicies": [
          "141eb1403015408aa3e6a77e2aelab22"
        ],
        "source": "167",
        "type": "eCT_BATCH"
      }
    ],
    "description": "",
    "name": "demodataflow",
    "nodes": [
      {
        "assignedPolicies": [
          {
            "id": "141eb1403015408aa3e6a77e2aelab22",
            "operations": [
              {
                "id": 1,
                "operationType": "",
                "propertyType": "ot_repositoryDestination"
              }
            ]
          }
        ],
        "drawInstructions": {
          "x": 360,
          "y": 180
        },
        "id": "166",
        "isDestination": true,
        "isGroup": false,
        "nodeId": "RepositoryHP@00-25FAAB-3B94CF-459595-78584C[1-1-8D]",
        "type": "HBBRepository"
      },

```

```

{
  "assignedPolicies": [
    {
      "id": "141eb1403015408aa3e6a77e2ae1ab22",
      "operations": [
      ]
    }
  ],
  "drawInstructions": {
    "x": 90,
    "y": 180
  },
  "id": "167",
  "isDestination": false,
  "isGroup": false,
  "nodeId": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
  "type": "OSHost"
},
{
  "id": "da2afe41-a936-4879-8cba-ed37aa421b26",
  "isActive": false,
  "modifiedDate": "2020-05-11T15:20:03Z",
  "numInError": 0,
  "numInProgress": 0,
  "numNotifications": 0,
  "numOffline": 0,
  "permissions": [
    {
      "access": "eA_READWRITE",
      "propertyType": "pt_user",
      "pt_user": {
        "upn": "Administrator@master"
      }
    }
  ],
  "version": 1,
  "versionActive": 0
}

```

*Contents of 'createdataflow.json'*

```

{
  "data": {
    "connections": [
      {
        "destination": "c3903",
        "id": "c4017",
        "label": "",
        "netCompression": false,
        "routedPolicies": [
          "141eb1403015408aa3e6a77e2ae1ab22"
        ],
        "source": "c3921",
        "type": "eCT_BATCH"
      }
    ],
    "description": "",
    "name": "demodataflow",
    "nodes": [
      {
        "assignedPolicies": [
          {
            "id": "141eb1403015408aa3e6a77e2ae1ab22",
            "operations": [
              {
                "id": 1
              }
            ]
          }
        ],
        "drawInstructions": {
          "h": 90,
          "w": 90,
          "x": 360,
          "y": 180
        },
        "id": "c3903",
        "name": " RepositoryHP",
        "nodeId": "RepositoryHP@00-25FAAB-3B94CF-459595-78584C[1-1-8D]",
        "type": "HBBRepository"
      },
      {
        "assignedPolicies": [
          {
            "id": "141eb1403015408aa3e6a77e2ae1ab22",
            "operations": []
          }
        ],
        "drawInstructions": {
          "h": 90,

```

```

        "w": 90,
        "x": 90,
        "y": 180
    },
    "id": "c3921",
    "name": "demomaster",
    "nodeId": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
    "type": "OSHost"
}
],
"userTags": []
}
}

```

## Compile and Activate Data Flow

Activation of a Data Flow first requires that it is “compiled”. This process ensures that the Data Flow and the Nodes and Policies used within it are valid and used correctly.

Both requests require JSON data consisting of an object with an “ids” array.

### Compile Data Flow

```

curl --insecure --cookie demoseession
"https://demomaster/API/7.1/master/RulesManager/services/Rules/actions/com
pile/invoke" --data "{\"ids\": [\"da2afe41-a936-4879-8cbaed37aa421b26\"]}"
--header "Content-Type: application/json" --request PUT

```

### Compile Data Flow Response

```

{
  "errors": [
  ],
  "summary": [
    "Compiling rules for data flow 'demodataflow'",
    "Rules summary for data flow 'demodataflow'",
    "'demomaster' implements:",
    "    'demopolicy'",
    "'demomaster' sends:",
    "    'demopolicy' from the local filesystem to 'RepositoryHP'",
    "RepositoryHP receives:",
    "    'demopolicy' from 'demomaster' for HBB storage",
    "Compilation of data flow 'demodataflow' completed successfully"
  ],
  "warnings": [
  ]
}

```

**Activate Data Flow**

```
curl --insecure --cookie demosection
"https://demomaster/API/7.1/master/RulesManager/services/Rules/actions/dis
tribute/invoke" --data "{\"ids\": [\"da2afe41-a936-4879-8cbaed37aa421b26\"]}"
--header "Content-Type: application/json" --request PUT
```

**Activate Data Flow Response**

```
{
  "handlerInfo": {
    "class": "Jobs",
    "handler": "JobStatusHandler",
    "instance": "{a46aa5ba-1797-4971-89ad-bbd85006bab8}",
    "machine": "master",
    "url": "master/JobStatusHandler/objects/Jobs/{a46aa5ba-1797-4971-89ad-
bbd85006bab8}"
  },
  "id": "{a46aa5ba-1797-4971-89ad-bbd85006bab8}"
}
```

## Trigger Backups

After activation a backup will occur automatically according to the Policy operation RPO and Schedule. A backup can also be triggered whenever required using the Trigger Handler.

To perform the trigger a number of trigger properties are required. First the triggers are retrieved, then the trigger properties are sent using the “triggerbackup.json” file.

If a backup is successfully triggered the response will contain a list of Job IDs.

**List Triggers**

```
curl --insecure --cookie demosection
"https://demomaster/API/7.1/master/TriggerHandler/objects/Triggers"
```

**List Triggers Response**

```
{
  "pageInfo": {
    "end": true,
    "totalCount": 1
  },
  "trigger": [
    {
      "dataOriginNodeId": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
      "dataOriginNodeInstance": 167,
      "dataflowId": "da2afe41-a936-4879-8cba-ed37aa421b26",
      "dataflowName": "demodataflow",
      "destinationNodeId": "RepositoryHP@00-25FAAB-3B94CF-459595-78584C[1-1-8D]",
      "destinationNodeInstance": 166,
      "id": "0",
      "moverType": "eMOVER_BATCH",
      "operationId": 1,
      "operationName": "Backup",
      "operationType": "eOPERATION_BACKUP",
      "policyId": "141eb1403015408aa3e6a77e2ae1ab22",
      "policyName": "demopolicy",
      "sourceNodeId": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
      "sourceNodeInstance": 167,
      "syncGroup": ""
    }
  ]
}
```

**Trigger backup**

```
curl --insecure --cookie demosession
"https://demomaster/API/7.1/master/TriggerHandler/services/Triggers/action
s/triggeroperation/invoke" --data @triggerbackup.json --header "Content-
Type: application/json" --request PUT
```

**Trigger backup Response**

```
{
  "job": [
    {
      "id": "{b9987ce9-4b35-4398-aefe-49b9acb29877}"
    }
  ]
}
```

*Contents of 'triggerbackup.json'*

```
{
  "dataOriginNodeId": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
  "dataOriginNodeInstance": 167,
  "dataflowId": "da2afe41-a936-4879-8cba-ed37aa421b26",
  "dataflowName": "demodataflow",
  "destinationNodeId": "RepositoryHP@00-25FAAB-3B94CF-459595-78584C[1-1-8D]",
  "destinationNodeInstance": 166,
  "id": "0",
  "moverType": "eMOVER_BATCH",
  "operationId": 1,
  "operationName": "Backup",
  "operationType": "eOPERATION_BACKUP",
  "policyId": "141eb1403015408aa3e6a77e2ae1ab22",
  "policyName": "demopolicy",
  "sourceNodeId": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
  "sourceNodeInstance": 167,
  "syncGroup": ""
}
```

## List Jobs

To verify the backup has been triggered and to track its progress the Job resource can be requested from the JobStatusHandler.

### List Jobs (maximum two items)

```
curl --insecure --cookie demosession
"https://demomaster/API/7.1/master/JobStatusHandler/objects/Jobs?offset=0&
count=2"
```

**List Jobs Response**

```

{
  "job": [
    {
      "actions": [
      ],
      "description": "Distribute Rules",
      "id": "{0021ad82-a8c2-4e32-8c48-caf3a89c199a}",
      "parameters": [
        {
          "key": "DataFlowID_1",
          "value": "1c020dc1-c60c-4a54-9b94-17bbfc12a9a3"
        },
        {
          "key": "DataFlowName_1",
          "value": "demodataflow"
        }
      ],
      "progress": "Completed",
      "reporterNode": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
      "status": "eJOB_SUCCEEDED",
      "subType": {
        "id": "Deactivate Data Flow(s)",
        "subSystem": "Rules Manager"
      },
      "timeCompleted": "2020-05-11T11:39:17Z",
      "timeStarted": "2020-05-11T11:39:14Z",
      "timeUpdated": "2020-05-11T11:39:17Z",
      "type": "eJOBTYPE_OTHER",
      "userTags": [
      ]
    },
    {
      "actions": [
      ],
      "description": "Compile Rules",
      "id": "{030401a2-948e-4189-b0b0-b14e1596079e}",
      "parameters": [
        {
          "key": "DataFlowID_1",
          "value": "bd215091-0d8a-49f3-92e7-7db37acbb8b4"
        },
        {
          "key": "DataFlowName_1",
          "value": "demodataflow"
        }
      ],
      "progress": "Completed",
      "reporterNode": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
      "status": "eJOB_SUCCEEDED",

```

```

        "subType": {
            "id": "Compile Data Flow(s)",
            "subSystem": "Rules Manager"
        },
        "timeCompleted": "2020-05-11T13:07:22Z",
        "timeStarted": "2020-05-11T13:07:22Z",
        "timeUpdated": "2020-05-11T13:07:22Z",
        "type": "eJOBTYPE_OTHER",
        "userTags": [
        ]
    }
],
"pageInfo": {
    "count": 2,
    "end": false,
    "offset": 0,
    "totalCount": 20
}
}

```

## Restore a backup to an alternate location

Once a backup has been completed the data can be restored. First the Recovery Point ID must be identified and then the restore request made.

If a restore is successfully started the response will contain a Job resource.

### List Repository Backups (maximum five records)

```

curl --insecure --cookie demoseession
"https://demomaster/API/7.1/master/RecoveryPointMetaDataAggregator/objects
/RecoveryPoints?offset=0&count=5"

```

*List Repository Backups Response*

```

{
  "pageInfo": {
    "count": 1,
    "end": true,
    "offset": 0,
    "totalCount": 1
  },
  "recoveryPoint": [
    {
      "applicationAwareBackup": false,
      "applicationMetadata": {
        "dataoriginos": "eOS_WINDOWS"
      },
      "applicationNode": {
        "id": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
        "type": "OSHost"
      },
      "browsable": true,
      "captureDate": "2020-05-11T15:20:12Z",
      "dataOriginNode": {
        "id": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
        "type": "OSHost"
      },
      "dataSourceNode": {
        "id": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
        "type": "OSHost"
      },
      "dataType": "eDATA_FILESYSTEM",
      "dataflow": {
        "dataOriginInstanceId": "167",
        "dataSourceInstanceId": "167",
        "id": "da2afe41-a936-4879-8cba-ed37aa421b26",
        "moverType": "eMOVER_BATCH",
        "name": "demodataflow",
        "storageInstanceId": "166",
        "version": 1
      },
      "expiryDate": "2020-11-13T15:20:12Z",
      "friendlyName": "File System Backup",
      "hadProblems": false,
      "id": "RepositoryHP@00-25FAAB-3B94CF-459595-78584C[1-1-8D]/{4faa5274-3e89-46bb-944a-75cbae59a162}",
      "instances": [
        {
          "logicalId": "{6b50400d-80e9-4073-8430-9e4ab321f82f}",
          "storeId": "{35701081-28d0-44f8-94ad-87f3a1cbf509}"
        }
      ],
      "jobId": "{2d6baeba-ac32-4c53-b415-ad35f10e4eaf}",

```

```

    "logicalId": "{4faa5274-3e89-46bb-944a-75cbae59a162}",
    "newDataSize": "0",
    "policy": {
      "id": "141eb1403015408aa3e6a77e2ae1ab22",
      "name": "demopolicy",
      "operation": {
        "id": "1",
        "name": "Backup",
        "type": "eOPERATION_BACKUP"
      },
      "version": 2
    },
    "rbacOwner": {
      "id": "4294967297",
      "type": "eRESOURCE"
    },
    "sessionId": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]#1589210408587-3592-0",
    "size": "0",
    "state": {
      "locked": false,
      "mountState": "eNOT_MOUNTED"
    },
    "storageNode": {
      "id": "RepositoryHP@00-25FAAB-3B94CF-459595-78584C[1-1-8D]",
      "type": "HBBRepository"
    },
    "systemTags": [
    ],
    "userTags": [
    ]
  }
]
}

```

### Restore to alternate location

```

curl --insecure --cookie demosession
"https://demomaster/API/7.1/master/RecoveryHandler/objects/RecoveryPoints/
RepositoryHP@00-25FAAB-3B94CF-459595-78584C%5B1-1-8D%5D%2F%7B4faa5274-
3e89-46bb-944a-75cbae59a162%7D/actions/Restore/invoke" --data
@restore.json --header "Content-Type: application/json" --request PUT

```

**Restore to alternate location Response**

```
{
  "handlerInfo": {
    "class": "Jobs",
    "handler": "JobStatusHandler",
    "instance": "{5ad959cf-e9fb-405b-b9e5-01ec01195318}",
    "machine": "master",
    "url": "master/JobStatusHandler/objects/Jobs/{5ad959cf-e9fb-405b-b9e5-01ec01195318}"
  },
  "id": "{5ad959cf-e9fb-405b-b9e5-01ec01195318}"
}
```

## View RPO Report

The RPO Report is a key resource in verifying that the required backups are being performed on schedule and within their required recovery point objective.

The report data can be retrieved with a normal API request.

**View the RPO Report**

```
curl --insecure --cookie demosection
"https://demomaster/API/7.1/master/ReportHandler/objects/RPOS/Current/collections/entries"
```

## View the Session Summary Report Response

```

{
  "pageInfo": {
    "end": true,
    "totalCount": 1
  },
  "rPOReportEntry": [
    {
      "dataFlow": {
        "id": "da2afe41-a936-4879-8cba-ed37aa421b26",
        "name": "demodataflow"
      },
      "dataOriginNode": {
        "id": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
        "instanceId": 173,
        "type": "OSHost"
      },
      "destinationNode": {
        "id": "RepositoryHP@00-25FAAB-3B94CF-459595-78584C[1-1-8D]",
        "instanceId": 172,
        "store": "",
        "type": "HBBRepository"
      },
      "id": "0",
      "jobId": "{367baad7-b198-4e2d-a163-1cdb7eeaa90e}",
      "lastBackup": "2020-05-12T13:06:53Z",
      "moverType": "eMOVER_BATCH",
      "policy": {
        "id": "141eb1403015408aa3e6a77e2ae1ab22",
        "name": "demopolicy",
        "operation": {
          "id": 1,
          "name": "Backup",
          "rpo": {
            "period": 1,
            "units": "eRPO_HOURS"
          },
          "type": "eOPERATION_BACKUP"
        },
        "version": 2
      },
      "sessionId": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]#1589288804995-836-0",
      "sourceNode": {
        "id": "demomaster@BZ-334922-442788-343Z2Z-YYZ744[0-1-1]",
        "instanceId": 173,
        "type": "OSHost"
      },
      "status": "eRPO_MET"
    }
  ]
}

```

```
]
}
```

## Delete Backup

Backups can be deleted directly from the storage system to increase available storage space. The Recovery Point ID used for the restore example is used again to delete the backup.

### Delete Backup

```
curl --insecure --cookie demoseession
"https://demomaster/API/7.1/RepositoryHP@00-25FAAB-3B94CF-459595-
78584C%5B1-1-
8D%5D/StorageManagement/objects/RecoveryPoints/RepositoryHP@00-25FAAB-
3B94CF-459595-78584C%5B1-1-8D%5D%2F%7B4faa5274-3e89-46bb-944a-
75cbae59a162%7D/" --request DELETE
```

### Delete Backup Response

```
null
```

## Log Out

It is recommended to log out to close the session once all required tasks have been completed.

### Log out

```
curl --insecure --cookie-jar demoseession
"https://demomaster/API/7.1/master/UIController/services/Users/actions/log
out/invoke" --silent
```

## Checking the Application version

It is possible to retrieve the application version without a logged in session. This allows users to verify the version and make the correct assumptions on available functionality before continuing.

### Get the Application version

```
curl --insecure
"https://demomaster/API/7.1/master/NodeManager/objects/ProductInformation/"
```

### Get the Application version Response

```
{  
  "id": "0",  
  "masterVersion": "7.1.0.84083-R7.1"  
}
```

# Glossary

## **Archive**

A copy that is created for long-term retention.

## **Asynchronous journalling**

Transactions are written to disk and also placed in a journal log file, to protect against data loss in the event of a system failure. Transactions from the log file are sent to the destination machine.

## **Asynchronous replication**

Transactions are held in memory before being sent over the network. If the network is unavailable then transactions are written to disk and sent to the destination machine when the connection is re-established. Asynchronous replication is optimal for connections with sporadic efficiency.

## **Backup**

A copy that is created for operational and disaster recovery.

## **Bandwidth throttling**

Used to control when and what proportion of available network bandwidth is used by Ops Center Protector for replication.

## **Batch backup**

A process by which the repository is updated periodically using scheduled resynchronizations. This method involves a scan of the source machine's file system, but only the changed bytes are transferred and stored. This method is useful for data that does not change often, such as data contained on the operating system disk. Linux based source nodes are only able to perform batch backups.

## **Clone**

An operation where a copy of the database is created in another storage location in a local or remote site.

## **COPY**

A hardware orchestration related status code that indicates that a volume pair is being created. An initial copy or resynchronization is being performed.

## **Data flow**

Identifies the data sources, movers and destinations participating in a backup, along with interconnection paths between them. Policies are assigned to each node to determine what type of data is backed up.

## **Data source**

A machine hosting a file system or application where the Protector client software is installed.

## **Deduplication**

A method of reducing the amount of storage space that your organization requires, to archive data, by replacing multiple instances of identical data with references to a single instance of that data.

## **Destination node**

A machine that is capable of receiving data for the purposes of archiving. This machine might be the Ops Center Protector Repository or Block device.

## **Dynamic Provisioning Virtual Volume (DP-VOL)**

Dynamic Provisioning Virtual Volume. A virtual volume that has no memory space. Used in Dynamic Provisioning.

**Hitachi Open Remote Copy Manager (HORCM)**

HORCM is a daemon process on the CCI server that communicates with the storage system and remote servers.

**Intelligent Storage Manager (ISM)**

A Protector Client node that acts as a proxy to Block storage devices.

ISM may also refer to the Intelligent Storage Manager process that runs within the Protector Client software.

**License key**

A unique, alphanumeric code that is associated with the unique machine ID that is generated during the Ops Center Protector installation. The license key must be activated in order to use the software.

**Logical Device (LDEV)**

An individual drive (or multiple drives in a RAID configuration) in the storage system. An LDEV might or might not contain any data and might or might not be assigned to any hosts. Each LDEV has a unique identifier, or address, within the storage system. The identifier is composed of the logical disk controller (LDKC) number, control unit (CU) number, and LDEV number.

The LDEV IDs within a storage system do not change. An LDEV formatted for use by open-system hosts is called a logical unit (LU).

**Master node**

The machine that controls the actions of other nodes within the Ops Center Protector network.

**Metadata Store (MDS)**

Records metadata that describes items that are held in repositories. The MDS supports indexing of stored data, thus enabling fast searches when locating data for restoration.

**Mover**

Defines the type of data movement operation to be performed between source and destination nodes, during the creation of a data flow. Batch movers perform block level data transfers on a scheduled basis, whereas continuous movers perform byte level data transfers on a near-continuous basis.

**Node Group**

Multiple machines of the same type can be assigned to one or more node groups. Within the Data Flow page, you can assign policies to nodes within node groups en-mass.

**PAIR**

A hardware orchestration related status code that indicates that a volume pair is now created. The initial copy has finished, and the paired volumes are synchronized.

**PFUL**

A hardware orchestration related status code that indicates that the amount of the data in the journal volume exceeds the threshold. The volume pair is not split and data continues to be copied.

**PFUS**

A hardware orchestration related status code that indicates that the amount of the data in the journal volume has reached 100% and the volume pair is split and data is no longer being copied.

**Policy**

A configurable data protection objective that is mapped to machines or groups, and to the data management agents that implement the policy. Multiple policies can be assigned to a single node.

**Primary Volume (P-VOL)**

The volume in a copy pair that contains the original data to be replicated. The data on the P-VOL is duplicated synchronously or asynchronously to the secondary volume (S-VOL).

**PSUE**

A hardware orchestration related status code that indicates that a volume pair is split due to an error.

**PSUS**

A hardware orchestration related status code that indicates that a volume pair is split by operation, or deleted from the storage system on the secondary site. This value is output for the P-VOL.

**Raw Device Mapping (RDM)**

Raw Device Mapping enables a LUN from a SAN to be directly connected to a VMware VM.

**Recovery Point Objective (RPO)**

The frequency at which a backup will occur. This governs the point in time to which data can be recovered should a restore be needed.

**Refreshed Thin Image (RTI)**

A local replication technique based on Thin Image snapshot. The S-VOL is refreshed based on a schedule or on demand. The S-VOL is a thin copy of the P-VOL and therefore needs the P-VOL to be available. Because the S-VOL is refreshed, its ID remains unchanged whenever its contents are updated.

**Replication**

An operation where a copy of the data is created in another local or remote location automatically.

**Repository**

A destination node that stores data from one or more source nodes. The Ops Center Protector Repository supports batch backup, archiving, and versioning policies.

**Secondary Volume (S-VOL)**

The volume in a copy pair that is the copy of the original data on the primary volume (P-VOL). See also primary volume.

**SMPL**

A hardware orchestration related status code that indicates that a volume is un-paired.

**Snapshot (Thin Image)**

A point in time copy of the data that is based on references to the original data.

**Source node**

Any node (server, workstation or virtual machine) that hosts data to be protected by Ops Center Protector. The source node has an Active Data Change Agent, which is responsible for monitoring the host file system and performing the relevant actions defined by the policies. Nodes need to be configured as a source node if they need to transfer locally stored data to a destination node, or implement data tracking, blocking and auditing functions. A node can be both a source and destination simultaneously.

**SSUS**

A hardware orchestration related status code that indicates that a volume pair is split by operation, or deleted from the storage system on the secondary site. This value is output for the S-VOL.

**SSWS**

A hardware orchestration related status code that indicates that the P-VOL and S-VOL are switched. The S-VOL is writable.

**Synchronous replication**

Transactions are transferred to the remote storage device immediately and the write operation is signaled as completed only once data is confirmed as written to both primary and secondary volumes. Synchronous replication is optimal for connections with high efficiency.

**Virtual Storage Machine (VSM)**

Virtual Storage Machine. A virtualised block storage device that exists within a physical storage array.

# Conventions for storage capacity values

Physical storage capacity values (for example, disk drive capacity) are calculated based on the following values:

Physical capacity unit	Value
1 kilobyte (KB)	1,000 ( $10^3$ ) bytes
1 megabyte (MB)	1,000 KB or $1,000^2$ bytes
1 gigabyte (GB)	1,000 MB or $1,000^3$ bytes
1 terabyte (TB)	1,000 GB or $1,000^4$ bytes
1 petabyte (PB)	1,000 TB or $1,000^5$ bytes
1 exabyte (EB)	1,000 PB or $1,000^6$ bytes

Logical capacity values (for example, logical device capacity, cache memory capacity) are calculated based on the following values:

Logical capacity unit	Value
1 block	512 bytes
1 cylinder	Mainframe: 870 KB Open-systems: <ul style="list-style-type: none"><li>▪ OPEN-V: 960 KB</li><li>▪ Others: 720 KB</li></ul>
1 KB	1,024 ( $2^{10}$ ) bytes
1 MB	1,024 KB or $1,024^2$ bytes
1 GB	1,024 MB or $1,024^3$ bytes
1 TB	1,024 GB or $1,024^4$ bytes
1 PB	1,024 TB or $1,024^5$ bytes
1 EB	1,024 PB or $1,024^6$ bytes

## Hitachi Vantara

Corporate Headquarters  
2535 Augustine Drive  
Santa Clara, CA 95054 USA



[HitachiVantara.com/contact](https://HitachiVantara.com/contact)