Migrate from ONTAPI to the REST API

ONTAP automation

NetApp
January 12, 2026

Table of Contents

Migrate from ONTAPI to the REST API

Migration considerations for the ONTAP REST API

The ONTAPI API (ZAPI) is the original set of proprietary calls included with the NetApp ONTAP software. The API is provided through the Network Manageability SDK and supports the automation of data storage administration and management tasks. If you are using ONTAPI, you should plan your migration to the ONTAP REST API to take advantage of the expanded ONTAP feature set available with the REST API.

Related information

- Understand the ONTAP automation options
- CPC-00410 Deferral of ONTAPI (ZAPI) End-of-Availability Announcement
- FAQs on ZAPI to ONTAP REST API transformation for CPC

General design differences

The ONTAP REST API and command line interface have a fundamentally different designs. The CLI commands and parameters do not map directly to the REST API calls. And even where there might be a similarity, the details of the input parameters can be different. For example, numeric units might be specified in bytes or using a suffix (such as KB). See Input variables controlling an API request and API reference for more information.

Data SVMs exposed through the REST API

ONTAP supports several types of storage virtual machines (SVMs). However, only the data SVMs are directly exposed through the ONTAP REST API. The configuration information describing the cluster and nodes is available through the REST API, however the cluster and nodes are not treated as separate SVMs.

Access the ONTAP CLI through the REST API

To assist ONTAPI API and CLI users in their transition to the ONTAP REST API, ONTAP provides a REST endpoint to access the ONTAP CLI. You can use this passthrough feature to execute any CLI command. Use of the REST endpoint is returned in AutoSupport data so NetApp can identify gaps in the REST API and make improvements in future ONTAP releases.

To issue a CLI command, you must make a REST API call that is properly formed based on rules regarding the following:

- Resource paths
- Field names
- HTTP methods

The base resource path for CLI access is `/private/cli`. Refer to the ONTAP API online documentation page for details about accessing the CLI through the REST API. NetApp also maintains a GitHub repository containing code samples and other helpful information. See ONTAP REST Python GitHub repository - CLI passthrough samples for more information.

Changes to SnapDiff availability in ONTAPI

Beginning with ONTAP 9.10.1, the SnapDiff v1 and v2 ONTAPI calls cannot be invoked. Any third-party application that invokes SnapDiff v1 or v2 ONTAPI calls will not function beginning with ONTAP 9.10.1. ONTAP users should verify that their backup application supports the SnapDiff v3 REST calls before upgrading to ONTAP 9.10.1.

SnapDiff API availability across ONTAP releases is defined as follows:

- ONTAP 9.7 and earlier releases: v1 and v2 (ONTAPI only)
- ONTAP 9.8 – 9.9.1: v1, v2 and v3 (both ONTAPI and REST API)
- ONTAP 9.10.1 and later: v3 only (REST API only)

Support was withdrawn at different points in each release. This includes ONTAP 9.10.1 P11 and later, 9.11.1 P7 and later, and 9.12.1 GA and later. See the ONTAP Release Notes for more information.

ONTAPI to ONTAP REST API mapping

The ONTAP REST API includes functionality that is equivalent to ONTAPI in most areas. NetApp provides documentation that describes the mapping from the ONTAPI calls to the equivalent REST API calls.

You can access the current ONTAP ONTAPI-to-REST mapping documentation online. There is also a version selector available to access previous versions of the documentation based on the ONTAP release.

Using performance counters with the ONTAP REST API

The ONTAP Counter Manager maintains extensive information about the performance of each ONTAP system. It exports this data as a set of *performance counters* you can use to assess the performance of your ONTAP system and help meet your performance goals.

Access the ONTAP performance counters

You can access the ONTAP performance counters using two different APIs as well as through the ONTAP command line interface.

> The ONTAP REST API is the preferred and strategic option when automating the administration of your ONTAP deployments.

ONTAPI API

The ONTAPI API is available with the NetApp Network Manageability SDK. When using ONTAPI, the performance counters are defined within a collection of objects. Each object corresponds to a physical or virtual component of the system. There can be one or more instances of each object based on the system configuration.

For example, if your ONTAP system has four physical disks, there will be four instances of the `disk` object, each with its own set of performance counters. You can use ONTAPI to access the individual counters for each disk instance.

ONTAP REST API

Beginning with ONTAP 9.11.1, you can also access the performance data through the REST API. In this case, the performance counters are organized in tables which are equivalent to the ONTAPI objects. Each table row is equivalent to an instance of an ONTAPI object.

For example, if your ONTAP system has four physical disks, the `disk` table will contain four rows. Each of the rows can be accessed individually and includes its own set of performance counters available as fields or columns in the row.

Prepare to use the REST API

You should prepare before using the ONTAP REST API to access the performance counters.

Performance counters organized in tables

A subset of the ONTAPI objects is available through the ONTAP REST API and presented as tables. For example, the ONTAPI **hostadapter** object is presented through the REST API as the **host_adpater** table. Each host adapter in the system is a row with its own set of performance counters.

Instance name	Performance counters					
host_adapter_1	total_read_ops_1	total_write_ops_1	bytes_read_1	bytes_written_1	max_link_data_rate_1	rscn_count_1
host_adapter_2	total_read_ops_2	total_write_ops_2	bytes_read_2	bytes_written_2	max_link_data_rate_2	rscn_count_2
host_adapter_3	total_read_ops_3	total_write_ops_3	bytes_read_3	bytes_written_3	max_link_data_rate_3	rscn_count_3

Summary of the REST endpoints

There are four major endpoints available to access the ONTAP performance counters and related tables.

> Each of the REST endpoints provides read-only access and only supports the **GET** HTTP method. See the API reference for more information.

- **/cluster/counter/tables**

  Returns a collection of counter tables and their schema definitions.

- **/cluster/counter/tables/{name}**

  Returns information about a single named counter table.

- **/cluster/counter/tables/{counter_name}/rows**

  Returns a collection of rows from a named counter table.

- **/cluster/counter/tables/{counter_name}/rows/{id}**

  Returns a specific row from a named counter table.

Migrating from ONTAPI to the REST API

NetApp provides extensive support for migrating your automation code from ONTAPI to the ONTAP REST API. This includes mapping documentation to identity the equivalent performance counter table available in the REST API for a given ONTAPI object.

You can access the current ONTAP performance counter mapping documentation online. There is also a version selector available to access previous versions of the documentation based on the ONTAP release.

Get started with the ONTAP REST API

The following examples illustrate how to use REST API to access the ONTAP performance counters. This includes retrieving a list of the available tables and exploring the table structure.

Before you begin

Review the following information before using the examples.

ONTAP credentials

You'll need an ONTAP administrator account including the password.

Cluster management IP

You'll need the cluster management IP address configured for your ONTAP system.

All API calls use the GET method

All of the examples included below can only be used to retrieve information with the HTTP GET method.

Variable substitution

Each curl example includes one or more variables as indicated with capitals and bracketed text. Make sure to replace these variables with actual values as appropriate for your environment.

Examples match endpoints

The sequence of examples below illustrates how to use the REST endpoints available for retrieving the performance counters. See Summary of the REST endpoints for more information.

Example 1: All performance counter tables

You can use this REST API call to discover all the available counter manager tables.

Curl example

```
curl --request GET --user admin:<PASSWORD>
'https://<ONTAP_IP_ADDRESS>/api/cluster/counter/tables'
```

JSON output example


```
{
  "records": [
    {
      "name": "copy_manager",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/copy_manager"
        }
      }
    },
    {
      "name": "copy_manager:constituent",
      "_links": {
        "self": {
          "href":
"/api/cluster/counter/tables/copy_manager%3Aconstituent"
        }
      }
    },
    {
      "name": "disk",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/disk"
        }
      }
    },
    {
      "name": "disk:constituent",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/disk%3Aconstituent"
        }
      }
    },
    {
      "name": "disk:raid_group",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/disk%3Araid_group"
        }
      }
    },
    {
```

```
      "name": "external_cache",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/external_cache"
        }
      }
    },
    {
      "name": "fcp",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/fcp"
        }
      }
    },
    {
      "name": "fcp:node",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/fcp%3Anode"
        }
      }
    },
    {
      "name": "fcp_lif",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/fcp_lif"
        }
      }
    },
    {
      "name": "fcp_lif:node",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/fcp_lif%3Anode"
        }
      }
    },
    {
      "name": "fcp_lif:port",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/fcp_lif%3Aport"
        }
      }
```

```
    },
    {
      "name": "fcp_lif:svm",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/fcp_lif%3Asvm"
        }
      }
    },
    {
      "name": "fcvi",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/fcvi"
        }
      }
    },
    {
      "name": "headroom_aggregate",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/headroom_aggregate"
        }
      }
    },
    {
      "name": "headroom_cpu",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/headroom_cpu"
        }
      }
    },
    {
      "name": "host_adapter",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/host_adapter"
        }
      }
    },
    {
      "name": "iscsi_lif",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/iscsi_lif"
```

```
        }
      }
    },
    {
      "name": "iscsi_lif:node",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/iscsi_lif%3Anode"
        }
      }
    },
    {
      "name": "iscsi_lif:svm",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/iscsi_lif%3Asvm"
        }
      }
    },
    {
      "name": "lif",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/lif"
        }
      }
    },
    {
      "name": "lif:svm",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/lif%3Asvm"
        }
      }
    },
    {
      "name": "lun",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/lun"
        }
      }
    },
    {
      "name": "lun:constituent",
      "_links": {
```

```
        "self": {
          "href": "/api/cluster/counter/tables/lun%3Aconstituent"
        }
      }
    },
    {
      "name": "lun:node",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/lun%3Anode"
        }
      }
    },
    {
      "name": "namespace",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/namespace"
        }
      }
    },
    {
      "name": "namespace:constituent",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/namespace%3Aconstituent"
        }
      }
    },
    {
      "name": "nfs_v4_diag",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/nfs_v4_diag"
        }
      }
    },
    {
      "name": "nic_common",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/nic_common"
        }
      }
    },
    {
```

```
      "name": "nvmf_lif",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/nvmf_lif"
        }
      }
    },
    {
      "name": "nvmf_lif:constituent",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/nvmf_lif%3Aconstituent"
        }
      }
    },
    {
      "name": "nvmf_lif:node",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/nvmf_lif%3Anode"
        }
      }
    },
    {
      "name": "nvmf_lif:port",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/nvmf_lif%3Aport"
        }
      }
    },
    {
      "name": "object_store_client_op",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/object_store_client_op"
        }
      }
    },
    {
      "name": "path",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/path"
        }
      }
```

```
    },
    {
      "name": "processor",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/processor"
        }
      }
    },
    {
      "name": "processor:node",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/processor%3Anode"
        }
      }
    },
    {
      "name": "qos",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/qos"
        }
      }
    },
    {
      "name": "qos:constituent",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/qos%3Aconstituent"
        }
      }
    },
    {
      "name": "qos:policy_group",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/qos%3Apolicy_group"
        }
      }
    },
    {
      "name": "qos_detail",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/qos_detail"
```

```
        }
      }
    },
    {
      "name": "qos_detail_volume",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/qos_detail_volume"
        }
      }
    },
    {
      "name": "qos_volume",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/qos_volume"
        }
      }
    },
    {
      "name": "qos_volume:constituent",
      "_links": {
        "self": {
          "href":
"/api/cluster/counter/tables/qos_volume%3Aconstituent"
        }
      }
    },
    {
      "name": "qtree",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/qtree"
        }
      }
    },
    {
      "name": "qtree:constituent",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/qtree%3Aconstituent"
        }
      }
    },
    {
      "name": "svm_cifs",
```

```
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/svm_cifs"
        }
      }
    },
    {
      "name": "svm_cifs:constituent",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/svm_cifs%3Aconstituent"
        }
      }
    },
    {
      "name": "svm_cifs:node",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/svm_cifs%3Anode"
        }
      }
    },
    {
      "name": "svm_nfs_v3",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/svm_nfs_v3"
        }
      }
    },
    {
      "name": "svm_nfs_v3:constituent",
      "_links": {
        "self": {
          "href":
"/api/cluster/counter/tables/svm_nfs_v3%3Aconstituent"
        }
      }
    },
    {
      "name": "svm_nfs_v3:node",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/svm_nfs_v3%3Anode"
        }
      }
```

```
    },
    {
      "name": "svm_nfs_v4",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/svm_nfs_v4"
        }
      }
    },
    {
      "name": "svm_nfs_v41",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/svm_nfs_v41"
        }
      }
    },
    {
      "name": "svm_nfs_v41:constituent",
      "_links": {
        "self": {
          "href":
"/api/cluster/counter/tables/svm_nfs_v41%3Aconstituent"
        }
      }
    },
    {
      "name": "svm_nfs_v41:node",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/svm_nfs_v41%3Anode"
        }
      }
    },
    {
      "name": "svm_nfs_v42",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/svm_nfs_v42"
        }
      }
    },
    {
      "name": "svm_nfs_v42:constituent",
      "_links": {
        "self": {
```

```
          "href":
"/api/cluster/counter/tables/svm_nfs_v42%3Aconstituent"
        }
      }
    },
    {
      "name": "svm_nfs_v42:node",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/svm_nfs_v42%3Anode"
        }
      }
    },
    {
      "name": "svm_nfs_v4:constituent",
      "_links": {
        "self": {
          "href":
"/api/cluster/counter/tables/svm_nfs_v4%3Aconstituent"
        }
      }
    },
    {
      "name": "svm_nfs_v4:node",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/svm_nfs_v4%3Anode"
        }
      }
    },
    {
      "name": "system",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/system"
        }
      }
    },
    {
      "name": "system:constituent",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/system%3Aconstituent"
        }
      }
    },
```

```
    {
      "name": "system:node",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/system%3Anode"
        }
      }
    },
    {
      "name": "token_manager",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/token_manager"
        }
      }
    },
    {
      "name": "volume",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/volume"
        }
      }
    },
    {
      "name": "volume:node",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/volume%3Anode"
        }
      }
    },
    {
      "name": "volume:svm",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/volume%3Asvm"
        }
      }
    },
    {
      "name": "wafl",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/wafl"
        }
```

```
      }
    },
    {
      "name": "wafl_comp_aggr_vol_bin",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/wafl_comp_aggr_vol_bin"
        }
      }
    },
    {
      "name": "wafl_hya_per_aggregate",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/wafl_hya_per_aggregate"
        }
      }
    },
    {
      "name": "wafl_hya_sizer",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/wafl_hya_sizer"
        }
      }
    }
  ],
  "num_records": 71,
  "_links": {
    "self": {
      "href": "/api/cluster/counter/tables"
    }
  }
}
```


Example 2: High-level information about a specific table

You can use this REST API call to display the description and metadata for a specific table. The output includes the purpose of the table and what type of data each performance counter contains. The **host_adapter** table is used in this example.

Curl example

```
curl --request GET --user admin:<PASSWORD>
'https://<ONTAP_IP_ADDRESS>/api/cluster/counter/tables/host_adapter'
```

JSON output example


```
{
  "name": "host_adapter",
  "description": "The host_adapter table reports activity on the Fibre
Channel, Serial Attached SCSI, and parallel SCSI host adapters the
storage system uses to connect to disks and tape drives.",
  "counter_schemas": [
    {
      "name": "bytes_read",
      "description": "Bytes read through a host adapter",
      "type": "rate",
      "unit": "per_sec"
    },
    {
      "name": "bytes_written",
      "description": "Bytes written through a host adapter",
      "type": "rate",
      "unit": "per_sec"
    },
    {
      "name": "max_link_data_rate",
      "description": "Max link data rate in Kilobytes per second for a
host adapter",
      "type": "raw",
      "unit": "kb_per_sec"
    },
    {
      "name": "node.name",
      "description": "System node name",
      "type": "string",
      "unit": "none"
    },
    {
      "name": "rscn_count",
      "description": "Number of RSCN(s) received by the FC HBA",
      "type": "raw",
      "unit": "none"
    },
    {
      "name": "total_read_ops",
      "description": "Total number of reads on a host adapter",
      "type": "rate",
      "unit": "per_sec"
    },
    {
```

```
      "name": "total_write_ops",
      "description": "Total number of writes on a host adapter",
      "type": "rate",
      "unit": "per_sec"
    }
  ],
  "_links": {
    "self": {
      "href": "/api/cluster/counter/tables/host_adapter"
    }
  }
}
```


Example 3: All rows in a specific table

You can use this REST API call to view all the rows in a table. This indicates what instances of the Counter Manager objects exist.

Curl example

```
curl --request GET --user admin:<PASSWORD>
'https://<ONTAP_IP_ADDRESS>/api/cluster/counter/tables/host_adapter/row
s'
```

JSON output example


```
{
  "records": [
    {
      "id": "dmp-adapter-01",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/host_adapter/rows/dmp-
adapter-01"
        }
      }
    },
    {
      "id": "dmp-adapter-02",
      "_links": {
        "self": {
          "href": "/api/cluster/counter/tables/host_adapter/rows/dmp-
adapter-02"
        }
      }
    }
  ],
  "num_records": 2,
  "_links": {
    "self": {
      "href": "/api/cluster/counter/tables/host_adapter/rows"
    }
  }
}
```


Example 4: Single row in a specific table

You can use this REST API call to view performance counter values for a specific counter manager instance in the table. In this example, the performance data for one of the host adapters is requested.

Curl example

```
curl --request GET --user admin:<PASSWORD>
'https://<ONTAP_IP_ADDRESS>/api/cluster/counter/tables/host_adapter/row
s/dmp-adapter-01'
```

JSON output example

```
{
  "counter_table": {
    "name": "host_adapter"
  },
  "id": "dmp-adapter-01",
  "properties": [
    {
      "name": "node.name",
      "value": "dmp-node-01"
    }
  ],
  "counters": [
    {
      "name": "total_read_ops",
      "value": 25098
    },
    {
      "name": "total_write_ops",
      "value": 48925
    },
    {
      "name": "bytes_read",
      "value": 1003799680
    },
    {
      "name": "bytes_written",
      "value": 6900961600
    },
    {
      "name": "max_link_data_rate",
      "value": 0
    },
    {
      "name": "rscn_count",
      "value": 0
    }
  ],
  "_links": {
    "self": {
      "href": "/api/cluster/counter/tables/host_adapter/rows/dmp-
adapter-01"
    }
  }
}
```

Tools and software supporting the ONTAP REST API

NetApp provides sample Python scripts and other related software to support your migration from ONTAPI to the ONTAP REST API. The most important of these samples are described below.

All the Python code samples are available at the NetApp ONTAP REST Python GitHub repository. You should also review the resources available in Learn more about the ONTAP REST API.

ONTAPI usage reporting tool

The ONTAPI usage reporting tool is designed to help NetApp professional services, customers, and partners identify the ONTAPI usage in their ONTAP environment. Scripts are provided for three different use cases as described in the table below.

Script	Description
apache_scraper.py	An Apache log scraper to find the ONTAPI calls issued against the ONTAP nodes
session_stats.py	A CLI script to retrieve session statistics data from ONTAP
zapi_to_rest.py	A script to extract the REST details of the ONTAPI calls and attributes passed

You can access the ONTAPI usage reporting tool to get started. Also see a Demo about the reporting tool and how to use it.

Private CLI passthrough

The REST API provides broad coverage of the features and facilities available with ONTAP. However, there may be instances when direct access to the ONTAP CLI through the REST API can be useful.

For an introduction to this feature, see Access the ONTAP CLI through the REST API. For the Python samples see REST CLI passthrough samples.

Python client library

The Python client library is a package you can install and use to access the ONTAP REST API with Python. It allows you to quickly develop robust code for the automation of your ONTAP deployments. To learn more about the Python client library, refer to Python client library.

ONTAP PowerShell Toolkit

The NetApp.ONTAP PowerShell Toolkit enhances your local PowerShell environment with a module that includes over 2,400 cmdlets. It allows you to quickly develop code for your Windows host to automate the ONTAP deployments. For more information, see Learn about the NetApp PowerShell Toolkit.

Copyright information

Trademark information