Dell EMC NetWorker

Version 18.2

REST API Getting Started Guide

302-005-330

REV 01

Published December 2018

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

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS-IS.“ DELL MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. USE, COPYING, AND DISTRIBUTION OF ANY DELL SOFTWARE DESCRIBED IN THIS PUBLICATION REQUIRES AN APPLICABLE SOFTWARE LICENSE.

Dell EMC
Hopkinton, Massachusetts 01748-9103
1-508-435-1000 In North America 1-866-464-7381
www.DellEMC.com

CONTENTS

TABLES

Preface

As part of an effort to improve its product lines, Dell EMC periodically releases revisions of its software and hardware. Therefore, some functions that are described in this document might not be supported by all versions of the software or hardware currently in use. The product release notes provide the most up-to-date information on product features.

Contact your Dell EMC technical support professional if a product does not function correctly or does not function as described in this document.

Note

This document was accurate at publication time. Go to Dell EMC Online Support (https://support.emc.com) to ensure that you are using the latest version of this document.

Purpose

This document describes how to configure and use NetWorker REST API.

Audience

This guide is part of the NetWorker documentation set, and is intended for use by developers who are creating programmatic interfaces to NetWorker systems.

Revision history

The following table presents the revision history of this document.

Table 1 Revision history

Revision	Date	Description
01	December 14, 2018	First release of the document for NetWorker 18.2.

Related documentation

The NetWorker documentation set includes the following publications, available on the Support website:

- *NetWorker E-LAB Navigator*
  Provides compatibility information, including specific software and hardware configurations that NetWorker supports. To access E-LAB Navigator, go to https://elabnavigator.emc.com/eln/elnhome.
- *NetWorker Administration Guide*
  Describes how to configure and maintain the NetWorker software.
- *NetWorker Network Data Management Protocol (NDMP) User Guide*
  Describes how to use the NetWorker software to provide data protection for NDMP filers.
- *NetWorker Cluster Integration Guide*
  Contains information related to configuring NetWorker software on cluster servers and clients.
- *NetWorker Installation Guide*
  Provides information on how to install, uninstall, and update the NetWorker software for clients, storage nodes, and servers on all supported operating systems.

- *NetWorker Updating from a Previous Release Guide*
  Describes how to update the NetWorker software from a previously installed release.
- *NetWorker Release Notes*
  Contains information on new features and changes, fixed problems, known limitations, environment and system requirements for the latest NetWorker software release.
- *NetWorker Command Reference Guide*
  Provides reference information for NetWorker commands and options.
- *NetWorker Data Domain Boost Integration Guide*
  Provides planning and configuration information on the use of Data Domain devices for data deduplication backup and storage in a NetWorker environment.
- *NetWorker Performance Optimization Planning Guide*
  Contains basic performance tuning information for NetWorker.
- *NetWorker Server Disaster Recovery and Availability Best Practices Guide*
  Describes how to design, plan for, and perform a step-by-step NetWorker disaster recovery.
- *NetWorker Snapshot Management Integration Guide*
  Describes the ability to catalog and manage snapshot copies of production data that are created by using mirror technologies on storage arrays.
- *NetWorkerSnapshot Management for NAS Devices Integration Guide*
  Describes how to catalog and manage snapshot copies of production data that are created by using replication technologies on NAS devices.
- *NetWorker Security Configuration Guide*
  Provides an overview of security configuration settings available in NetWorker, secure deployment, and physical security controls needed to ensure the secure operation of the product.
- *NetWorker VMware Integration Guide*
  Provides planning and configuration information on the use of VMware in a NetWorker environment.
- *NetWorker Error Message Guide*
  Provides information on common NetWorker error messages.
- *NetWorker Licensing Guide*
  Provides information about licensing NetWorker products and features.
- *NetWorker REST API Getting Started Guide*
  Describes how to configure and use the NetWorker REST API to create programmatic interfaces to the NetWorker server.
- *NetWorker REST API Reference Guide*
  Provides the NetWorker REST API specification used to create programmatic interfaces to the NetWorker server.
- *NetWorker 18.2 with CloudBoost 18.2 Integration Guide*
  Describes the integration of NetWorker with CloudBoost.
- *NetWorker 18.2 with CloudBoost 18.2 Security Configuration Guide*
  Provides an overview of security configuration settings available in NetWorker and Cloud Boost, secure deployment, and physical security controls needed to ensure the secure operation of the product.
- NetWorker Management Console Online Help
  Describes the day-to-day administration tasks performed in the NetWorker Management Console and the NetWorker Administration window. To view the online help, click **Help** in the main menu.

- NetWorker User Online Help
  Describes how to use the NetWorker User program, which is the Windows client interface, to connect to a NetWorker server to back up, recover, archive, and retrieve files over a network.

Special notice conventions that are used in this document

The following conventions are used for special notices:

NOTICE

Identifies content that warns of potential business or data loss.

Note

Contains information that is incidental, but not essential, to the topic.

Typographical conventions

The following type style conventions are used in this document:

Table 2 Style conventions

Bold	Used for interface elements that a user specifically selects or clicks, for example, names of buttons, fields, tab names, and menu paths. Also used for the name of a dialog box, page, pane, screen area with title, table label, and window.
Italic	Used for full titles of publications that are referenced in text.
`Monospace`	Used for: • System code • System output, such as an error message or script • Pathnames, file names, file name extensions, prompts, and syntax • Commands and options
`Monospace italic`	Used for variables.
`Monospace bold`	Used for user input.
[ ]	Square brackets enclose optional values.
|	Vertical line indicates alternate selections. The vertical line means or for the alternate selections.
{ }	Braces enclose content that the user must specify, such as x, y, or z.
...	Ellipses indicate non-essential information that is omitted from the example.

You can use the following resources to find more information about this product, obtain support, and provide feedback.

Where to find product documentation

- https://www.dell.com/support
- https://community.emc.com

Where to get support

The Support website https://www.dell.com/support provides access to product licensing, documentation, advisories, downloads, and how-to and troubleshooting information. The information can enable you to resolve a product issue before you contact Support.

To access a product-specific page:

1. Go to https://www.dell.com/support.
2. In the search box, type a product name, and then from the list that appears, select the product.

Knowledgebase
The Knowledgebase contains applicable solutions that you can search for either by solution number (for example, KB000xxxxxx) or by keyword.

To search the Knowledgebase:

1. Go to https://www.dell.com/support.
2. On the **Support** tab, click **Knowledge Base**.
3. In the search box, type either the solution number or keywords. Optionally, you can limit the search to specific products by typing a product name in the search box, and then selecting the product from the list that appears.

Live chat
To participate in a live interactive chat with a support agent:

1. Go to https://www.dell.com/support.
2. On the **Support** tab, click **Contact Support**.
3. On the **Contact Information** page, click the relevant support, and then proceed.

Service requests
To obtain in-depth help from Licensing, submit a service request. To submit a service request:

1. Go to https://www.dell.com/support.
2. On the **Support** tab, click **Service Requests**.

Note

To create a service request, you must have a valid support agreement. For details about either an account or obtaining a valid support agreement, contact a sales representative. To get the details of a service request, in the `Service Request Number` field, type the service request number, and then click the right arrow.

To review an open service request:

1. Go to https://www.dell.com/support.
2. On the **Support** tab, click **Service Requests**.
3. On the **Service Requests** page, under **Manage Your Service Requests**, click **View All Dell Service Requests**.

Online communities
For peer contacts, conversations, and content on product support and solutions, go to the Community Network https://community.emc.com. Interactively engage with customers, partners, and certified professionals online.

How to provide feedback
Feedback helps to improve the accuracy, organization, and overall quality of publications. You can send feedback to DPAD.Doc.Feedback@emc.com.

CHAPTER 1

Introduction

This chapter includes the following topics:

Overview of the NetWorker REST API

The NetWorker REST API is an interface that provides programmatic access to the NetWorker data protection service. By using this REST API, customers can build client applications to automate NetWorker operations. This document describes how to access the NetWorker data protection service by using the NetWorker REST API, and provides example tutorials for common tasks.

NetWorker REST API service

The NetWorker REST API uses a service to facilitate operations. The NetWorker REST API service is deployed in the same Apache tomcat container as the NetWorker authentication service. The NetWorker REST API uses the same set of tomcat processes to deliver its service.

API versions for the NetWorker REST API

There are three versions of the NetWorker REST API, which correspond to releases of the NetWorker software. Versions of the REST API can be viewed at the root URI of `https://{nw-server-hostname}:9090/nwrestapi`.

Table 3 NetWorker REST API versions

API version	NetWorker release	URI
NetWorker REST API v1	NetWorker 9.0 and minor releases	`https://nw-server-hostname:9090/nwrestapi/v1`
NetWorker REST API v2	NetWorker 9.1 and minor releases	`https://nw-server-hostname:9090/nwrestapi/v2`
NetWorker REST API v3	NetWorker 9.2, 18.1 and 18.2	`https://nw-server-hostname:9090/nwrestapi/v3`

Code that is written using a specific version of the API continues to function when subsequent versions of the API are released. However, in some cases, older versions of certain resources might be deprecated in the future. NetWorker only deprecates resources when necessary. If an older version of a resource can continue to function, it is accessible at its original API version number URI.

Installation

The NetWorker REST API is installed as part of NetWorker installation.

As part of the installation process, an Apache Tomcat instance is installed and a non-root -user, nsrtomcat, is created. If your system has special user security requirements, ensure that proper operational permissions are granted to the nsrtomcat users.

The *NetWorker Installation Guide* contains further information.

Getting started with the NetWorker REST API

This document provides example tutorials about how to perform some of the common tasks that programmers working with the NetWorker software will want to accomplish. After working through these tutorials, the use of the API should be familiar enough that you will be able to develop interfaces that make use of the full range of available functionality.

NetWorker REST API specification

The complete specification for the NetWorker REST API is available in PDF format in the *NetWorker REST API Getting Started Guide*.

Note

The NetWorker REST API is targeted to automate daily backup tasks. One-time configuration tasks, such as client and device configuration, should be done outside of the REST API using NetWorker Management Console wizards.

Using the example tutorials in this guide

The example tutorials provided in this guide follow a similar pattern, first presenting an operation that achieves a particular objective (for example, creating a client) in the form of a NetWorker command line operation that can be run locally. Next, the same objective is presented in the form of a NetWorker REST API call. Users familiar with the NetWorker command line will recognize the command line form of these operations.

Because of limited command line support for VMware client operations, the examples in Working with VMware-based clients on page 37 use NetWorker Management Console examples, rather than command line examples.

This document uses an example NetWorker server installed on a server named mars, and NetWorker REST API is exposed in the following URL: `https://mars:9090/nwrestapi/`.

When following these tutorials, you can use a curl client, or any other HTTP client. Typical curl syntax to call the API will look as follows:

```
curl -X POST -H "Content-Type: application/json" -H "Authorization:
Basic YWRtaW5pc3RyYXRvcjohNFUyYnVpbGQ=" -d "{ "hostname" :
"saturn","saveset":"[C:\\]"}"
"https://mars:9090/nwrestapi/v2/global/clients" -k -1
```

The following table examines the parameters being passed into the curl command in the above example.

Table 4 Parameters being passed into the curl command

Parameter	Value
`-X` POST	This specifies the HTTP method that is being used. Generally: • The `GET` method is used to retrieve data. • The `PUT` method is used to update data. • The `POST` method is used to create. • The `DELETE` method is used to remove.

Table 4 Parameters being passed into the curl command (continued)

Parameter	Value
`-H` "Content-Type: application/json"	Specifies the type of payload being sent to the API. The NetWorker REST API uses json exclusively, so this never changes.
`-H` "Authorization: Basic YWRtaW5pc3RyYXRvcjohNFUyYnVpbGQ="	Specifies an authorization header that contains Base64 encoded username and password.
`-d` '{ "hostname" : "saturn" }'	Specifies the actual payload being sent to the API.
https://mars:9090/nwrestapi/v2/global/clients	Specifies the URL of the API endpoint to which the request is being sent.
`-k` -1	Specifies version of the SSL protocol to be used for communication with the API. This option is environment specific.

To make the examples more readable, examples in this document simplify the API request into tool independent form, where only the data that actually changes between requests is specified (generally the HTTP method, endpoint URL, and the actual payload). For the endpoint URL, the hostname in the URL will also be ignored (for example, https://mars:9090).

So the example will in fact be denoted as:

```
POST /global/clients{ "hostname" : "saturn" }
```

Note that the payload is only required for PUT and POST requests.

Authentication and authorization

With the NetWorker REST API, all endpoints other than initial landing endpoint (`https://`*`your-server-name`*`:9090/nwrestapi/`) require authentication.

The NetWorker REST API uses the standard, basic HTTP authentication as described https://en.wikipedia.org/wiki/Basic_access_authentication. Basic HTTP Authentication is a common standard and is supported by most common HTTP tools such as curl, Postman, web browsers and programming frameworks.

When you are using Basic HTTP Authentication, user credentials are passed to the API with every request by using the authorization header. The authorization header takes the form of "Basic *token*," where the *token* is a Base64-encoded string in the format username:password.

It is important to note that Base64 encoding is not an encryption, and does not provide any protection for the password being used. Therefore, it is essential for security purposes that the NetWorker REST API uses HTTPS, which provides transport layer encryption for all API interactions.

Once a user has been authenticated by the API, permissions to NetWorker resources will be based on the user's NetWorker permissions.

New features in the REST API version 3 of NetWorker release NetWorker 18.2

The REST API version 3 changes for NetWorker 18.2 are available at the URL `https://`*`nw-server-hostname`*`:9090/nwrestapi/v3`. This section summarizes the new features and changes for this release.

Note

- To use the new features in the REST API, ensure that the server and clients that are running the same version of NetWorker.

The *NetWorker REST API Reference Guide* provides further information about the new parameters discussed in this section.

Access to additional client attributes
The REST API version 3 of NetWorker 18.2 provides access to the following advanced client attributes:

- Client direct
- Pool
- Save operations
- Job control
- Data domain backup
- Data domain interface
- Server network interface
- NetWorker version

You can access these advanced client attributes by using the query filter “type=advanced” .

Upgrading the NetWorker REST API

NetWorker 18.2 includes support for the new features available in REST API version 3.

If you do not need any new functionality provided in REST API version 3, you can continue to use the existing endpoints (version 1 or version 2) as it is. Your existing application will work without requiring any changes. However, it is recommended that you upgrade to REST API version 3.

All existing functionality available in NetWorker REST API version 3 of earlier releases remains unchanged. To upgrade to REST API version 3, perform the following steps:

1. Regenerate the classes using either the Open API or JSON Schema definitions provided with version 3 of the NetWorker 18.2 release.
2. The URIs for certain resources have changed in version 3. Change the URIs in your code as required to accommodate these changes. New features in the REST API version 3 of NetWorker release NetWorker 18.2 on page 15 provides information about changed URIs. Also change the version numbers passed in the REST API URL from version 1 or version 2 to version 3.

CHAPTER 2

NetWorker REST API example usage

This chapter contains the following topics:

Creating a client

This section demonstrates how to create a client by using the NetWorker REST API.

Note

When creating module clients, use the Management Console Client Backup Configuration Wizard rather than the REST API.

Command line example

The following example command uses the `nsradmin` program to create a client:

```
nsradmin> create type: NSR client; name: saturn; save set: /etc/
hosts
                         type: NSR client;
                         name: saturn;
                     save set: /etc/hosts;
Create? y created resource id
46.0.236.47.0.0.0.0.71.53.50.87.10.5.167.140(1)
```

API request

The same client is created in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v2/global/clients
{
    "hostname" : "saturn",
    "saveSets" : ["/etc/hosts"]
}
```

API response

The following response is received from the API:

```
201 Created
```

Viewing all clients

This section demonstrates how to view all clients configured on a NetWorker server using the NetWorker REST API.

Command line example

The following example command uses the `nsradmin` program to list all client configured on the NetWorker server, and displays the returned data (in this case for one client, with the hostname "saturn"):

```
nsradmin> p type: NSR client
...
                        type: NSR client;
                        name: saturn;
                     comment: ;
                   directive: ;
       protection group list: ;
                    save set: /etc/hosts;
          Checkpoint enabled: Disabled;
```

```
Parallel save streams per save set: Disabled;
               remote access: ;
                 remote user: ;
                    password: ;
         NAS management user: ;
     NAS management password: ;
        NAS file access user: ;
    NAS file access password: ;
        index backup content: No;
              backup command: ;
                 Pre command: ;
                Post command: ;
     application information: ;
     ndmp vendor information: ;
                        ndmp: No;
  ndmp multi-streams enabled: No;
                Disable IPv6: No;
                  NAS device: No;
             NDMP array name: ;
  NAS device management name: ;
storage replication policy name: ;
         Probe resource name: ;
          Block based backup: No;
             executable path: ;
                     aliases: saturn, saturn.company.com;
                 parallelism: 12;
                 backup type: ;
                         tag: ;
        restricted data zone: ;
```

API request

The following command uses the NetWorker REST API to list all configured clients:

```
GET nwrestapi/v3/global/clients
```

API response

The following response is received from the API:

```
200 OK
{
  "clients": [
    {
      "aliases": [
        "10.118.244.148"
      ],
      "applicationInformation": [],
      "backupType": "vProxy",
      "blockBasedBackup": false,
      "checkpointEnabled": false,
      "clientId": "ae014526-00000004-5bea643b-5bea6fb5-00035000-
b2123256",
      "hostname": "10.118.244.148",
      "indexBackupContent": false,
      "links": [
        {
          "href": "https://10.118.252.15:9090/nwrestapi/v3/global/
clients/50.0.252.9.0.0.0.0.68.99.234.91.10.118.252.15",
          "rel": "item"
        }
      ],
      "nasDevice": false,
      "ndmp": false,
      "ndmpMultiStreamsEnabled": false,
      "ndmpVendorInformation": [],
      "parallelSaveStreamsPerSaveSet": false,
      "parallelism": 4,
      "protectionGroups": [],
      "remoteAccessUsers": [
        "user=cn=kiran,cn=Users,dc=Windows2012r2,host=windows2012r2"
      ],
      "resourceId": {
        "id": "50.0.252.9.0.0.0.0.68.99.234.91.10.118.252.15",
        "sequence": 1
      },
      "saveSets": [
        "All"
      ],
      "scheduledBackup": true,
      "storageNodes": [],
      "tags": []
    }
  ],
  "count": 1
}
```

API request for advanced attributes

The following command uses the NetWorker REST API to list all configured clients with advanced attributes:

```
GET nwrestapi/v3/global/clients?type=advanced
```

API response

The following response is received from the API:

```
200 OK
    {
  "clients": [
    {
      "aliases": [
        "10.118.244.148"
      ],
      "applicationInformation": [],
      "backupType": "vProxy",
      "blockBasedBackup": false,
      "checkpointEnabled": false,
      "clientDirectEnabled": true,
      "clientId": "ae014526-00000004-5bea643b-5bea6fb5-00035000-
b2123256",
      "dataDomainBackup": false,
      "dataDomainInterface": "IP",
      "hostname": "10.118.244.148",
      "indexBackupContent": false,
      "jobControl": [],
      "links": [
        {
          "href": "https://10.118.252.15:9090/nwrestapi/v3/global/
clients/50.0.252.9.0.0.0.0.68.99.234.91.10.118.252.15",
          "rel": "item"
        }
      ],
      "nasDevice": false,
      "ndmp": false,
      "ndmpMultiStreamsEnabled": false,
      "ndmpVendorInformation": [],
      "networkerVersion": "99.0.99.9680.Build.9680",
      "parallelSaveStreamsPerSaveSet": false,
      "parallelism": 4,
      "pool": "Default",
      "protectionGroups": [],
      "remoteAccessUsers": [
        "user=cn=kiran,cn=Users,dc=Windows2012r2,host=windows2012r2"
      ],
      "resourceId": {
        "id": "50.0.252.9.0.0.0.0.68.99.234.91.10.118.252.15",
        "sequence": 1
      },
      "saveOperations": "I18N:mode=nativepath",
      "saveSets": [
        "All"
      ],
      "scheduledBackup": true,
      "serverNetworkInterface": "10.118.244.149",
      "storageNodes": [],
      "tags": []
    }
  ],
  "count": 1
}
```

Viewing only selected fields for all clients

Because the example above listed all fields for all clients, the returned JSON objects will sometimes be very large. You can limit the number of fields that are returned by the NetWorker REST API, as follows.

Command line example

The following example command uses the `nsradmin` program to view all clients configured on a NetWorker, but restricting the returned data to only the client name and aliases (two clients are found in this example, with the hostnames "saturn" and "jupiter"):

```
nsradmin> show aliases; name
nsradmin> p type: NSR client
                        name: saturn;
                     aliases: saturn, saturn.company.com;

                        name: jupiter;
                     aliases: jupiter;
```

API request

The following command performs the same functionality using the NetWorker REST API:

```
GET nwrestapi/v3/global/clients?fl=aliases,hostname
```

API response

The following response is received from the API:

```
200 OK
{
  "clients": [
    {
      "aliases": [
        "jupiter"
      ],
      "hostname": "jupiter"
    },
    {
      "aliases": [
        "saturn",
        "saturn.company.com"
      ],
      "hostname": "saturn"
    }
  ],
  "count": 2
}
```

Searching for clients

You can also restrict the number of clients that are found by using search criteria. In this case example, only one client with the hostname of "saturn" will be returned.

Command line example
The following example command uses the `nsradmin` program to search for a client:

```
nsradmin> . type: NSR client; name: saturn
Current query set
nsradmin>            name: saturn;
                     aliases: saturn, saturn.company.com;
```

API request
The same search is performed in the following example by using the NetWorker REST API:

```
GET /nwrestapi/v3/global/clients?
fl=aliases,hostname&q=hostname:saturn
```

API response
The following response is received from the API:

```
200 OK
{
  "clients": [
    {
      "aliases": [
        "saturn",
        "saturn.company.com"
      ],
      "hostname": "saturn"
    }
  ],
  "count": 1
}
```

Creating an Advanced File Type Device

This section demonstrates how to create an Advanced File Type Device (AFTD) using the NetWorker REST API.

Command line example
The following example command uses the `nsradmin` program to create an AFTD:

```
nsradmin> create type: NSR device; name: /space/storage; device
access information: /space/storage; media type: adv_file
                         type: NSR device;
                         name: /space/storage;
    device access information: /space/storage;
                   media type: adv_file;
Create? y created resource id
47.0.236.47.0.0.0.0.71.53.50.87.10.5.167.140(1) nsradmin> q
```

API request
The same device is created in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v2/global/devices
{
    "name": "/space/storage",
    "deviceAccessInfo": "/space/storage",
    "mediaType": "adv_file"
}
```

API response
The following response is received from the API:

```
201 Created
```

Labeling a device

This section demonstrates how to label a device using the NetWorker REST API.

Command line example
The following example command uses the `nsrmm` command to label a device:

```
# nsrmm -l -m -y -b Default -f /space/storage
Using volume name `mars.001' for pool `Default'
```

API request
The device is labeled in the same manner in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v3/global/devices/{deviceName}/op/label
{
  "labelWithoutMount": "false",
  "pool": "Default"
}
```

API response

The following response is received from the API:

```
202 Accepted
```

Creating a protection group

This section demonstrates how to create a protection group by using the NetWorker REST API.

Command line example

The following example command uses the `nsrpolicy` command to create a protection group :

```
# nsrpolicy group create client -g EngineeringWorkstations -C saturn
nsrpolicy: Group 'EngineeringWorkstations' was successfully created
```

API request

The same protection group is created in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v2/global/protectiongroups
{
    "workItemType" : "Client",
    "name" : "EngineeringWorkstations",
    "workItems" :
["48.0.254.77.0.0.0.0.149.148.51.87.128.222.109.201"]
}
```

Note

In the command line example, the name of the client is used. However when using the REST API, the resourceId of the client must be used instead, as show in this example in the `workItems` element. This can be retrieved from the resourceId element in the client resource, as shown in the API response section of Viewing all clients on page 18.

API response

The following response is received from the API:

```
201 Created
```

Creating a protection policy

This section demonstrates how to create a protection policy by using the NetWorker REST API.

Command line example
The following example command uses the `nsrpolicy` command to create a protection policy:

```
# nsrpolicy policy create -p Engineering
nsrpolicy: Policy 'Engineering' was successfully created
```

API request
The same protection policy is created in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v2/global/protectionpolicies
{
    "name": "Engineering"
}
```

API response
The following response is received from the API:

```
201 Created
```

Creating a workflow

This section demonstrates how to create a workflow by using the NetWorker REST API.

Command line example
The following example command uses the `nsrpolicy` command to create a workflow:

```
# nsrpolicy workflow create -p Engineering -w ClientProtection -g
EngineeringWorkstations -S 21:00
nsrpolicy: workflow 'ClientProtection' was successfully created
```

API request
The same workflow is created in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v2/global/protectionpolicies/Engineering/workflows
{
    "name" : "ClientProtection",
    "protectionGroups" : ["EngineeringWorkstations"],
    "startTime" : "21:00"
}
```

API response

The following response is received from the API:

```
201 Created
```

Creating actions for a traditional backup

This section demonstrates how to create actions for a traditional backup by using the NetWorker REST API.

Command line example

The following example command uses the `nsrpolicy` command to create a actions for a traditional backup:

```
# nsrpolicy action create backup traditional -p Engineering -w
ClientProtection -A Backup
nsrpolicy: Assigned default schedule period of 'week' and schedule
activity of 'full, incr, incr, incr, incr, incr, incr'.
nsrpolicy: Action 'Backup' was given '1 Months' retention period
nsrpolicy: Action 'Backup' was given 'nsrserverhost' storage node
nsrpolicy: Action 'Backup' was given 'Default' destination pool
nsrpolicy: Action 'Backup' was successfully created
```

API request

The same actions are created in the following example by using the NetWorker REST API:

```
PUT /nwrestapi/v2/global/protectionpolicies/Engineering/workflows/
ClientProtection
{
  "actions":[
  {
    "name":"Backup",
    "actionSpecificData":{
      "backup":{
         "backupSpecificData":{
            "traditional":{}
         }
      }
    }
 }
 ]
}
```

API response

The following response is received from the API:

```
204 No Content
```

Performing an on-demand backup

This section demonstrates how to perform an on-demand backup of a client by using the NetWorker REST API.

Command line example

The following example commands use the `nsrpolicy` command or the `nsrworkflow` command to perform an on-demand backup of a client:

```
# nsrpolicy start -p Engineering -w ClientProtection -c saturn:/etc/
hosts
144091:nsrpolicy: Workflow 'Engineering/ClientProtection' started
and has job id 32023
```

Or

```
# nsrworkflow -p Engineering -w ClientProtection -c saturn:/etc/
hosts
133550:nsrworkflow: Starting Protection Policy 'Engineering'
workflow 'ClientProtection'.
123316:nsrworkflow: Starting action 'Engineering/ClientProtection/
Backup' with command: 'savegrp -Z backup:traditional -v'.
123321:nsrworkflow: Action 'Engineering/ClientProtection/Backup's
log will be in '/nsr/logs/policy/Engineering/ClientProtection/
Backup_032028.raw'.
123325:nsrworkflow: Action 'Engineering/ClientProtection/Backup'
succeeded.
133553:nsrworkflow: Workflow 'Engineering/ClientProtection'
succeeded.
```

API request

The same on-demand backup is performed in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v2/global/protectionpolicies/Engineering/workflows/
ClientProtection/op/backup
{
    "clients": ["saturn:/etc/hosts"]
}
```

API response

The following response is received from the API:

```
201 Created
```

Retrieving information about the backup job

The location header in the response from the above API contains a URL to the job that was created. Use the following GET request to that URL to retrieve information about the job:

```
GET /nwrestapi/v2/global/jobs/64081
```

The following information is returned:

```
{
   "adhocJob": false,
   "command": "/usr/sbin/nsrworkflow -p Engineering -w
ClientProtection -L -c saturn:/etc/hosts",
   "dependentJobIds": [
     0
   ],
   "id": 64081,
   "itemIdLong": 64081,
   "links": [
     {
       "href": "https://mars:9090/nwrestapi/v2/global/jobs/64081/op/
cancel",
       "title": "Cancel job"
     }
   ],
   "logFile": "/nsr/logs/policy/Engineering/
workflow_ClientProtection_064081.raw",
   "name": "Engineering",
   "ndmp": false,
   "parentJobId": 0,
   "previousJobId": 0,
   "rootParentJobId": 0,
   "runOnHost": "mars",
   "siblingJobIds": [],
   "startTime": "2016-05-17T12:34:07-04:00",
   "state": "Active",
   "stopped": false,
   "tenant": "",
   "type": "workflow job"
}
```

Viewing all backups for a client

This section demonstrates how to view all backups for a client by using the NetWorker REST API.

Command line example

The following example command uses the `mminfo` command to view the backups for a client:

```
# mminfo -avot -q "client=saturn"
 volume    type   client    date     time      size ssid
fl     lvl  name
mars.001   adv_file mars     05/17/16 11:41:38 6 KB 4030413746 cb
manual /etc/hosts
...
```

API request

The same request is created in the following example by using the NetWorker REST API:

```
GET /nwrestapi/v2/global/clients/
82.0.31.22.0.0.0.0.33.12.41.87.128.222.109.201/backups
```

Note

In the command line example, the name of the client is used. However when using the REST API, the resourceId of the client must be used instead, as show in this example URL. This can be retrieved from the resourceId element in the client resource, as shown in the API response section of Viewing all clients on page 18.

API response

The following response is received from the API:

```
{
  "backups": [
    {
      "attributes": [
        {
          "key": "*backup start time",
          "values": [
            "1463499698"
          ]
        },
        {
          "key": "*ss clone retention",
          "values": [
            "          1463499698:          1463499698:        -2"
          ]
        },
        {
          "key": "saveset features",
          "values": [
            "CLIENT_SAVETIME"
          ]
        }
      ],
      "browseTime": "2038-01-18T22:14:06-05:00",
      "clientHostname": "mars",
      "clientId":
"b7d73be0-00000004-57290c2f-57290c2e-00010c00-80c64a29",
      "completionTime": "2016-05-17T11:41:38-04:00",
      "creationTime": "2016-05-17T11:41:38-04:00",
      "fileCount": 3,
      "id": "67f3852c-00000006-f03b3bb2-573b3bb2-00120c00-80c64a29",
      "instances": [
        {
          "clone": false,
          "id": "1463499698",
          "status": "Browsable",
          "volumeIds": [
            "4264957582"
          ]
        }
      ],
      "level": "Manual",
      "links": [
        {
          "href": "https:\/\/mars:9090\/nwrestapi\/v2\/global\/
clients\/82.0.31.22.0.0.0.0.33.12.41.87.128.222.109.201\/backups\/
67f3852c-00000006-f03b3bb2-573b3bb2-00120c00-80c64a29",
          "rel": "item"
        }
      ],
      "name": "\/etc\/hosts",
      "retentionTime": "2038-01-18T22:14:06-05:00",
      "saveTime": "2016-05-17T11:41:38-04:00",
      "shortId": "4030413746",
      "size": {
        "unit": "Byte",
        "value": 6736
      },
      "type": "File"
    }
   ...
  ]
}
```

Viewing Client File Index

This section describes the process to support Indexing query and pagination, by using NetWorker REST API.

Viewing client indexes and pagination
The json response can be large, as it depends on client's index database. It provides paginated output. You can set the desired number of entries in `pagesize` query filter. The NetWorker REST API uses custom header `X-NW-CONTINUATION-TOKEN` to paginate the output. The NetWorker server returns `X-NW-CONTINUATION-TOKEN` header in response if client file indexes are more than `pagesize` (default maximum limit is 100). The requester has to provide `X-NW-CONTINUATION-TOKEN` with value received in last response header. The networker REST server reads `X-NW-CONTINUATION-TOKEN` header value and returns next set of index records.

Note

- The user gets a maximum value of `100` for query parameter `pagesize`
- The token has a Time-To-Live (TTL) of 120 sec.
- The token has `Read-Once` property, which means once server gets a valid token and serves the request, the token becomes invalid.
- The query parameters set at the beginning of the request cannot be modified for subsequent calls.

By using API request
Following API request helps you to view client indexes:

```
GET v3/global/clients/{clientResourceID}/indexes
```

Note

You must use the *ResourceID* of the client as the clientResourceID. For example,

```
GET nwrestapi/v3/global/clients
```

```
200 OK
.
.
.
.
"resourceId": {
"id": "50.0.252.9.0.0.0.0.68.99.234.91.10.118.252.15",
.
.
.
```

API request example

```
GET https://10.31.227.39:9090/nwrestapi/v3/global/clients/
159.0.52.3.0.0.0.0.226.242.11.90.10.31.227.39/indexes
```

API response example

Response header

```
Content-Type ?application/json
Date ?Thu, 21 Dec 2017 08:59:41 GMT
Server ?Apache-Coyote/1.1
Transfer-Encoding ?chunked
X-NW-CONTINUATION-TOKEN ?
MjAyZDUwMmI1ZDU2MzMwZDZlYzZiMDcwZDg5MjEwMjU6MTcxMTY4ODA=
```

Response body

```
[
 {
   "fileName": "C:\\Program Files\\EMC NetWorker\\Management\\GST\
\cst\\acm_lb.clb",
   "fileSizeInBytes": 4528,
   "namespace": "backup",
   "offset": 0,
   "saveTime": "2018-01-17T14:00:37+05:30"
 },
 {
   "fileName": "C:\\Program Files\\EMC NetWorker\\Management\\GST\
\cst\\acm_lb.clb.bak",
   "fileSizeInBytes": 3920,
   "namespace": "backup",
   "offset": 4528,
   "saveTime": "2018-01-17T14:00:37+05:30"
 },
…..
]
```

Recovering Filesystem backup

This section describes the process to perform a file system saveset and granular recovery by using the NetWorker REST API.

Recovering a Saveset
This section describes the process to perform a saveset based recovery by using the NetWorker REST API.

By using NMC
To recover a saveset using NMC:

1. Navigate to the **Recover** tab.
2. Right click and select **New Recover**.
3. In the Recover wizard, select **Traditional Networker Client Recovery**.
4. Select a source client, destination client, and available recovery types for the File System, Block Based, and NDMP based File System.
5. Select the **Saveset Recover** tab and the saveset to be recovered.
6. Select **File Path** for **Recovery** and **Duplicate File Options**.

By using REST API
To perform file system saveset recovery, perform the steps mentioned in the following sections in the defined sequence:

1. Refer Prerequisites for recovery request on page 32
2. Refer Creating recover resource by using REST API on page 32
3. Refer Viewing details of recover resource on page 33
4. Refer Monitoring the status of recover job on page 34

Prerequisites for recovery request
Use `GET /nwrestapi/v3/global/backups` to get `backupInstance` information.

The backup ID can be retrieved from the value of attribute `id` by using the command `GET /backups` or `GET /backups/{backupId}/instances` for a specific backup instance.

Creating recover resource by using REST API
You can use the NetWorker REST API for a saveset recovery as described in the following example:

Filesystem (FS):

```
POST /nwrestapi/v3/global/recovers
{
"recoveryType": "Filesystem",
"recoveryDestination":"e:\\nsrRecover",
"backupInstance":
 {
  "backupID":"45b6b47c-00000006-faac2017-59ac2017-00075000-
bde5b856",
  "instanceID":"1504883884"
 },
"actionForDuplicateItems": "Rename"
}
```

Block based backup:

```
POST /nwrestapi/v3/global/recovers
{
"recoveryType": "BBB",
"targetVolume":"E:\\",
"backupInstance":
 {
  "backupID":"59ef04c5-00000006-e3aed96a-59aed96a-001f5000-bde5b856"
  "instanceID":"1504889037"
 },
"actionForDuplicateItems": "Overwrite"
}
```

NDMP based filesystem backup:

```
POST/nwrestapi/v3/global/recovers
{
"recoveryType": "NDMP",
"recoveryDestination":"/ifs/data/ajay/MohanRec",
"backupInstance":
 {
  "backupID":"855e9b15-00000006-f9b7fe7b-59b7fe7b-00085000-bde5b856"
  "instanceID":"1506346646"
 }
}
```

When making a API request, start with a backup ID for the client saveset. The following response is received from the REST API:

```
201 Created
```

Viewing details of recover resource

The API response contains a Resource URI in the location header. Use the Resource URI to get the recovery resource by using the `GET recovers/{recoverId}` command, which contains a link that you can use to monitor the job. Following is the sample output of `GET recovers/{recoverId}`

```
{
 "destinationClientResID":
"c9797b4d-00000004-59e73721-59e8490b-000c5000-a6138c56",
 "links":[
     {
        "href": "https://10.207.86.34:9090/nwrestapi/v3/global/jobs/
800142",
        "rel": "item"
         }
         ],
 "name": "NDMP_8817_189862517965772",
 "recoveryDestination": "/ifs/data/ajay/MohanRec",
 "recoveryStartTime": "2017-11-23T01:50:51-08:00",
 "recoveryType": "NDMP",
 "sourceClient": "ISILON",
 "backupInstance":
     {
        "backupID":"855e9b15-00000006-f9b7fe7b-59b7fe7b-00085000-
bde5b856"
        "instanceID":"1506346646"
     },
 "resourceId": {
     "id": "49.0.96.18.0.0.0.0.47.81.22.90.10.207.86.34",
     "sequence": 1
    }
}
```

Note

In highly loaded systems, job Id might not be created immediately. In such scenarios, consumer of the API has to periodically check the recover resource URI till the job is created. If job is not created for some reason, consumer of the API might need to time out with appropriate message.

Monitoring the status of recover job

To monitor the recovery job, use the following command:

```
GET jobs/{jobId}
```

Performing granular recovery of selected files or folders in a backup

This section demonstrates how to recover specific files and folders by using the NetWorker REST API.

By using NMC

To perform granular recovery using NMC:

1. Navigate to the **Recover** tab.
2. Right click and select **New Recover**.
3. In the Recover wizard, select **Traditional Networker Client Recovery**.
4. Select a source client, destination client, and available recovery types for the File System, Block Based, and NDMP based File System.
5. Select the **Browse** tab and select the file and folder to be recovered.
6. Select **File Path** for **Recovery** and **Duplicate File Options**.

By using REST API

To perform file system granular recovery, perform the steps mentioned in the following sections in the defined sequence:

1. Refer Prerequisites for recovery request on page 34
2. Refer Creating recover resource by using REST API on page 35
3. Refer Viewing details of recover resource on page 36
4. Refer Monitoring the status of recover job on page 36

Prerequisites for recovery request

For the `recoveryDestination` attribute, user should be aware of the file system.

- Retrieve `sourceClientResID` from the `resourceId` attribute of the command `GET /nwrestapi/v3/global/clients/{client resource id}`.
- Retrieve `timeStamp` from the `saveTime` attribute of the command `GET /nwrestapi/v3/global/backups`.

Note

User should know the filesystem to specify the value for `recoveryDestination` attribute.

Creating recover resource by using REST API

You can use the NetWorker REST API for a granular recovery as described in the following example:

Note

The supported formats for timestamp are epoch time in seconds and human readable format in <<YYYY-MM-DDTHH:MM:SS+GMT>>. For example: "2017-10-05T23:50:14+05:30"

Filesystem (FS):

```
POST /nwrestapi/v3/global/recovers
{
 "recoveryType": "Filesystem",
 "recoveryDestination":"e:\\nsrRecover",
 "timeStampBasedGranularRecover":
 {

"sourceClientResID":"159.0.80.10.0.0.0.0.176.177.169.89.32.2.190.239
.190.239.0.0.149.137.80.106",
   "timeStamp":"1506593756"
 },
 "itemsToRecover": ["C:\\vbm_tools\\vbm_version.txt", "C:\\vbm_tools
\\trusted.reg"],
 "actionForDuplicateItems": "Rename"
}
```

Block based backup:

```
POST /nwrestapi/v3/global/recovers
 {
  "recoveryType": "BBB",
  "recoveryDestination":"E:\\BBB_recover",
  "timeStampBasedGranularRecover":
  {

"sourceClientResID":"160.0.184.15.0.0.0.0.80.152.83.90.10.63.30.91",
   "timeStamp":"2018-01-09T09:38:33+05:30"
  },
 "itemsToRecover": ["F:\\BBB\\a.pdf", "F:\\BBB\\b.pdf"]
}
```

NDMP based filesystem backup:

```
POST /nwrestapi/v3/global/recovers
 {
  "recoveryType": "NDMP",
  "recoveryDestination":"/ifs/data/ajay/MohanRec",
  "timeStampBasedGranularRecover":
  {

"sourceClientResId":"43.0.168.31.0.0.0.0.252.65.185.89.32.2.190.239.
190.239.0.0.149.137.80.106",
    "timeStamp":"1506346646"
  },
"itemsToRecover": ["/ifs/data/ajay/images/firmware.xml", "/ifs/data/
ajay/images/devices.xml"]
}
```

The following response is received from the REST API:

```
201 Created
```

Viewing details of recover resource

The API response contains a Resource URI in the location header. Use the Resource URI to get the recovery resource by using the `GET recovers/{recoverId}` command, which contains a link that you can use to monitor the job. Following is the sample output of `GET recovers/{recoverId}`

```
{
    "destinationClientResID":
"c9797b4d-00000004-59e73721-59e8490b-000c5000-a6138c56",
    "itemsToRecover": [
        "/ifs/data/ajay/images/firmware.xml",
        "/ifs/data/ajay/images/devices.xml"
    ],
    "links": [
        {
            "href": "https://10.207.86.34:9090/nwrestapi/v3/global/
jobs/800142",
            "rel": "item"
        }
    ],
    "name": "NDMP_8817_189862517965772",
    "recoveryDestination": "/ifs/data/ajay/MohanRec",
    "recoveryStartTime": "2017-11-23T01:50:51-08:00",
    "recoveryType": "NDMP",
    "resourceId": {
        "id": "49.0.96.18.0.0.0.0.47.81.22.90.10.207.86.34",
        "sequence": 1
    },
    "sourceClient": "ISILON",
    "timeStampBasedGranularRecover": {
        "sourceClientResID":
"c9797b4d-00000004-59e73721-59e8490b-000c5000-a6138c56",
        "timeStamp": "2017-11-23T00:38:11-08:00"
    }
}
```

Note

In highly loaded systems, job Id might not be created immediately. In such scenarios, consumer of the API has to periodically check the recover resource URI till the job is created. If job is not created for some reason, consumer of the API might need to time out with appropriate message.

Monitoring the status of recover job

To monitor the recovery job, use the following command:

```
GET jobs/{jobId}
```

Working with VMware-based clients

Because of limited command line support for VMware client operations, the examples in this section use NetWorker Management Console (NMC) examples, rather than command line examples.

Note

The NetWorker REST API does not have an ability to register a vCenter or a vProxy object, so we assume that these objects have been already configured in NMC.

Performing vCenter Operations

This section describes the process to create, view, modify and delete vCenter, by using the NetWorker REST API.

Creating vCenter resource
You can create vCenter resource using `POST` call of following URI `/vmware/vcenters`

By using API request
Following API request helps you to create vCenter resource:

```
POST vmware/vcenters
```

Example 1 API request example

```
POST https://10.207.86.34:9090/nwrestapi/v3/global/vmware/vcenters
{
  "hostname": "10.63.30.165",
  "userName": "administrator@vsphere.local",
  "userPassword": "Emclegato@123!"
}
```

Viewing the list of vCenter resources
You can view the list of all the vCenter resources using NetWorker REST API.

By using API request

```
GET https://10.63.30.91:9090/nwrestapi/v3/global/vmware/vcenters
```

Example 2 API response example

```
200 OK
{
 "count": 2,
 "vCenters":
  [
   {
```

Example 2 API response example (continued)

```
    "cloudDeployment": false,
    "hostname": "10.207.86.28",
    "links":
     [
      {
       "href": "https://10.63.30.91:9090/nwrestapi/v3/global/vmware/
vcenters/10.207.86.28",
       "rel": "item"
      }
     ],
     "resourceId":
     {
      "id": "50.0.68.13.0.0.0.0.18.13.116.90.10.63.30.91",
      "sequence": 1
     },
      "userName": "administrator@vsphere.local",
      "userPassword": "*******"
        },
        {
            "cloudDeployment": false,
            "hostname": "10.63.30.165",
            "links":
             [
              {
               "href": "https://10.63.30.91:9090/nwrestapi/v3/
global/vmware/vcenters/10.63.30.165",
               "rel": "item"
              }
             ],
            "resourceId":
             {
              "id": "164.0.184.15.0.0.0.0.80.152.83.90.10.63.30.91",
              "sequence": 5833
             },
            "userName": "administrator@vsphere.local",
            "userPassword": "*******"
        }
    ]
}
```

Viewing vCenter resource

You can view vCenter resource using `GET` call of following URI `/vmware/vcenters/{vcenter-id}`

By using API request

Following API request helps you to view vCenter resource:

```
GET /vmware/vcenters/{vcenter-id}
```

Example 3 API request example

```
GET https://10.207.86.34:9090/nwrestapi/v3/global/vmware/vcenters/
10.63.30.165
```

Example 4 API response example

Example 4 API response example

```
{
 "cloudDeployment": false,
 "hostname": "10.63.30.165",
 "links":
  [
   {
    "href": "https://10.207.86.34:9090/nwrestapi/v3/global/vmware/
vcenters/10.63.30.165/op/refresh",
            "title": "Refresh vCenter view"
   },
   {
    "href": "https://10.207.86.34:9090/nwrestapi/v3/global/vmware/
vcenters/10.63.30.165/protectedvms",
            "title": "List of protected virtual machines"
   },
   {
    "href": "https://10.207.86.34:9090/nwrestapi/v3/global/vmware/
vcenters/10.63.30.165/vms",
            "title": "List of virtual machines"
   },
   {
    "href": "https://10.207.86.34:9090/nwrestapi/v3/global/vmware/
vcenters/10.63.30.165/plugins",
            "title": "Intall vCenter plugins"
   }
  ],
  "resourceId":
   {
    "id": "44.0.84.7.0.0.0.0.179.242.94.90.10.207.86.34",
    "sequence": 1
   },
  "userName": "administrator@vsphere.local",
  "userPassword": "*******"
}
```

Modifying vCenter resource

You can modify vCenter resource using `PUT` call of following URI `/vmware/vcenters/{vcenter-id}`

By using API request

Following API request helps you to modify vCenter resource:

```
PUT /global/vmware/vcenters/{vcenter-id}
```

Example 5 API request example

```
PUT https://10.207.86.34:9090/nwrestapi/v3/global/vmware/vcenters/
10.63.30.165
{
 "cloudDeployment": true
}
```

Deleting vCenter resource

You can delete vCenter resource using DELETE call of following URI /vmware/vcenters/{vcenter-id}

By using API request

Following API request helps you to delete vCenter resource:

```
DELETE /vmware/vcenters/{vcenter-id}
```

Example 6 API request example

```
DELETE https://10.207.86.34:9090/nwrestapi/v3/global/vmware/
vcenters/10.63.30.165
```

Performing vProxy Operations

This section describes the process to create, register, view, modify and delete vProxy, by using the NetWorker REST API.

Creating vProxy resource

You can create vProxy resource using POST call of following URI /vmware/vproxies

By using API request

Following API request helps you to create vProxy resource:

```
POST /vmware/vproxies
```

Example 7 API request example

```
POST https://10.207.86.34:9090/nwrestapi/v3/global/vmware/vproxies
{
  "enabled": true,
  "hostname": "10.63.30.90",
  "maxHotaddSessions": 13,
  "maxNbdSessions": 8,
  "userName": "admin",
  "vCenterHostname": "10.63.30.165",
  "password":"emclegato",
  "vProxyPort": 9090
}
```

Registering vProxy resource

You can register vProxy resource using POST call of following URI /vmware/vproxies/op/register

By using API request

Following API request helps you to register vProxy resource:

```
POST /vmware/vproxies/op/register
```

Example 8 API request example

```
POST https://10.207.86.34:9090/nwrestapi/v3/global/vmware/
vproxies/op/register
{
 "enabled": true,
 "hostname": "10.63.30.90",
 "maxHotaddSessions": 13,
 "maxNbdSessions": 8,
 "userName": "admin",
 "vCenterHostname": "10.63.30.165",
 "password":"emclegato",
 "vProxyPort": 9090
}
```

Viewing the list of vProxy resources

You can view the list of all the vProxy resources using NetWorker REST API.

Example 9 API request example

```
GET https://10.63.30.91:9090/nwrestapi/v3/global/vmware/vproxies
```

Example 10 API response example

```
200 OK
{
 "count": 2,
 "vProxies":
  [
   {
    "datastores": [],
    "enabled": true,
    "encryptNbdSessions": false,
    "hostname": "10.63.30.89",
    "links":
     [
      {
       "href": "https://10.63.30.91:9090/nwrestapi/v3/global/vmware/
vproxies/10.63.30.89",
       "rel": "item"
      }
     ],
 "maxHotaddSessions": 13,
 "maxNbdSessions": 8,
 "password": "*******",
 "resourceId":
  {
   "id": "53.0.100.15.0.0.0.0.143.124.105.90.10.63.30.91",
```

Example 10 API response example (continued)

```
   "sequence": 7
  },
 "userName": "admin",
 "vCenterHostname": "10.63.30.165",
 "vProxyPort": 9090,
 "version": "Release: '3.0.0-1_SNAPSHOT20180126185157', Build
number: '1', Build date: '2018-01-26T18:51:57Z'"
},
{
 "datastores": [],
 "enabled": true,
 "encryptNbdSessions": false,
 "hostname": "10.63.30.90",
 "links":
[
 {
  "href": "https://10.63.30.91:9090/nwrestapi/v3/global/vmware/
vproxies/10.63.30.90",
  "rel": "item"
 }
],
  "maxHotaddSessions": 13,
  "maxNbdSessions": 8,
  "password": "*******",
  "resourceId":
   {
    "id": "51.0.68.13.0.0.0.0.18.13.116.90.10.63.30.91",
    "sequence": 1
   },
  "userName": "admin",
  "vCenterHostname": "10.63.30.165",
  "vProxyPort": 9090
  }
 ]
}
```

Viewing vProxy resource

You can view vProxy resource using `GET` call of following URI `/global/vmware/vproxies/{vproxy-id}`

By using API request

Following API request helps you to view vProxy resource:

```
GET /vmware/vproxies/{vproxy-id}
```

Example 11 API request example

```
GET https://10.207.86.34:9090/nwrestapi/v3/global/vmware/vproxies/
10.63.30.90
```

Example 12 API response example

Example 12 API response example (continued)

```
{
 "datastores": [],
 "enabled": false,
 "encryptNbdSessions": false,
 "hostname": "10.63.30.90",
 "maxHotaddSessions": 13,
 "maxNbdSessions": 8,
 "password": "*******",
 "resourceId":
  {
    "id": "46.0.84.7.0.0.0.0.179.242.94.90.10.207.86.34",
    "sequence": 2
  },
 "userName": "admin",
 "vCenterHostname": "10.63.30.165",
 "vProxyPort": 9090
}
```

Modifying vProxy resource

You can modify vProxy resource using `PUT` call of following URI `/vmware/vproxies/{vproxy-id}`

By using API request

Following API request helps you to modify vProxy resource:

```
PUT /vmware/vproxies/{vproxy-id}
```

Example 13 API request example

```
PUT https://10.207.86.34:9090/nwrestapi/v3/global/vmware/vproxies/
10.63.30.90
{
  "maxHotaddSessions": 11,
  "maxNbdSessions": 7
}
```

Deleting vProxy resource

You can delete vProxy resource using `DELETE` call of following URI `/vmware/vproxies/{vproxy-id}`

By using API request

Following API request helps you to delete vProxy resource:

```
DELETE /vmware/vproxies/{vproxy-id}
```

Example 14 API request example

Example 14 API request example (continued)

```
DELETE https://10.207.86.34:9090/nwrestapi/v3/global/vmware/
vproxies/10.63.30.90
```

Creating a protection group for VMware-based clients

This section demonstrates how to create a protection group for VMware-based clients by using the NetWorker REST API.

NMC example
To create a protection group in NMC:

1. Navigate to the **Protection** tab.
2. Select **Groups**.
3. Right-click and select **New** and enter a name for the group.
4. Set the group type to **VMware** and sub-type to **All**.
5. Select the vCenter to be used.
6. Browse the vCenter to select individual VMs and containers such as resource pools.

API request
The same protection group is created in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v2/global/protectiongroups
{
  "name": "ApiGroup",
  "vmwareWorkItemSelection": {
    "containerMorefs": [
      "resgroup-7675"
    ],
    "vCenterHostname": "hostname.company.com",
    "vmUuids": [
      "50085479-6e0f-2383-7cb8-15883df49fde"
    ],
    "vmdks": []
  },
  "workItemQueries": [],
  "workItemSource": "Static",
  "workItemSubType": "All",
  "workItemType": "VMware"
}
```

This protection group contains two selection items:

- A VM with a specific UUID—`50085479-6e0f-2383-7cb8-15883df49fde`. Resources such as `/vmware/vcenters/{vcenter-hostname}/vms`, which returns a list of VMs on the specified vCenter, can be used to retrieve the UUID for a VM.

  Note

  You can use POST at the `/vmware/vcenters/{vcenter-hostname}/op/refresh` endpoint gather the latest information from the specific vCenter.

- A container with a specified Managed Object Reference (MoRef)—`resgroup-7675`
  When a container is included, then all VMs included in this container will be backed up, even if they are added after the protection group was created. The NetWorker REST API does not expose all the containers present in the vCenter. These should be obtained through another mechanism such as directly from the vCenter.

API response
The following response is received from the API:

```
201 Created
```

Creating a protection policy for VMware-based clients

This section demonstrates how to create a protection policy for VMware-based clients by using the NetWorker REST API.

NMC example
To create a protection group in NMC:

1. Navigate to the **Protection** tab.
2. Select **Policies**.
3. Right-click and select **New** and enter a name for the policy.
4. Select **Create a new workflow** and enter a name for the workflow.
5. Associate the new workflow with a protection group. In this example, we will associate it with the ApiGroup created in Creating a protection group for VMware-based clients on page 44.
6. Create an action, enter a name for the action, and set the action type to **Backup** and the backup subtype to **VMware (vProxy)**.

API request
The same protection policy is created in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v2/global/protectionpolicies
{
  "name": "ApiPolicy",
  "workflows": [
  {
    "actions": [
    {
      "actionSpecificData": {
        "backup": {
          "backupSpecificData": {
            "vmwareVProxy": {
              "quiesceApp": false,
              "destinationPool": "Data Domain Default"
            }
          },
          "destinationStorageNodes": [
            "nsrserverhost"
          ]
        }
      },
      "name": "ApiAction",
      "scheduleActivities": [
        "incr",
        "incr",
        "incr",
        "incr",
        "incr",
        "incr",
        "incr"
      ],
      "schedulePeriod": "Week"
    }
    ],
    "name": "ApiWorkflow",
    "protectionGroups": [
      "ApiGroup"
    ]
  }
  ]
}
```

Some properties, such as `scheduleActivities`, are optional and will default to the same values as those shown in NMC.

After the policy resource is created, you can perform a GET for the resource URL returned in the location header to see the full resource with all populated default values.

API response
The following response is received from the API:

```
201 Created
```

Adding a new item to a for VMware-based protection group

This section demonstrates how to add a new item, such as a new VM, to a VMware-based protection group by using the NetWorker REST API.

NMC example
To add a new item to a VMware-based protection group in NMC:

1. Navigate to the **Protection** tab.
2. Select **Groups**.
3. Select the group.
4. Right-click and select **Properties** and add the item.

API request
To add an item to a VMware-based protection group by using the NetWorker REST API:

```
POST /nwrestapi/v2/global/protectiongroups/ApiGroup/op/
updatevmwareworkitems
{
  "addWorkItems": {
    "vCenterHostname": "hostname.company.com",
    "vmUuids": [
      "5008edba-edfe-127c-1d00-68830b14b0fc"
    ]
  }
}
```

Resources such as `/vmware/vcenters/{vcenter-hostname}/vms`, which returns a list of VMs on the specified vCenter, can be used to retrieve the UUID for a VM.

API response
The following response is received from the API:

```
204 No Content
```

Performing a VMware-based on-demand backup with an override

This section demonstrates how to perform a VMware-based on-demand backup with an override by using the NetWorker REST API.

NMC example
With NMC, you can run a backup only for a specific VM, container, or subset of VMs in a container:

1. Navigate to the **Protection** tab.
2. Double-click **VMware View**.
3. Select a VM or container associated with a protection group.
4. Right-click and select **Start Individual Client...**.
5. The **Start Workflow** dialog will appear, where you can verify or adjust the selection.

API request
The same on-demand backup is created in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v2/global/protectionpolicies/ApiPolicy/workflows/
ApiWorkflow/op/backup
{
  "vmwareWorkItemSelection" : {
    "vCenterHostname" : "hostname.company.com",
    "vmUuids" : ["50085479-6e0f-2383-7cb8-15883df49fde"]
  }
}
```

API response
The following response is received from the API:

```
201 Created
```

The API response will contain a Job URI in the location header. This URI can be used to monitor the job by using `GET jobs/{jobId}`, and to cancel it if necessary by using `POST /jobs/{jobId}/op/cancel`.

Performing an on-demand backup for a specific VM

This section demonstrates how to perform an on-demand backup for a specific VM by using the NetWorker REST API.

NMC example
To perform an on-demand backup for a specific VM in NMC:

1. Navigate to the **Protection** tab.
2. Double-click **VMware View**.
3. Select a VM or container associated with a protection group.

Performing a VMware-based on-demand backup with an override 47

4. Right-click and select **Start Individual Client...**.
5. The **Start Workflow** dialog will appear, where you can select a policy or workflow if the VM is protected by more than one.

API request
The same on-demand backup is created in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v2/global/vmware/vcenters/hostname.company.com/vms/
5008edba-edfe-127c-1d00-68830b14b0fc/op/backup
{
  "policy" : "ApiPolicy",
  "workflow" : "ApiWorkflow"
}
```

API response
The following response is received from the API:

```
201 Created
```

The API response will contain a Job URI in the location header. This URI can be used to monitor the job by using `GET jobs/{jobId}`, and to cancel it if necessary by using `POST /jobs/{jobId}/op/cancel`.

Recovering a VM to a previous state

This section demonstrates how to recover a VM to a previous state (revert) by using the NetWorker REST API.

NMC example
To recover a VM to a previous state (revert) in NMC:

1. Navigate to the **Recover** tab.
2. Right-click and select **New recover** and enter a name for the group.
3. In the **Recover** wizard, select **Virtual Machine Recovery**.
4. Select a vCenter.
5. Select the VM and backup to be recovered.
6. For **Virtual Machine Recovery Type**, select **Revert a Virtual Machine**.
   This example assumes that all disks will be recovered.

API request
The same recovery is performed in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v2/global/vmware/vcenters/hostname.company.com/
protectedvms/50085479-6e0f-2383-7cb8-15883df49fde/backups/
9524aeb0-00000006-f824e4c1-5824e4c1-00090c00-80c64a29/instances/
1478812865/op/recover
{
  "recoverMode" : "Revert",
  "powerOn": true,
  "reconnectNic": false
}
```

When making this API request, start with a backup (saveset) of the VM. This can be retrieved using `GET /backups/{backupId}/instances`.

API response

The following response is received from the API:

```
201 Created
```

The API response will contain a Job URI in the location header. This URI can be used to monitor the job by using `GET jobs/{jobId}`, and to cancel it if necessary by using `POST /jobs/{jobId}/op/cancel`.

Recovering a VM as a new VM

This section demonstrates how to recover a VM as a new VM by using the NetWorker REST API.

NMC example

To recover a VM as a new VM in NMC:

1. Navigate to the **Recover** tab.
2. Right-click and select **New recover** and enter a name for the group.
3. In the **Recover** wizard, select **Virtual Machine Recovery**.
4. Select a vCenter.
5. Select the VM and backup to be recovered.
6. For **Virtual Machine Recovery Type**, select **Virtual Machine Recovery**.
7. Select the location where the VM should be recovered.
8. Select the disks to be recovered.

API request

The same recovery is performed in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v2/global/vmware/vcenters/hostname.company.com/
protectedvms/50085479-6e0f-2383-7cb8-15883df49fde/backups/
9524aeb0-00000006-f824e4c1-5824e4c1-00090c00-80c64a29/op/recover
{
   "computeResourceMoref" : "domain-s7",
   "datacenterMoref": "datacenter-2",
   "datastoreMoref" : "datastore-11",
   "disks" : [
   {
      "datastoreMoref":"datastore-11",
      "key":"2000",
      "name":"Hard disk 1"
   },
   {
    "datastoreMoref":"datastore-11",
    "key":"2001",
    "name":"Hard disk 2"
  }
  ],
    "hostMoref": "host-9",
    "jobName": "jobName",
    "powerOn": false,
    "reconnectNic" : false,
    "recoverMode" : "New",
    "vCenterHostname" : "hostname.company.com",
    "vmName" : "newVmName"
}
```

This example requires information about the VMware recovery location and disk configuration, which can only be obtained from the vCenter outside of the API. If only recovery to the original VM location is required, most of the parameters in this example, including the disks, can be retrieved from the VM state recorded during the backup. This information is available in the `vmInformation` property of the backup object.

API response
The following response is received from the API:

```
201 Created
```

The API response will contain a Job URI in the location header. This URI can be used to monitor the job by using `GET jobs/{jobId}`, and to cancel it if necessary by using `POST /jobs/{jobId}/op/cancel`.

Browsing files and folders in a backup

This section describes the process of browsing files and folders in a backup.

To browse files and folders in a VMware backup, you need to perform the following steps in the defined sequence:

1. Refer Identifying VMware backup for browsing files and folders on page 50
2. Refer Mounting the backup on page 51
3. Refer Viewing the Job ID of a successful mounted VM on page 52
4. Refer Creating browsing session on page 53
5. Refer Viewing the browse session details on page 54
6. Refer Retrieving the content of current working directory of a backup on page 55
7. Refer Changing the current working directory of a browsing session on page 56

Identifying VMware backup for browsing files and folders

You can identify the VMware backup, by using API request, for browsing files and folders.

By using a API request
A `GET` request on either of these URI results responds with the list of available VMware backups and details of the backups. One of the backups can be identified for the files and folders recovery.

```
GET /vmware/vcenters/{vcenter-hostname}/protectedvms/{vm-uuid}/
backups
```

or

```
GET /vmware/vcenters/{vcenter-hostname}/protectedvms/{vm-uuid}/
backups/{backup-id}
```

Example 15 URI example to identify VMware backup for browsing files and folders

Example 15 URI example to identify VMware backup for browsing files and folders (continued)

```
https://10.118.252.115:9094/nwrestapi/v3/global/vmware/vcenters/
10.207.86.28/
protectedvms/500c2ae2-eb16-373e-ddc1-abdc61dd2625/backups?
fl=vmInformation
```

Example 16 API response example

```
{
  "backups": [
    {
      "vmInformation":
                        {
        "datastoreMoref": "datastore-107",
        "hostMoref": "host-94",
        "disks": [
          {
            "datastoreMoref": "datastore-107",
            "datastoreName": "remote",
            "key": "2000",
            "name": "Hard disk 1",
            "sizeInKb": 16777216,
            "thinProvisioned": false
          }
        ],
      "morefPath": "/datacenter-87/domain-s92/vm-122",
      "vCenterHostname": "10.207.86.28",
      "vmMoref": "vm-122",
      "vmName": "REHL7"
      }
    }
  ],
  "count": 1
}
```

Note

You need to note the values of the attributes `vmMoref` and `vmName` from the identified backups. These values are used in the VM mount request.

Mounting the backup

You can mount the identified backup to browse files or folders.

By using a API request

You can mount the VMware backup by using the following NetWorker REST API request:

```
POST /vmware/vcenters/{vcenter-hostname}/protectedvms/{vm-uuid}/
backups/{backup-id}/op/vmmount
```

Example 17 URI example to mount the backup

Example 17 URI example to mount the backup (continued)

```
https://10.207.86.34:9090/nwrestapi/v3/global/vmware/vcenters/
10.207.86.28/protectedvms
/500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-a6138c56/op/vmmount
```

Example 18 API request example

```
{
           "installFlrAgent": true,
           "targetVCenterHostname": "10.207.86.28",
           "targetVmAdminUserId": "administrator",
           "targetVmAdminUserPassword": "Password123!",
           "targetVmMoref": "vm-108",
           "targetVmName": "Win-host2",
           "targetVmUserId": "administrator",
           "targetVmUserPassword": "Password123!",
           "uninstallFlrAgent": false,

"vProxy":"10.207.86.35"

"

}
```

Note

You should note URI from the response header. The URI is in the following format:

```
/jobs/{jobId}
```

Viewing the Job ID of a successful mounted VM

Response of a successful mount operation contains Job reference link in the response header. Use a `GET` call on this Job reference URI to view the mounted session ID and the status of the mounting operation. To perform a VMware file level browse operation, value of attribute `vProxyMountState` should be mounted.

Note

You need to note the mount Job ID, which is needed for populating the attribute `mountJobId` in VMware FLR recovery.

By using a API request

To view the status of mounted session, use the following URI:

```
GET /jobs/{jobId}
```

Example 19 URI example to view the job ID of a successful mounted VM

Example 19 URI example to view the job ID of a successful mounted VM (continued)

```
https://10.207.86.34:9090/nwrestapi/v3/global/jobs/704689
```

Example 20 API response example

```
{
    "adhocJob": true,
    "command": "nsrvproxy_flr -c '10.207.86.28' -m 'vm-42' -v 
'Win2k12_host' -u 'Administrator'
                    -S '3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-a6138c56' -i -U 'administrator'",
    "dependentJobIds": [
        0
    ],
    "id": 704689,
    "itemIdLong": 704689,
    "links": [
        {
           "href": "https://10.207.86.34:9090/nwrestapi/v3/global/
jobs/704689/op/cancel",
           "title": "Cancel job"
        }
    ],
    "logFile": "C:\\Program Files\\EMC NetWorker\\nsr\\logs\\adhoc\
\adhoc-job-000002",
    "message": "End of mount request processing; mountpoint is 'C:\
\Program Files (x86)\\EMC\\
                Vproxy FLR Agent\\flr\\mountpoints\\FLR116562623' 
(no error)",
    "name": "nsrvproxy_flr",
    "ndmp": false,
    "parentJobId": 0,
    "previousJobId": 0,
    "rootParentJobId": 0,
    "runOnHost": "win2k12-ser",
    "saveSetId": "3356bee8-00000006-f00e8b12-5a0e8b12-00115000-
a6138c56",
    "siblingJobIds": [],
    "startTime": "2017-11-20T02:46:33-08:00",
    "state": "SessionActive",
    "stopped": false,
    "tenant": "",
    "type": "vproxyflr mount job",
    "vProxyErrorMessages": [],
    "vProxyFlrAgentInstallRequired": false,
    "vProxyHostname": "10.207.86.35",
    "vProxyMountSessionId": "4d8e3404-bb5e-44cc-806b-b69e1b0d4544",
    "vProxyMountState": "Mounted",
    "vProxyMountTargetVmInvalidAdminUser": false,
    "vProxyMountTargetVmInvalidUser": false,
    "vProxyMountTargetVmMoref": "vm-42",
    "vProxyMountTargetVmName": "Win2k12_host",
    "vProxyMountTargetVmUserId": "Administrator",
    "vProxyMountVCenterHostname": "10.207.86.28"
}
```

Creating browsing session

You can create a VMware browsing session by using API request.

By using API request

To browse specific files or folders, use the following URI:

```
POST /vmware/vcenters/{vcenter-hostname}/protectedvms/{vm-uuid}/
backups/{backup-id}/op/
vmmount/{vproxy-mount-session-id}/vmbrowse
```

Example 21 URI example to create browsing session

```
https://10.207.86.34:9090/nwrestapi/v3/global/vmware/vcenters/
10.207.86.28/protectedvms/
500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-a6138c56/
op/vmmount/4d8e3404-bb5e-44cc-806b-b69e1b0d4544/vmbrowse
```

Example 22 API request example

```
{
"currentWorkingDirectory": "01-E$"
}
```

Viewing the browse session details

You can view the details of browse session by using API request.

By using a API request

To view the status of browse operation, use the following NetWorker REST API request:

```
GET /vmware/vcenters/{vcenter-hostname}/protectedvms/{vm-uuid}/
backups/{backup-id}/op/
vmmount/{vproxy-mount-session-id}/vmbrowse/{vproxy-browse-session-
id}
```

Response of `GET` has the link to view the contents of the current working directory of the backup

Example 23 URI example to view the browse session details

```
https://10.207.86.34:9090/nwrestapi/v3/global/vmware/vcenters/
10.207.86.28/protectedvms
/500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-a6138c56
/op/vmmount/4d8e3404-bb5e-44cc-806b-b69e1b0d4544/vmbrowse/
cbe0c2be-5979-4cba-8beb-d7a0af6f61d8
```

Example 24 API response example

Example 24 API response example (continued)

```
{
    "config": {
        "sessionId": "cbe0c2be-5979-4cba-8beb-d7a0af6f61d8",
        "browseDestination": "false",
        "cacheRetentionSeconds": "60",
        "currentWorkingDirectory": "01-E$",
        "idleTimeout": "300"
    },
    "links": [
        {
            "href": "https://10.207.86.34:9090/nwrestapi/v3/global/
vmware/vcenters/10.207.86.28
/protectedvms/500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/
3356bee8-00000006-f00e8b12
-5a0e8b12-00115000-a6138c56/op/vmmount/4d8e3404-bb5e-44cc-806b-
b69e1b0d4544/vmbrowse/cbe0c2be-5979
-4cba-8beb-d7a0af6f61d8/contents",
            "title": "VM browse contents"
        }
    ],
  "status": {
     "state": "Success",
     "description": "Fetched directory contents of 'C:\\Program
Files (x86)\\EMC\\Vproxy FLR Agent
\\flr\\mountpoints\\FLR116562623\\01-E$'"
    }
}
```

Retrieving the content of current working directory of a backup

You can retrieve the content of the current working directory of browsing session by using API request.

By using a API request

To retrieve the content of the current working directory of a backup, use the following NetWorker REST API request:

```
GET /vmware/vcenters/{vcenter-hostname}/protectedvms/{vm-uuid}/
backups/{backup-id}/op/
vmmount/{vproxy-mount-session-id}/vmbrowse/{vproxy-browse-session-
id}/contents
```

Example 25 URI example to retrieve the content of current working directory of a backup

```
https://10.207.86.34:9090/nwrestapi/v3/global/vmware/vcenters/
10.207.86.28/protectedvms/
500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-a6138c56
/op/vmmount/4d8e3404-bb5e-44cc-806b-b69e1b0d4544/vmbrowse/
cbe0c2be-5979-4cba-8beb-d7a0af6f61d8/contents
```

Example 26 API response example

Example 26 API response example (continued)

```
{
  "error": {
    "text": "",
    "code": "0"
  },
  "noMoreData": "true",
  "totalObjects": "26",
  "freeSpace": "1915236352",
  "fileList": [
    {
      "fileName": "etc",
      "fileType": "directory",
      "fileSize": "0",
      "fileModificationTime": "2018-03-20T05:25:46Z"
    },
    {
      "fileName": "recover",
      "fileType": "directory",
      "fileSize": "0",
      "fileModificationTime": "2018-02-07T10:33:27Z"
    }
  ]
}
```

Changing the current working directory of a browsing session

You can change the current working directory of browsing session by using API request.

By using a API request

To change the current working directory of a browsing session, use the following NetWorker REST API request:

```
PUT /vmware/vcenters/{vcenter-hostname}/protectedvms/{vm-uuid}/
backups/{backup-id}/op/
vmmount/{vproxy-mount-session-id}/vmbrowse/{vproxy-browse-session-
id}
```

Example 27 URI example to change the current working directory of a browsing session

```
https://10.207.86.34:9090/nwrestapi/v3/global/vmware/vcenters/
10.207.86.28/protectedvms
/500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-a6138c56
/op/vmmount/4d8e3404-bb5e-44cc-806b-b69e1b0d4544/vmbrowse/
cbe0c2be-5979-4cba-8beb-d7a0af6f61d8
```

Example 28 API request example

Example 28 API request example (continued)

```
{
"currentWorkingDirectory": "01-E$\\demo"
}
```

Browsing the Destination VM

This section demonstrates how to browse a destination VM by using NetWorker REST API.

Creating a new resource
You can create a new resource by using NetWorker REST API.

By using API request

```
POST  https://10.207.86.34:9090/nwrestapi/v3/global/vmware/vcenters/
10.207.86.28/protectedvms/
500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-
a6138c56/op/vmmount/4d8e3404-bb5e-44cc-806b-b69e1b0d4544/vmbrowse
```

Example 29 API request example

```
{
  "browseDestination": true,
  "osType": "Windows"
}
```

Updating an existing resource
You can update an existing resource by using NetWorker REST API.

By using API request

```
PUT  https://10.207.86.34:9090/nwrestapi/v3/global/vmware/vcenters/
10.207.86.28/protectedvms/
500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-
a6138c56/op/vmmount/4d8e3404-bb5e-44cc-806b-b69e1b0d4544/vmbrowse/
cbe0c2be-5979-4cba-
8beb-d7a0af6f61d8
```

Example 30 API request example

```
{
     “currentWorkingDirectory:” “C:\\”
     "browseDestination": true,
     "osType": "Windows"
}
```

Example 30 API request example (continued)

Browsing the destination VM

You can browse the destination VM by using NetWorker REST API.

By using API request

```
GET https://10.207.86.34:9090/nwrestapi/v3/global/vmware/vcenters/
10.207.86.28/protectedvms/
500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-a6138c56/op/
vmmount/4d8e3404-bb5e-44cc-806b-b69e1b0d4544/vmbrowse/
cbe0c2be-5979-4cba-8beb-d7a0af6f61d8/contents
```

Example 31 API request example

```
{
  "error": {
    "text": "",
    "code": "0"
  },
  "noMoreData": "true",
  "totalObjects": "15",
  "freeSpace": "10831069184",
  "fileList": [
    {
      "fileName": "demo",
      "fileType": "directory",
      "fileSize": "0",
      "fileModificationTime": "2018-03-06T06:32:27Z"
    }
  ]
}
```

Creating VMware file level recovery to same virtual machine

This section describes the process to perform VMware File level recovery to same virtual machine (VM) that contains the backup, by using the NetWorker REST API.

To create VMware file level recovery to same VM, you need to perform the following in the defined sequence:

1. Refer Browsing files and folders in a backup on page 50
2. Refer Recovering specific files or folders from the identified backup to the same VM on page 58
3. Refer Viewing the status of recovery operation on page 60

Recovering specific files or folders from the identified backup to the same VM

You can initiate the VMware file level recovery by using API request.

By using API request

Response of the VMware file level recover `POST` operation contains a job reference link.

```
POST /vmware/vcenters/{vcenter-hostname}/protectedvms/{vm-uuid}/
backups/{backup-id}/op/recover
```

A `GET` request on this URI result responds with the status of the VMware recovery operation.

```
/jobs/{jobId}
```

Note

In highly loaded system, there is a remote possibility of job creation failure, if the job is not created within 10 seconds. In such cases, the POST response header returns the URI reference link of its own resource. Consumer has to monitor the URI reference link for recover Job ID is created in the response body

```
GET /recovers/{RecoverResourceID}
```

If URI reference link for recover Job ID is not created for some reason, consumer of the API might need to time out with appropriate message.

Example 32 URI example to recover specific files or folders from the identified backup to same VM

```
https://10.207.86.34:9090/nwrestapi/v3/global/vmware/vcenters/
10.207.86.28/protectedvms/
500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-a6138c56/op/recover
```

Example 33 API request example

```
{
"recoverMode": "FLR",
"vCenterHostname": "10.207.86.28",
"mountJobId": "704689",
"vmwareVmFlrOptions": {
     "terminateMountSession": "false",
     "overwrite": "true",
     "itemsToRecover":["01-E$\\demo\\A.txt","01-E$\\demo\\B.txt"],
     "recoveryDestination": "C:\\target"
     }
}
```

Example 34 API response example

Example 34 API response example (continued)

```
https://10.207.86.34:9090/nwrestapi/v3/global/jobs/704997
```

Viewing the status of recovery operation

You can view the status of VMware file level recovery by using API request.

By using API request

Use `GET` request on the Job URI with recover Job ID to view the status of VMware file level recovery.

```
GET /jobs/{jobId}
```

Example 35 URI example to view the status of recovery operation

```
https://10.207.86.34:9090/nwrestapi/v3/global/jobs/704997
```

Example 36 API response example

```
{
"adhocJob": true,
"command":"nsrvproxy_flr_recover -m 704689 -s 10.207.86.34 -d \"C:\\
\\target\" -I - -f",
"completionStatus": "Succeeded",
"dependentJobIds": [
        0
    ],
"endTime": "2017-11-20T03:07:05-08:00",
"exitCode": 0,
"id": 704997,
"itemIdLong": 704997,
"links": [],
"logFile": "C:\\Program Files\\EMC NetWorker\\nsr\\logs\\adhoc\
\nsrvproxy_flr_recover\\704997.log",
"message": "suppressed 18275 bytes of output.
\n159373:nsrvproxy_flr_recover:
vProxy Log: 2017/11/20 03:06:46 INFO:
[9101]
--------------------------------------------------------------------
----------\r\r\
n159373:nsrvproxy_flr_recover: vProxy Log: 2017/11/20 03:06:46
INFO:   [9101]
Total    Copied   Skipped  Mismatch    FAILED    Extras\r\r
\n159373:nsrvproxy_flr_recover:
vProxy Log: 2017/11/20 03:06:46 INFO:   [9101]     Dirs :
1         0         0         0
0         0\r\r\n159373:nsrvproxy_flr_recover: vProxy Log:
2017/11/20 03:06:46 INFO:   [9101]
Files :         1         1         0         0         0
0\r\r\
n159373:nsrvproxy_flr_recover: vProxy Log: 2017/11/20 03:06:46
INFO:   [9101]    Bytes :
23        23         0         0         0         0\r\r
```

Example 36 API response example (continued)

```
\n159373:nsrvproxy_flr_recover:
vProxy Log: 2017/11/20 03:06:46 INFO:   [9101]    Times :
0:00:00   0:00:00
0:00:00   0:00:00\r\r\n159373:nsrvproxy_flr_recover: vProxy Log:
2017/11/20 03:06:46 INFO: [9101]    Speed :              23000
Bytes/sec.\r\r\n159373:nsrvproxy_flr_recover: vProxy Log:
2017/11/20 03:06:46 INFO:   [9101]    Speed : 1.316 MegaBytes/min.\r
\r\n159373:
nsrvproxy_flr_recover: vProxy Log: 2017/11/20 03:06:46 INFO:
[9101]
Ended : Monday, November 20, 2017 3:06:46 AM\r\r
\n159373:nsrvproxy_flr_recover:
vProxy Log: 2017/11/20 03:06:46 INFO:   [9101] \r\r
\n159373:nsrvproxy_flr_recover:
vProxy Log: 2017/11/20 03:06:46 INFO:   [9101] \r\r
\n159373:nsrvproxy_flr_recover: vProxy Log:
2017/11/20 03:06:46 INFO:   [9101] File copying completed.\r\r
\n159373:nsrvproxy_flr_recover:
vProxy Log: 2017/11/20 03:06:46 TRACE:  [9101] C:\\Program Files
(x86)\\EMC\\vProxy FLR Agent
\\bin\\vflrcopy.exe exiting with status code 0\r\r
\n152793:nsrvproxy_flr_recover:
vProxy Log Ends ===============================================\r
\r\n162217:
nsrvproxy_flr_recover: FLR recover job completed successfully.\r\r
\n",
  "name": "VMWARE_FLR_4584_373032369185415",
  "ndmp": false,
  "parentJobId": 704996,
  "previousJobId": 0,
  "rootParentJobId": 704996,
  "runOnHost": "10.207.86.34",
  "siblingJobIds": [],
  "startTime": "2017-11-20T03:06:39-08:00",
  "state": "Completed",
  "stopped": true,
  "tenant": "",
  "terminationReason": "vProxy FLR recover job completed
successfully.",
  "type": "recover job"
```

Creating VMware file level recovery to other virtual machine

This section describes the process to perform VMware file level recovery to other virtual machine. The process is similar to the process of creating VMware file level recovery to same virtual machine (VM). The only difference is that you need to mount the backup to a different target VM by selecting correct values for the attributes `targetVmMoref`, `targetVmName`, `targetVmUserId`, `targetVmUserPassword`, `targetVmAdminUserId` and `targetVmAdminUserPassword`

Supporting Licenses View

This section describes the process to view the configured network licenses, by using the NetWorker REST API.

Viewing the configured network licenses
The networker licenses can be viewed using the URI `/licenses`.

By using API request

```
GET /licenses
```

Example 37 API request example

```
https://10.31.227.66:9090/nwrestapi/v3/global/licenses
```

Example 38 API response example

```
{
  "count": 2,
  "licenses": [
    {
      "enablerCode": "none",
      "expirationDate": "2018-03-15T00:00:00-07:00",
      "hostId": "a99310e9",
      "licenseType": "E10",
      "links": [
        {
          "href": "https://10.31.227.174:9090/nwrestapi/v3/global/
licenses/33.0.216.26.0.0.0.0.38.106.51.90.10.31.227.174",
          "rel": "item"
        }
      ],
      "name": "NetWorker/10 Eval",
      "resourceId": {
        "id": "33.0.216.26.0.0.0.0.38.106.51.90.10.31.227.174",
        "sequence": 1
      }
    },
    {
      "comment": "testlicense",
      "enablerCode": "79fefb-3385a4-050bc3",
      "expirationDate": "2018-04-12T00:00:00-07:00",
      "hostId": "a99310e9",
      "licenseType": "D15",
      "links": [
        {
          "href": "https://10.31.227.174:9090/nwrestapi/v3/global/
licenses/43.0.108.33.0.0.0.0.227.85.67.90.10.31.227.174",
          "rel": "item"
        }
      ],
      "name": "NetWorker Module for SAP, UNIX Client/1",
      "resourceId": {
        "id": "43.0.108.33.0.0.0.0.227.85.67.90.10.31.227.174",
        "sequence": 1
      }
    }
  ]
}
```

Viewing the configured networker license for the given Resource ID

A configured networker license whose resource id is known can be viewed using the URI `/licenses/{licenseId}`

By using API request

```
GET /licenses/{licenseId}
```

Example 39 API request example

```
https://10.31.227.66:9090/nwrestapi/v3/global/licenses/
33.0.133.27.0.0.0.0.165.61.212.89.10.31.227.66
```

Example 40 API response example

```
{
  "comment": "testlicense",
  "enablerCode": "79fefb-3385a4-050bc3",
  "expirationDate": "2018-04-12T00:00:00-07:00",
  "hostId": "a99310e9",
  "licenseType": "D15",
  "name": "NetWorker Module for SAP, UNIX Client/1",
  "resourceId": {
    "id": "43.0.108.33.0.0.0.0.227.85.67.90.10.31.227.174",
    "sequence": 1
  }
}
```

Supporting Plugin Installer

This section describes the process to install EBR or VC plugins, by using the NetWorker REST API.

Installing the EBR or VC plugins in the vCenter
The NetWorker REST API supports the installation of the EBR or VC plugin in the vCenter using the URI `vmware/vcenters/{vcenter-id}/plugins`.

The mandatory attributes are `pluginType, nwUserId`, and `nwPassword`. The optional attributes are`httpPort (80)`and `httpsPort (443)`.

By using API request
Following URI helps you to install EBR or VC plugin:

```
POST /vmware/vcenters/{vcenter-id}/plugins
```

Example 41 API request example for VC plugin type

```
POST https://10.31.227.154:9090/nwrestapi/v3/global/vmware/vcenters/
10.31.183.69/plugins
{
  "pluginType": "VC",
  "nwUserId": "administrator",
```

Example 41 API request example for VC plugin type (continued)

```
  "nwPassword": "Changeme@123",
  "httpPort": 80,
  "httpsPort": 443
}
```

Example 42 API response example for VC plugin type

```
200 OK
{
"resultCode": "Success",
"data": "Registration Success",
"debug": "",
"responseString": null
}
```

Example 43 API request example for EBR plugin type

```
POST https://10.31.227.154:9090/nwrestapi/v3/global/vmware/vcenters/
10.31.183.69/plugins
{
  "pluginType": "EBR",
  "nwUserId": "administrator",
  "nwPassword": "Changeme@123",
  "httpPort": 80,
  "httpsPort": 443
}
```

Example 44 API response example for EBR plugin type

```
200 OK
{
"@class": "com.emc.nw.model.parameters.base.NwOutput",
"status": "SUCCESS",
"messages":{},
"output":
{
"errorMessage": null,
"registered": true
}
}
```

Viewing CloudBoost Appliances

This section describes the process to view the CloudBoost appliances, by using the NetWorker REST API.

Viewing the list of CloudBoost appliances
You can view the CloudBoost appliances by using the URI `/cloudboostappliances`

By using API request
Following URI helps you to view the CloudBoost appliances:

```
GET /cloudboostappliances
```

Example 45 API request example

```
GET https://10.31.227.154:9090/nwrestapi/v3/global/
cloudboostappliances
```

Example 46 API response example

```
{
 "cloudBoostAppliances":
 [
  {
   "host": "10.234.165.138",
   "links":
    [
     {
      "href": "https://10.234.165.139:9090/nwrestapi/v3/global/
cloudboostappliances/
45.0.248.7.0.0.0.0.156.112.46.90.10.234.165.139",
      "rel": "item"
     }
    ],
   "name": "10.234.165.138",
   "password": "*******",
   "resourceId":{"id":
"45.0.248.7.0.0.0.0.156.112.46.90.10.234.165.139", "sequence": 7},
   "username": "maginatics"
   }
 ],
 "count": 1
}
```

Viewing the configured NetWorker CloudBoost appliance for a given Resource ID
You can view the configured NetWorker CloudBoost appliance, whose resource ID is known, using the URI `/cloudboostappliances/{cloudboostapplianceId}`

By using API request

```
GET /cloudboostappliances/{cloudboostapplianceId}
```

Example 47 API request example

```
GET https://10.234.165.139:9090/nwrestapi/v3/global/
cloudboostappliances/45.0.248.7.0.0.0.0.156.112.46.90.10.234.165.139
```

Example 48 API response example

```
{
 "host": "10.234.165.138",
 "name": "10.234.165.138",
 "password": "*******",
 "resourceId":
 {
  "id": "45.0.248.7.0.0.0.0.156.112.46.90.10.234.165.139",
  "sequence": 7
 },
 "username": "maginatics"
}
```

Viewing the job log file

You can view the job log file by using `GET /jobs/{jobId}/log` .

API request
Following API request helps you to view the log file associated with the given job:

```
GET /jobs/{jobId}/log
```

API request example

```
GET https://10.31.227.154:9090/nwrestapi/v3/global/jobs/22/log
```

API response example

```
200 OK
savefs ncdqd0154: succeeded
```

CHAPTER 3

Supporting Query Filter

This chapter includes the following topics:

Overview of Networker REST API Query Filter Support

This section describes the types of query filters supported by NetWorker REST API.

NetWorker REST API URI supports two types of filters:

- Query filter (q)
- Field List parser (fl)

Using Query Filter

This section demonstrates how to use query filter (q) by using NetWorker REST API. Query filter searches for specified key:value in resource list. The **`[Value]`** should be URL encoded and double quoted.

By using URI
To use the query filter (q), use the following URI structure:

```
https://[Endpoint][Resource]?q=[AttributeName]:[Value]
```

Example 49 API request example for query filter

```
https://10.207.84.200:9090/nwrestapi/v3/global/clients?
q=hostname:ISILON
{
 "clients":
  [
   {
     "aliases":
      [
        "ISILON"
      ],
      "applicationInformation":
       [
         "BUTYPE=dump",
         "DIRECT=Y",
         "HIST=Y",
         "UPDATE=Y",
         "USE_TBB_IF_AVAILABLE=Y",
         "NSR_PS_DEBUG_LEVEL=0"
       ],
      "backupCommand": "nsrndmp_save -T dump",
      "backupType": "Filesystem",
      "blockBasedBackup": false,
      "checkpointEnabled": false,
      "clientId":
"d0f9d027-00000004-5a423a81-5a423b77-00025000-65061a56",
      "hostname": "ISILON",
      "indexBackupContent": false,
      "links":
       [
        {
         "href": "https://10.207.84.200:9090/nwrestapi/v3/global/
clients/160.0.113.53.0.0.0.0.77.58.66.90.10.207.84.200",
         "rel": "item"
        }
       ],
      "nasDevice": false,
```

Example 49 API request example for query filter (continued)

```
      "ndmp": true,
      "ndmpMultiStreamsEnabled": false,
      "ndmpVendorInformation":
       [
         "os_type=Isilon OneFS",
         "os_version=v8.0.0.0",
         "vendor_name=Isilon",
         "product_name=Isilon NDMP",
         "server_revision=2.3.0"
       ],
      "parallelSaveStreamsPerSaveSet": false,
      "parallelism": 4,
      "password": "*******",
      "protectionGroups":
       [
         "ndmp_pg"
       ],
      "remoteAccessUsers":
       [
       ],
      "remoteUser": "ndmp",
      "resourceId":
       {
         "id": "160.0.113.53.0.0.0.0.77.58.66.90.10.207.84.200",
         "sequence": 13
       },
      "saveSets":
       [
         "/ifs/ajay/kiran"
       ],
      "scheduledBackup": true,
      "storageNodes": [],
      "tags": []
   }
  ],
 "count": 1
}
```

Using Field List Filter

This section demonstrates how to use field list filter by using NetWorker REST API. Field list filter (fl) is used to display only specified attributes from response body.

By using URI
To use field list filter (fl), use the following structure for the request URI:

```
https://[Endpoint][Resource]?fl=field1,field2
```

Example 50 API request example for field list filter

```
https://10.207.84.200:9090/nwrestapi/v3/global/clients?
fl=hostname,backupType
{
  "clients":
   [
```

Example 50 API request example for field list filter (continued)

```
     {
       "backupType": "Filesystem",
       "hostname": "ISILON"
     },
     {
       "backupType": "Filesystem",
       "hostname": "centos67_base"
     },
     {
       "backupType": "Filesystem",
       "hostname": "centos67_base"
     },
     {
       "hostname": "centos67_base"
     },
     {
       "hostname": "centos67_base"
     }
   ],
  "count": 5
}
```

Using Query Filter and Field List Filter simultaneously

This section demonstrates how to use query and field list filter simultaneously using NetWorker REST API.

By using URI
To use query filter (q) and field list filter (fl) simultaneously, use the following structure for the request URI:

```
https://[Endpoint][Resource]?q=[AttributeName]:[Value]  &
fl=field1,field2
```

Example 51 API request examples for using both q and fl simultaneously

```
https://10.207.84.200:9090/nwrestapi/v3/global/backups?
q=clientHostname: ISILON and level:Incr&fl=name,shortId,level
{
    "backups": [
        {
            "level": "Incr",
            "name": "/ifs/ajay/kiran",
            "shortId": "4016477993"
        },
        {
            "level": "Incr",
            "name": "/ifs/ajay/kiran",
            "shortId": "3966232744"

    ],
    "count": 2
}
```

Example 51 API request examples for using both q and fl simultaneously (continued)

The following is an example for query using multiple conditions:

```
https://10.207.84.200:9090/nwrestapi/v3/global/clients?
q=aliases:"localhost6" and backupType:Filesystem

{
   "clients":[
      {
         "aliases":[
            "localhost6",
            "ncdqd039",
            "ncdqd039.coredev.com"
         ],
         "applicationInformation":[
            "NSR_PS_DEBUG_LEVEL=0"
         ],
         "backupType":"Filesystem",
         "blockBasedBackup":false,
         "checkpointEnabled":false,

"clientId":"ed956501-00000004-5bb5df2f-5bb5df2e000150001f1db356",
         "hostname":"10.207.84.200",
         "indexBackupContent":false,
         "links":[
            {
               "href":"https://10.207.84.200:9090/nwrestapi/v3/
global/clients/45.0.4.11.0.0.0.0.57.161.177.91.10.207.84.200",
               "rel":"item"
            }
         ],
         "nasDevice":false,
         "ndmp":false,
         "ndmpMultiStreamsEnabled":false,
         "ndmpVendorInformation":[

         ],
         "parallelSaveStreamsPerSaveSet":false,
         "parallelism":4,
         "protectionGroups":[
            "nw-group"
         ],
         "remoteAccessUsers":[

         ],
         "resourceId":{
            "id":"45.0.4.11.0.0.0.0.57.161.177.91.10.207.84.200",
            "sequence":17
         },
         "saveSets":[
            "C:\\"
         ],
         "scheduledBackup":true,
         "storageNodes":[
            "nsrserverhost"
         ],
         "tags":[

         ]
      }
   ],
   "count":1
}
```

The following is an example for using both q and fl simultaneously:

Example 51 API request examples for using both q and fl simultaneously (continued)

```
https://10.207.84.200:9090/nwrestapi/v3/global/backups?
q=clientHostname: ISILON and saveTime:
[" 2018-01-20T21:00:09-05:00" to "2018-01-22T21:00:09-05:00"]
&fl=saveTime
{
    "backups": [
        {
            "saveTime": "2018-01-22T21:00:09-05:00"
        },
        {
            "saveTime": "2018-01-21T21:00:09-05:00"
        }
    ],
    "count": 2
}
```

APPENDIX A

Troubleshooting

This appendix includes the following topics:

NetWorker REST API log files

The following table provides a summary of the log files available for the NetWorker REST API.

Table 5 NetWorker REST API log files

Component	File name and default location	Description
Installation log Tomcat Access log Apache Catalina log	Refer to the NetWorker Authentication Service logs for these log files.	The NetWorker REST API service is deployed in the same Apache Tomcat container as the NetWorker authentication service. The NetWorker REST API uses the same installation log, Tomcat Access log and Apache Catalina log with NetWorker Authentication service.
NetWorker REST API log	• Linux: `/nsr/logs/restapi/restapi.log` • Windows: C:\Program `Files\EMC NetWorker\nsr\logs\restapi\restapi.log`	Main NetWorker REST API log file.

NetWorker REST API log file management

The NetWorker REST API uses the `Logback` API to manage log files.

To modify how NetWorker REST API manages the `restapi.log` log file, edit the `logback.xml` file, which is found in the following locations:

- Linux: `/nsr/authc/webapps/nwrestapi/WEB-INF/classes`
- Windows: `C:\Program Files\EMC NetWorker\nsr\authc-server\tomcat\webapps\nwrestapi\WEB-INF\classes`

This section describes how to modify the commonly used log attributes in the logback.xml file. Logback project documentation provides more detailed information about each attribute in the logback.xml file.

Modifying the logging level
In the `logger` configuration element, the value of the level attribute defines the level of logging that the NetWorker REST API writes to the log files. By default, the NetWorker REST API sets the logging level to info and messages appear in the log files. There are five standard log levels: trace, debug, info, warn, error.

To change the logging level to error, modify the level attribute to appear as follows:

```
<logger name="com.emc.nw.webapi" level="error"/>
```

Modifying the rollover period

In the `rollingPolicy` configuration element, the `fileNamePattern` element can be used to define the rollover period of the `restapi.log` file. When the log file reaches the end of a rolling period, the NetWorker REST API renames the log file for archival purposes and creates new log file. The rollover period is inferred from the value of `fileNamePattern`. By default, the NetWorker REST API sets the rollover period to `monthly`.

To change the rollover period to daily, modify the `fileNamePattern` element to appear as follows:

```
<fileNamePattern>${logdir}/restapi_%d.log</fileNamePattern>
```

Please refer to Logback project document on how to modify the rollover period.

Modifying the number of rollover log files

In the `rollingPolicy` configuration element, the `maxHistory` element defines the number of restapi.log rollover log files that the NetWorker REST API maintains. It controls the maximum number of archive files to keep, deleting older files. By default, the NetWorker REST API maintains six rollover log files.