



Manage storage qtrees

REST API reference

NetApp
September 12, 2025

This PDF was generated from https://docs.netapp.com/us-en/ontap-restapi-9121/storage_qtrees_endpoint_overview.html on September 12, 2025. Always check docs.netapp.com for the latest.

Table of Contents

Manage storage qtrees	1
Storage qtrees endpoint overview	1
Overview	1
Qtree QoS policy	1
Qtree APIs	1
Examples	1
Retrieve qtrees	11
Expensive properties	12
Related ONTAP commands	12
Parameters	12
Response	16
Error	19
Definitions	19
Create a qtree in a FlexVol or FlexGroup volume	26
Required properties	26
Recommended optional properties	26
Related ONTAP commands	27
Parameters	27
Request Body	27
Response	30
Error	30
Definitions	31
Delete a qtree	38
Related ONTAP commands	38
Parameters	38
Response	39
Error	40
Definitions	41
Retrieve qtree properties	42
Expensive properties	43
Related ONTAP commands	43
Parameters	43
Response	43
Error	46
Definitions	47
Update properties for a qtree	53
Related ONTAP commands	53
Parameters	53
Request Body	54
Response	56
Error	57
Definitions	57

Manage storage qtrees

Storage qtrees endpoint overview

Overview

A qtree is a logically defined file system that can exist as a special subdirectory of the root directory within a FlexVol volume or a FlexGroup volume.

Qtree QoS policy

Qtree QoS policy and settings enforce Service Level Objectives (SLOs) on a qtree. SLOs can be set by specifying "qos_policy.max_throughput_iops" and/or "qos_policy.max_throughput_mbps" or "qos_policy.min_throughput_iops" and/or "qos_policy.min_throughput_mbps". Specifying "min_throughput_iops" or "min_throughput_mbps" is only supported on volumes hosted on a node that is flash optimized. A pre-created QoS policy can also be used by specifying "qos_policy.name" or "qos_policy.uuid" properties. Setting or assigning a QoS policy to a qtree is not supported if its containing volume or SVM has a QoS policy attached, or a file or LUN in its containing volume already has a QoS policy attached.

Qtree APIs

The following APIs are used to create, retrieve, modify, and delete qtrees.

– POST /api/storage/qtrees

– GET /api/storage/qtrees

– GET /api/storage/qtrees/{volume-uuid}/{qtree-id}

– PATCH /api/storage/qtrees/{volume-uuid}/{qtree-id}

– DELETE /api/storage/qtrees/{volume-uuid}/{qtree-id}

Examples

Creating a qtree inside a volume for an SVM

This API is used to create a qtree inside a volume for an SVM.

The following example shows how to create a qtree in a FlexVol volume with a given security style, user, group, UNIX permissions, an export policy, and a QoS policy.

```
# The API:  
POST /api/storage/qtrees  
  
# The call:  
curl -X POST 'https://<mgmt-ip>/api/storage/qtrees?return_records=true' -H  
'accept: application/hal+json' -d @test_qtree_post.txt
```

```

test_qtree_post.txt(body) :
{
  "svm": {
    "name": "svm1"
  },
  "volume": {
    "name": "fv"
  },
  "name": "qt1",
  "security_style": "unix",
  "user": {
    "name": "unix_user1"
  },
  "group": {
    "name": "unix_group1"
  },
  "unix_permissions": 744,
  "export_policy": {
    "name": "default"
  },
  "qos_policy": {
    "max_throughput_iops": 1000
  }
}

# The response:
{
  "num_records": 1,
  "records": [
    {
      "svm": {
        "name": "svm1"
      },
      "volume": {
        "name": "fv"
      },
      "name": "qt1",
      "security_style": "unix",
      "user": {
        "name": "unix_user1"
      },
      "group": {
        "name": "unix_group1"
      },
      "unix_permissions": 744,
      "export_policy": {

```

```

        "name": "default"
    },
    "qos_policy": {
        "min_throughput_iops": 0,
        "min_throughput_mbps": 0,
        "max_throughput_iops": 1000,
        "max_throughput_mbps": 0,
        "uuid": "39ac471f-ff35-11e9-b0f9-005056a7ab52",
        "name": "vs0_auto_gen_policy_39a9522f_ff35_11e9_b0f9_005056a7ab52"
    },
    "_links": {
        "self": {
            "href": "/api/storage/qtrees/?volume.name=fv&name=qt1"
        }
    }
},
"job": {
    "uuid": "84edef3c-4f6d-11e9-9a71-005056a7f717",
    "_links": {
        "self": {
            "href": "/api/cluster/jobs/84edef3c-4f6d-11e9-9a71-005056a7f717"
        }
    }
}
}

```

Retrieving qtrees

This API is used to retrieve qtrees.

The following example shows how to retrieve qtrees belonging to SVM **svm1** and volume **fv**. The `svm.name` and `volume.name` query parameters are used to find the required qtrees.

```

# The API:
GET /api/storage/qtrees

# The call:
curl -X GET "https://<mgmt-
ip>/api/storage/qtrees/?svm.name=svm1&volume.name=fv" -H 'accept:
application/hal+json'

# The response

```

```
{
  "records": [
    {
      "svm": {
        "uuid": "b68f961b-4cee-11e9-930a-005056a7f717",
        "name": "svml",
        "_links": {
          "self": {
            "href": "/api/svm/svms/b68f961b-4cee-11e9-930a-005056a7f717"
          }
        }
      },
      "volume": {
        "uuid": "cb20da45-4f6b-11e9-9a71-005056a7f717",
        "name": "fv",
        "_links": {
          "self": {
            "href": "/api/storage/volumes/cb20da45-4f6b-11e9-9a71-005056a7f717"
          }
        }
      },
      "id": 0,
      "name": "",
      "_links": {
        "self": {
          "href": "/api/storage/qtrees/cb20da45-4f6b-11e9-9a71-005056a7f717/0"
        }
      }
    },
    {
      "svm": {
        "uuid": "b68f961b-4cee-11e9-930a-005056a7f717",
        "name": "svml",
        "_links": {
          "self": {
            "href": "/api/svm/svms/b68f961b-4cee-11e9-930a-005056a7f717"
          }
        }
      },
      "volume": {
        "uuid": "cb20da45-4f6b-11e9-9a71-005056a7f717",
        "name": "fv",
        "_links": {
          "self": {
            "href": "/api/storage/volumes/cb20da45-4f6b-11e9-9a71-005056a7f717"
          }
        }
      }
    }
  ]
}
```

```

        "href": "/api/storage/volumes/cb20da45-4f6b-11e9-9a71-
005056a7f717"
    }
}
},
"id": 1,
"name": "qt1",
"_links": {
    "self": {
        "href": "/api/storage/qtrees/cb20da45-4f6b-11e9-9a71-
005056a7f717/1"
    }
}
},
{
"svm": {
    "uuid": "b68f961b-4cee-11e9-930a-005056a7f717",
    "name": "svml",
    "_links": {
        "self": {
            "href": "/api/svm/svms/b68f961b-4cee-11e9-930a-005056a7f717"
        }
    }
},
"volume": {
    "uuid": "cb20da45-4f6b-11e9-9a71-005056a7f717",
    "name": "fv",
    "_links": {
        "self": {
            "href": "/api/storage/volumes/cb20da45-4f6b-11e9-9a71-
005056a7f717"
        }
    }
},
"id": 2,
"name": "qt2",
"_links": {
    "self": {
        "href": "/api/storage/qtrees/cb20da45-4f6b-11e9-9a71-
005056a7f717/2"
    }
}
},
"num_records": 3,
"_links": {

```

```

    "self": {
      "href": "/api/storage/qtrees/?svm.name=svml&volume.name=fv"
    }
  }
}

```

Retrieving properties of a specific qtree using a qtree identifier

This API is used to retrieve properties of a specific qtree using qtree.id.

The following example shows how to use the qtree identifier to retrieve all properties of the qtree using the fields query parameter.

```

# The API:
GET /api/storage/qtrees/{volume.uuid}/{id}

# The call:
curl -X GET 'https://<mgmt-ip>/api/storage/qtrees/cb20da45-4f6b-11e9-9a71-
005056a7f717/2?fields=' -H 'accept: application/hal+json'
{
  "svm": {
    "uuid": "b68f961b-4cee-11e9-930a-005056a7f717",
    "name": "svml",
    "_links": {
      "self": {
        "href": "/api/svm/svms/b68f961b-4cee-11e9-930a-005056a7f717"
      }
    }
  },
  "volume": {
    "uuid": "cb20da45-4f6b-11e9-9a71-005056a7f717",
    "name": "fv",
    "_links": {
      "self": {
        "href": "/api/storage/volumes/cb20da45-4f6b-11e9-9a71-
005056a7f717"
      }
    }
  },
  "id": 2,
  "name": "qt2",
  "security_style": "unix",
  "user": {

```

```
        "name": "unix_user1"
    },
    "group": {
        "name": "unix_group1"
    },
    "unix_permissions": 744,
    "export_policy": {
        "name": "default",
        "id": 12884901889,
        "_links": {
            "self": {
                "href": "/api/protocols/nfs/export-policies/12884901889"
            }
        }
    },
    "qos_policy": {
        "min_throughput_iops": 0,
        "min_throughput_mbps": 0,
        "max_throughput_iops": 1000,
        "max_throughput_mbps": 0,
        "uuid": "39ac471f-ff35-11e9-b0f9-005056a7ab52",
        "name": "vs0_auto_gen_policy_39a9522f_ff35_11e9_b0f9_005056a7ab52",
        "_links": {
            "self": {
                "href": "/api/storage/qos/policies/39ac471f-ff35-11e9-b0f9-005056a7ab52"
            }
        }
    },
    "statistics": {
        "timestamp": "2019-04-09T05:50:42Z",
        "status": "ok",
        "iops_raw": {
            "read": 0,
            "write": 0,
            "other": 3,
            "total": 3
        },
        "throughput_raw": {
            "read": 0,
            "write": 0,
            "other": 0,
            "total": 0
        }
    },
    "path": "/fv/qt2",
```

```

"nas": {
    "path": "/fv/qt2",
},
"_links": {
    "self": {
        "href": "/api/storage/qtrees/cb20da45-4f6b-11e9-9a71-005056a7f717/2"
    }
}
}

```

Retrieving properties of a specific qtree using the qtree name

This API is used to retrieve properties of a specific qtree using "qtree.name". The following example shows how to retrieve all of the properties belonging to qtree "qt2". The `svm.name` and `volume.name` query parameters are used here along with the qtree name.

```

# The API:
GET /api/storage/qtrees/

# The call:
curl -X GET 'https://<mgmt-
ip>/api/storage/qtrees/?svm.name=svm1&volume.name=fv&name=qt2&fields=*' -H
'accept: application/hal+json'
{
    "svm": {
        "uuid": "b68f961b-4cee-11e9-930a-005056a7f717",
        "name": "svm1",
        "_links": {
            "self": {
                "href": "/api/svm/svms/b68f961b-4cee-11e9-930a-005056a7f717"
            }
        }
    },
    "volume": {
        "uuid": "cb20da45-4f6b-11e9-9a71-005056a7f717",
        "name": "fv",
        "_links": {
            "self": {
                "href": "/api/storage/volumes/cb20da45-4f6b-11e9-9a71-
005056a7f717"
            }
        }
    },
}

```

```
"id": 2,
"name": "qt2",
"security_style": "unix",
"user": {
  "name": "unix_user1"
},
"group": {
  "name": "unix_group1"
},
"unix_permissions": 744,
"export_policy": {
  "name": "default",
  "id": 12884901889,
  "_links": {
    "self": {
      "href": "/api/protocols/nfs/export-policies/12884901889"
    }
  }
},
"qos_policy": {
  "min_throughput_iops": 0,
  "min_throughput_mbps": 0,
  "max_throughput_iops": 1000,
  "max_throughput_mbps": 0,
  "uuid": "39ac471f-ff35-11e9-b0f9-005056a7ab52",
  "name": "vs0_auto_gen_policy_39a9522f_ff35_11e9_b0f9_005056a7ab52",
  "_links": {
    "self": {
      "href": "/api/storage/qos/policies/39ac471f-ff35-11e9-b0f9-005056a7ab52"
    }
  }
},
"statistics": {
  "timestamp": "2019-04-09T05:50:42Z",
  "status": "ok",
  "iops_raw": {
    "read": 0,
    "write": 0,
    "other": 3,
    "total": 3
  },
  "throughput_raw": {
    "read": 0,
    "write": 0,
    "other": 0,
    "total": 0
  }
}
```

```
        "total": 0
    }
},
"_links": {
    "self": {
        "href": "/api/storage/qtrees/cb20da45-4f6b-11e9-9a71-005056a7f717/2"
    }
}
```

Updating a qtree

This API is used to update a qtree.

The following example shows how to update properties in a qtree.

```
# The API:
PATCH /api/storage/qtrees/{volume.uuid}/{id}

# The call:
curl -X PATCH 'https://<mgmt-ip>/api/storage/qtrees/cb20da45-4f6b-11e9-9a71-005056a7f717/2' -H 'accept: application/hal+json' -d
'@test_qtree_patch.txt'
test_qtree_patch.txt(body):
{
    "security_style": "mixed",
    "user": {
        "name": "unix_user1"
    },
    "group": {
        "name": "unix_group1"
    },
    "unix_permissions": 777,
    "export_policy": {
        "id": "9",
        "name": "exp1"
    },
    "qos_policy": {
        "uuid": "39ac471f-ff35-11e9-b0f9-005056a7ab53"
    }
}
```

Renaming a qtree

This API is used to rename a qtree.

The following example below shows how to rename a qtree with a new name.

```
# The API:  
PATCH /api/storage/qtrees/{volume.uuid}/{id}  
  
# The call:  
curl -X PATCH 'https://<mgmt-ip>/api/storage/qtrees/cb20da45-4f6b-11e9-  
9a71-005056a7f717/1' -H 'accept: application/hal+json' -d '{ "name":  
"new_qt1" }'
```

Deleting a qtree inside a volume of an SVM

This API is used to delete a qtree inside a volume of an SVM.

The following example shows how to delete a qtree.

```
# The API:  
DELETE /api/storage/qtrees/{volume.uuid}/{id}  
  
# The call:  
curl -X DELETE "https://<mgmt-ip>/api/storage/qtrees/cb20da45-4f6b-11e9-  
9a71-005056a7f717/2" -H 'accept: application/hal+json'
```

Retrieve qtrees

GET /storage/qtrees

Introduced In: 9.6

Retrieves qtrees configured for all FlexVol volumes or FlexGroup volumes.

Use the `fields` query parameter to retrieve all properties of the qtree. If the `fields` query parameter is not used, then GET returns the qtree `name` and qtree `id` only.

Expensive properties

There is an added computational cost to retrieving values for these properties. They are not included by default in GET results and must be explicitly requested using the `fields` query parameter. See [Requesting specific fields](#) to learn more.

- `statistics.*`

Related ONTAP commands

- `qtree show`

Parameters

Name	Type	In	Required	Description
filesystem_path	string	query	False	Filter by filesystem_path <ul style="list-style-type: none">• Introduced in: 9.10
user.id	string	query	False	Filter by user.id <ul style="list-style-type: none">• Introduced in: 9.9
user.name	string	query	False	Filter by user.name <ul style="list-style-type: none">• Introduced in: 9.9
svm.uuid	string	query	False	Filter by svm.uuid
svm.name	string	query	False	Filter by svm.name
id	integer	query	False	Filter by id <ul style="list-style-type: none">• Max value: 4994• Min value: 0
export_policy.name	string	query	False	Filter by export_policy.name
export_policy.id	integer	query	False	Filter by export_policy.id
name	string	query	False	Filter by name

Name	Type	In	Required	Description
volume.uuid	string	query	False	Filter by volume.uuid
volume.name	string	query	False	Filter by volume.name
statistics.status	string	query	False	Filter by statistics.status <ul style="list-style-type: none"> • Introduced in: 9.8
statistics.timestamp	string	query	False	Filter by statistics.timestamp <ul style="list-style-type: none"> • Introduced in: 9.8
statistics.iops_raw.total	integer	query	False	Filter by statistics.iops_raw.total <ul style="list-style-type: none"> • Introduced in: 9.8
statistics.iops_raw.read	integer	query	False	Filter by statistics.iops_raw.read <ul style="list-style-type: none"> • Introduced in: 9.8
statistics.iops_raw.write	integer	query	False	Filter by statistics.iops_raw.write <ul style="list-style-type: none"> • Introduced in: 9.8
statistics.iops_raw.other	integer	query	False	Filter by statistics.iops_raw.other <ul style="list-style-type: none"> • Introduced in: 9.8

Name	Type	In	Required	Description
statistics.throughput._raw.total	integer	query	False	Filter by statistics.throughput._raw.total <ul style="list-style-type: none"> • Introduced in: 9.8
statistics.throughput._raw.read	integer	query	False	Filter by statistics.throughput._raw.read <ul style="list-style-type: none"> • Introduced in: 9.8
statistics.throughput._raw.write	integer	query	False	Filter by statistics.throughput._raw.write <ul style="list-style-type: none"> • Introduced in: 9.8
statistics.throughput._raw.other	integer	query	False	Filter by statistics.throughput._raw.other <ul style="list-style-type: none"> • Introduced in: 9.8
group.id	string	query	False	Filter by group.id <ul style="list-style-type: none"> • Introduced in: 9.9
group.name	string	query	False	Filter by group.name <ul style="list-style-type: none"> • Introduced in: 9.9
qos_policy.max_throughput_iops	integer	query	False	Filter by qos_policy.max_throughput_iops <ul style="list-style-type: none"> • Introduced in: 9.8

Name	Type	In	Required	Description
qos_policy.min_thro_ughput_mbps	integer	query	False	Filter by qos_policy.min_thro_ughput_mbps <ul style="list-style-type: none"> • Introduced in: 9.8
qos_policy.max_thro_ughput_mbps	integer	query	False	Filter by qos_policy.max_thro_ughput_mbps <ul style="list-style-type: none"> • Introduced in: 9.8
qos_policy.min_thro_ughput_iops	integer	query	False	Filter by qos_policy.min_thro_ughput_iops <ul style="list-style-type: none"> • Introduced in: 9.8
qos_policy.name	string	query	False	Filter by qos_policy.name <ul style="list-style-type: none"> • Introduced in: 9.8
qos_policy.uuid	string	query	False	Filter by qos_policy.uuid <ul style="list-style-type: none"> • Introduced in: 9.8
path	string	query	False	Filter by path
nas.path	string	query	False	Filter by nas.path <ul style="list-style-type: none"> • Introduced in: 9.9
security_style	string	query	False	Filter by security_style
unix_permissions	integer	query	False	Filter by unix_permissions
fields	array[string]	query	False	Specify the fields to return.

Name	Type	In	Required	Description
max_records	integer	query	False	Limit the number of records returned.
return_records	boolean	query	False	The default is true for GET calls. When set to false, only the number of records is returned. <ul style="list-style-type: none"> Default value: 1
return_timeout	integer	query	False	The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached. <ul style="list-style-type: none"> Max value: 120 Min value: 0 Default value: 1
order_by	array[string]	query	False	Order results by specified fields and optional [asc]

Response

Status: 200, Ok

Name	Type	Description
_links	_links	
num_records	integer	Number of records
records	array[qtree]	

Example response

```
{  
  "_links": {  
    "next": {  
      "href": "/api/resourcelink"  
    },  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "num_records": 1,  
  "records": [  
    {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "export_policy": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "id": 100,  
        "name": "default"  
      },  
      "filesystem_path": "/dir1/qtree1",  
      "group": {  
        "id": "20001",  
        "name": "unix_group1"  
      },  
      "id": 1,  
      "name": "string",  
      "nas": {  
        "path": "/volume3/qtree1"  
      },  
      "path": "/volume3/qtree1",  
      "qos_policy": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "max throughput iops": 10000,  
        "min throughput iops": 1000,  
        "policy": {  
          "id": 1,  
          "name": "string",  
          "type": "string",  
          "value": "string"  
        },  
        "type": "string",  
        "value": "string"  
      }  
    }  
  ]  
}
```

```

        "max_throughput_mbps": 500,
        "min_throughput_iops": 2000,
        "min_throughput_mbps": 500,
        "name": "performance",
        "uuid": "1cd8a442-86d1-11e0-a1c-123478563412"
    },
    "security_style": "string",
    "statistics": {
        "iops_raw": {
            "read": 200,
            "total": 1000,
            "write": 100
        },
        "status": "ok",
        "throughput_raw": {
            "read": 200,
            "total": 1000,
            "write": 100
        },
        "timestamp": "2017-01-25T11:20:13Z"
    },
    "svm": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "name": "svml1",
        "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
    },
    "unix_permissions": 755,
    "user": {
        "id": "10001",
        "name": "unix_user1"
    },
    "volume": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "name": "volume1",
        "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
    }
}
]

```

```
}
```

Error

Status: Default, Error

Name	Type	Description
error	error	

Example error

```
{
  "error": {
    "arguments": [
      {
        "code": "string",
        "message": "string"
      }
    ],
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

export_policy

Export Policy

Name	Type	Description
_links	_links	
id	integer	
name	string	

group

The user set as owner of the qtree.

Name	Type	Description
id	string	The numeric ID of the group that owns the qtree. Valid in POST or PATCH.
name	string	Alphanumeric group name of group that owns the qtree. Valid in POST or PATCH.

nas

Name	Type	Description
path	string	Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH.

qos_policy

Name	Type	Description
_links	_links	
max_throughput_iops	integer	Specifies the maximum throughput in IOPS, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.
max_throughput_mbps	integer	Specifies the maximum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.
min_throughput_iops	integer	Specifies the minimum throughput in IOPS, 0 means none. Setting "min_throughput" is supported on AFF platforms only, unless FabricPool tiering policies are set. This is mutually exclusive with name and UUID during POST and PATCH.
min_throughput_mbps	integer	Specifies the minimum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.
name	string	The QoS policy group name. This is mutually exclusive with UUID and other QoS attributes during POST and PATCH.
uuid	string	The QoS policy group UUID. This is mutually exclusive with name and other QoS attributes during POST and PATCH.

iops_raw

The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

throughput_raw

Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

statistics

These are raw IOPS and throughput performance numbers. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.

Name	Type	Description
iops_raw	iops_raw	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled with the next closest collection and tagged with "backfilled_data". "inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "negative_delta" is returned when an expected monotonically increasing value has decreased in value. "inconsistent_old_data" is returned when one or more nodes does not have the latest data.
throughput_raw	throughput_raw	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

svm

Required in POST

Name	Type	Description
_links	_links	
name	string	The name of the SVM.

Name	Type	Description
uuid	string	The unique identifier of the SVM.

user

The user set as owner of the qtree.

Name	Type	Description
id	string	The numeric ID of the user who owns the qtree. Valid in POST or PATCH.
name	string	Alphanumeric username of user who owns the qtree. Valid in POST or PATCH.

volume

Required in POST

Name	Type	Description
_links	_links	
name	string	The name of the volume.
uuid	string	Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move. <ul style="list-style-type: none"> example: 028baa66-41bd-11e9-81d5-00a0986138f7 Introduced in: 9.6

qtree

A qtree is a directory at the top level of a volume to which a custom export policy (for fine-grained access control) and a quota rule can be applied, if required.

Name	Type	Description
_links	_links	
export_policy	export_policy	Export Policy

Name	Type	Description
filesystem_path	string	Path of the qtree directory. This path is relative to the volume root directory.
group	group	The user set as owner of the qtree.
id	integer	The identifier for the qtree, unique within the qtree's volume.
name	string	The name of the qtree. Required in POST; optional in PATCH.
nas	nas	
path	string	Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. This field is to be deprecated and replaced with nas.path.
qos_policy	qos_policy	
security_style	string	Security style. Valid in POST or PATCH.
statistics	statistics	These are raw IOPS and throughput performance numbers. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.
svm	svm	Required in POST
unix_permissions	integer	The UNIX permissions for the qtree. Valid in POST or PATCH.
user	user	The user set as owner of the qtree.
volume	volume	Required in POST

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Create a qtree in a FlexVol or FlexGroup volume

POST /storage/qtrees

Introduced In: 9.6

Creates a qtree in a FlexVol volume or a FlexGroup volume.

After a qtree is created, the new qtree is assigned an identifier. This identifier is obtained using a qtree GET request. This identifier is used in the API path for the qtree PATCH and DELETE operations.

Required properties

- `svm.uuid` or `svm.name` - Existing SVM in which to create the qtree.
- `volume.uuid` or `volume.name` - Existing volume in which to create the qtree.
- `name` - Name for the qtree.

Recommended optional properties

If not specified in POST, the values are inherited from the volume.

- `security_style` - Security style for the qtree.
- `unix_permissions` - UNIX permissions for the qtree.
- `export_policy.name` or `export_policy.id` - Export policy of the SVM for the qtree.

Related ONTAP commands

- `qtree create`

Parameters

Name	Type	In	Required	Description
return_timeout	integer	query	False	<p>The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.</p> <ul style="list-style-type: none">• Default value: 1• Max value: 120• Min value: 0
return_records	boolean	query	False	<p>The default is false. If set to true, the records are returned.</p> <ul style="list-style-type: none">• Default value:

Request Body

Name	Type	Description
export_policy	export_policy	Export Policy
filesystem_path	string	Path of the qtree directory. This path is relative to the volume root directory.
group	group	The user set as owner of the qtree.
id	integer	The identifier for the qtree, unique within the qtree's volume.
name	string	The name of the qtree. Required in POST; optional in PATCH.
nas	nas	
path	string	Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. This field is to be deprecated and replaced with nas.path.
qos_policy	qos_policy	
security_style	string	Security style. Valid in POST or PATCH.
svm	svm	Required in POST
unix_permissions	integer	The UNIX permissions for the qtree. Valid in POST or PATCH.
user	user	The user set as owner of the qtree.
volume	volume	Required in POST

Example request

```
{  
  "export_policy": {  
    "id": 100,  
    "name": "default"  
  },  
  "filesystem_path": "/dir1/qtree1",  
  "group": {  
    "id": "20001",  
    "name": "unix_group1"  
  },  
  "id": 1,  
  "name": "string",  
  "nas": {  
    "path": "/volume3/qtree1"  
  },  
  "path": "/volume3/qtree1",  
  "qos_policy": {  
    "max_throughput_iops": 10000,  
    "max_throughput_mbps": 500,  
    "min_throughput_iops": 2000,  
    "min_throughput_mbps": 500,  
    "name": "performance",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "security_style": "string",  
  "svm": {  
    "name": "svm1",  
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
  },  
  "unix_permissions": 755,  
  "user": {  
    "id": "10001",  
    "name": "unix_user1"  
  },  
  "volume": {  
    "name": "volume1",  
    "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"  
  }  
}
```

Response

Status: 202, Accepted

Name	Type	Description
job	job_link	

Example response

```
{  
  "job": {  
    "uuid": "string"  
  }  
}
```

Headers

Name	Description	Type
Location	Useful for tracking the resource location	string

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
917927	The specified volume was not found.
918232	Either volume.name or volume.uuid must be provided.
918236	The specified volume.uuid and volume.name refer to different volumes.
1703954	Export Policy name specified is invalid.
2621462	The specified SVM does not exist.
2621706	The specified svm.uuid and svm.name do not refer to the same SVM.
2621707	No SVM was specified. Either svm.name or svm.uuid must be provided.

Error Code	Description
5242886	Failed to create qtree.
5242951	Export Policy supplied does not belong to the specified Export Policy ID.
5242952	Export Policy ID specified is invalid.
5242953	Qtree name must be provided.
5242967	UNIX user or group ID must be 32-bit unsigned integer.

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

export_policy

Export Policy

Name	Type	Description
id	integer	
name	string	

group

The user set as owner of the qtree.

Name	Type	Description
id	string	The numeric ID of the group that owns the qtree. Valid in POST or PATCH.
name	string	Alphanumeric group name of group that owns the qtree. Valid in POST or PATCH.

nas

Name	Type	Description
path	string	Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH.

qos_policy

Name	Type	Description
max_throughput_iops	integer	Specifies the maximum throughput in IOPS, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.
max_throughput_mbps	integer	Specifies the maximum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.
min_throughput_iops	integer	Specifies the minimum throughput in IOPS, 0 means none. Setting "min_throughput" is supported on AFF platforms only, unless FabricPool tiering policies are set. This is mutually exclusive with name and UUID during POST and PATCH.
min_throughput_mbps	integer	Specifies the minimum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.
name	string	The QoS policy group name. This is mutually exclusive with UUID and other QoS attributes during POST and PATCH.
uuid	string	The QoS policy group UUID. This is mutually exclusive with name and other QoS attributes during POST and PATCH.

iops_raw

The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

throughput_raw

Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

statistics

These are raw IOPS and throughput performance numbers. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.

Name	Type	Description
iops_raw	iops_raw	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
status	string	<p>Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled with the next closest collection and tagged with "backfilled_data".</p> <p>"inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "negative_delta" is returned when an expected monotonically increasing value has decreased in value.</p> <p>"inconsistent_old_data" is returned when one or more nodes does not have the latest data.</p>
throughput_raw	throughput_raw	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

svm

Required in POST

Name	Type	Description
name	string	The name of the SVM.
uuid	string	The unique identifier of the SVM.

user

The user set as owner of the qtree.

Name	Type	Description
id	string	The numeric ID of the user who owns the qtree. Valid in POST or PATCH.
name	string	Alphanumeric username of user who owns the qtree. Valid in POST or PATCH.

volume

Required in POST

Name	Type	Description
name	string	The name of the volume.
uuid	string	Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move. <ul style="list-style-type: none"> example: 028baa66-41bd-11e9-81d5-00a0986138f7 Introduced in: 9.6

qtree

A qtree is a directory at the top level of a volume to which a custom export policy (for fine-grained access control) and a quota rule can be applied, if required.

Name	Type	Description
export_policy	export_policy	Export Policy
filesystem_path	string	Path of the qtree directory. This path is relative to the volume root directory.
group	group	The user set as owner of the qtree.
id	integer	The identifier for the qtree, unique within the qtree's volume.
name	string	The name of the qtree. Required in POST; optional in PATCH.

Name	Type	Description
nas	nas	
path	string	Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. This field is to be deprecated and replaced with nas.path.
qos_policy	qos_policy	
security_style	string	Security style. Valid in POST or PATCH.
svm	svm	Required in POST
unix_permissions	integer	The UNIX permissions for the qtree. Valid in POST or PATCH.
user	user	The user set as owner of the qtree.
volume	volume	Required in POST

job_link

Name	Type	Description
uuid	string	The UUID of the asynchronous job that is triggered by a POST, PATCH, or DELETE operation.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

Name	Type	Description
arguments	array[error_arguments]	Message arguments

Name	Type	Description
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Delete a qtree

DELETE /storage/qtrees/{volume.uuid}/{id}

Introduced In: 9.6

Deletes a qtree.

Related ONTAP commands

- `qtree delete`

Parameters

Name	Type	In	Required	Description
volume.uuid	string	path	True	Volume UUID
id	string	path	True	Qtree ID

Name	Type	In	Required	Description
return_timeout	integer	query	False	<p>The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.</p> <ul style="list-style-type: none"> • Default value: 1 • Max value: 120 • Min value: 0

Response

Status: 202, Accepted

Name	Type	Description
job	job_link	

Example response

```
{  
  "job": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "uuid": "string"  
  }  
}
```

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
918235	A volume with UUID was not found.
5242955	The UUID of the volume is required.
5242957	Failed to delete qtree with ID in the volume and SVM.

Name	Type	Description
error	error	

Example error

```
{  
  "error": {  
    "arguments": [  
      {  
        "code": "string",  
        "message": "string"  
      }  
    ],  
    "code": "4",  
    "message": "entry doesn't exist",  
    "target": "uuid"  
  }  
}
```

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

job_link

Name	Type	Description
_links	_links	
uuid	string	The UUID of the asynchronous job that is triggered by a POST, PATCH, or DELETE operation.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Retrieve qtree properties

GET /storage/qtrees/{volume.uuid}/{id}

Introduced In: 9.6

Retrieves properties for a specific qtree identified by the `volume.uuid` and the `id` in the API path.

Expensive properties

There is an added computational cost to retrieving values for these properties. They are not included by default in GET results and must be explicitly requested using the `fields` query parameter. See [Requesting specific fields](#) to learn more.

- `statistics.*`

Related ONTAP commands

- `qtree show`

Parameters

Name	Type	In	Required	Description
<code>volume.uuid</code>	string	path	True	Volume UUID
<code>id</code>	string	path	True	Qtree ID
<code>fields</code>	array[string]	query	False	Specify the fields to return.

Response

Status: 200, Ok

Name	Type	Description
<code>_links</code>	_links	
<code>export_policy</code>	export_policy	Export Policy
<code>filesystem_path</code>	string	Path of the qtree directory. This path is relative to the volume root directory.
<code>group</code>	group	The user set as owner of the qtree.
<code>id</code>	integer	The identifier for the qtree, unique within the qtree's volume.

Name	Type	Description
name	string	The name of the qtree. Required in POST; optional in PATCH.
nas	nas	
path	string	Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. This field is to be deprecated and replaced with nas.path.
qos_policy	qos_policy	
security_style	string	Security style. Valid in POST or PATCH.
statistics	statistics	These are raw IOPS and throughput performance numbers. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.
svm	svm	Required in POST
unix_permissions	integer	The UNIX permissions for the qtree. Valid in POST or PATCH.
user	user	The user set as owner of the qtree.
volume	volume	Required in POST

Example response

```
{  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "export_policy": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "id": 100,  
    "name": "default"  
  },  
  "filesystem_path": "/dir1/qtree1",  
  "group": {  
    "id": "20001",  
    "name": "unix_group1"  
  },  
  "id": 1,  
  "name": "string",  
  "nas": {  
    "path": "/volume3/qtree1"  
  },  
  "path": "/volume3/qtree1",  
  "qos_policy": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "max_throughput_iops": 10000,  
    "max_throughput_mbps": 500,  
    "min_throughput_iops": 2000,  
    "min_throughput_mbps": 500,  
    "name": "performance",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "security_style": "string",  
  "statistics": {  
    "iops_raw": {  
      "read": 200,  
      "total": 1000,  
      "write": 1000  
    }  
  }  
}
```

```

    "write": 100
  },
  "status": "ok",
  "throughput_raw": {
    "read": 200,
    "total": 1000,
    "write": 100
  },
  "timestamp": "2017-01-25T11:20:13Z"
},
"svm": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "svm1",
  "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
},
"unix_permissions": 755,
"user": {
  "id": "10001",
  "name": "unix_user1"
},
"volume": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "volume1",
  "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
}
}
}

```

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
918235	A volume with UUID was not found.

Error Code	Description
5242956	Failed to obtain a qtree with ID.

Name	Type	Description
error	error	

Example error

```
{
  "error": {
    "arguments": [
      {
        "code": "string",
        "message": "string"
      }
    ],
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

export_policy

Export Policy

Name	Type	Description
_links	_links	
id	integer	
name	string	

group

The user set as owner of the qtree.

Name	Type	Description
id	string	The numeric ID of the group that owns the qtree. Valid in POST or PATCH.
name	string	Alphanumeric group name of group that owns the qtree. Valid in POST or PATCH.

nas

Name	Type	Description
path	string	Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH.

qos_policy

Name	Type	Description
_links	_links	
max_throughput_iops	integer	Specifies the maximum throughput in IOPS, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.
max_throughput_mbps	integer	Specifies the maximum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.
min_throughput_iops	integer	Specifies the minimum throughput in IOPS, 0 means none. Setting "min_throughput" is supported on AFF platforms only, unless FabricPool tiering policies are set. This is mutually exclusive with name and UUID during POST and PATCH.
min_throughput_mbps	integer	Specifies the minimum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.
name	string	The QoS policy group name. This is mutually exclusive with UUID and other QoS attributes during POST and PATCH.
uuid	string	The QoS policy group UUID. This is mutually exclusive with name and other QoS attributes during POST and PATCH.

iops_raw

The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

throughput_raw

Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

statistics

These are raw IOPS and throughput performance numbers. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.

Name	Type	Description
iops_raw	iops_raw	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.
status	string	Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled with the next closest collection and tagged with "backfilled_data". "inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "negative_delta" is returned when an expected monotonically increasing value has decreased in value. "inconsistent_old_data" is returned when one or more nodes does not have the latest data.
throughput_raw	throughput_raw	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

svm

Required in POST

Name	Type	Description
_links	_links	
name	string	The name of the SVM.

Name	Type	Description
uuid	string	The unique identifier of the SVM.

user

The user set as owner of the qtree.

Name	Type	Description
id	string	The numeric ID of the user who owns the qtree. Valid in POST or PATCH.
name	string	Alphanumeric username of user who owns the qtree. Valid in POST or PATCH.

volume

Required in POST

Name	Type	Description
_links	_links	
name	string	The name of the volume.
uuid	string	Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move. <ul style="list-style-type: none"> example: 028baa66-41bd-11e9-81d5-00a0986138f7 Introduced in: 9.6

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message
target	string	The target parameter that caused the error.

Update properties for a qtree

PATCH /storage/qtrees/{volume.uuid}/{id}

Introduced In: 9.6

Updates properties for a specific qtree.

Related ONTAP commands

- `qtree modify`
- `qtree rename`

Parameters

Name	Type	In	Required	Description
volume.uuid	string	path	True	Volume UUID
id	string	path	True	Qtree ID

Name	Type	In	Required	Description
return_timeout	integer	query	False	<p>The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.</p> <ul style="list-style-type: none"> • Default value: 1 • Max value: 120 • Min value: 0

Request Body

Name	Type	Description
export_policy	export_policy	Export Policy
filesystem_path	string	Path of the qtree directory. This path is relative to the volume root directory.
group	group	The user set as owner of the qtree.
id	integer	The identifier for the qtree, unique within the qtree's volume.

Name	Type	Description
name	string	The name of the qtree. Required in POST; optional in PATCH.
nas	nas	
path	string	Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. This field is to be deprecated and replaced with nas.path.
qos_policy	qos_policy	
security_style	string	Security style. Valid in POST or PATCH.
unix_permissions	integer	The UNIX permissions for the qtree. Valid in POST or PATCH.
user	user	The user set as owner of the qtree.

Example request

```
{  
  "export_policy": {  
    "id": 100,  
    "name": "default"  
  },  
  "filesystem_path": "/dir1/qtree1",  
  "group": {  
    "id": "20001",  
    "name": "unix_group1"  
  },  
  "id": 1,  
  "name": "string",  
  "nas": {  
    "path": "/volume3/qtree1"  
  },  
  "path": "/volume3/qtree1",  
  "qos_policy": {  
    "max_throughput_iops": 10000,  
    "max_throughput_mbps": 500,  
    "min_throughput_iops": 2000,  
    "min_throughput_mbps": 500,  
    "name": "performance",  
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
  },  
  "security_style": "string",  
  "unix_permissions": 755,  
  "user": {  
    "id": "10001",  
    "name": "unix_user1"  
  }  
}
```

Response

Status: 202, Accepted

Name	Type	Description
job	job_link	

Example response

```
{  
  "job": {  
    "uuid": "string"  
  }  
}
```

Error

Status: Default

ONTAP Error Response Codes

Error Code	Description
918235	A volume with UUID was not found.
5242951	Export policy supplied does not belong to the specified export policy ID.
5242955	The UUID of the volume is required.
5242956	Failed to obtain a qtree with ID.
5242958	Failed to rename the qtree with ID in the volume and SVM.
5242959	Successfully renamed qtree but the modify operation failed.
5242967	UNIX user or group ID must be 32-bit unsigned integer.

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

export_policy

Export Policy

Name	Type	Description
id	integer	
name	string	

group

The user set as owner of the qtree.

Name	Type	Description
id	string	The numeric ID of the group that owns the qtree. Valid in POST or PATCH.
name	string	Alphanumeric group name of group that owns the qtree. Valid in POST or PATCH.

nas

Name	Type	Description
path	string	Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH.

qos_policy

Name	Type	Description
max_throughput_iops	integer	Specifies the maximum throughput in IOPS, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.
max_throughput_mbps	integer	Specifies the maximum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.
min_throughput_iops	integer	Specifies the minimum throughput in IOPS, 0 means none. Setting "min_throughput" is supported on AFF platforms only, unless FabricPool tiering policies are set. This is mutually exclusive with name and UUID during POST and PATCH.
min_throughput_mbps	integer	Specifies the minimum throughput in Megabytes per sec, 0 means none. This is mutually exclusive with name and UUID during POST and PATCH.
name	string	The QoS policy group name. This is mutually exclusive with UUID and other QoS attributes during POST and PATCH.
uuid	string	The QoS policy group UUID. This is mutually exclusive with name and other QoS attributes during POST and PATCH.

iops_raw

The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.

Name	Type	Description
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

throughput_raw

Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.

Name	Type	Description
other	integer	Performance metric for other I/O operations. Other I/O operations can be metadata operations, such as directory lookups and so on.
read	integer	Performance metric for read I/O operations.
total	integer	Performance metric aggregated over all types of I/O operations.
write	integer	Performance metric for write I/O operations.

statistics

These are raw IOPS and throughput performance numbers. These numbers are aggregated across all nodes in the cluster and increase with the uptime of the cluster.

Name	Type	Description
iops_raw	iops_raw	The number of I/O operations observed at the storage object. This should be used along with delta time to calculate the rate of I/O operations per unit of time.

Name	Type	Description
status	string	<p>Any errors associated with the sample. For example, if the aggregation of data over multiple nodes fails then any of the partial errors might be returned, "ok" on success, or "error" on any internal uncategorized failure. Whenever a sample collection is missed but done at a later time, it is back filled with the next closest collection and tagged with "backfilled_data".</p> <p>"inconsistent_delta_time" is encountered when the time between two collections is not the same for all nodes. Therefore, the aggregated value might be over or under inflated. "negative_delta" is returned when an expected monotonically increasing value has decreased in value.</p> <p>"inconsistent_old_data" is returned when one or more nodes does not have the latest data.</p>
throughput_raw	throughput_raw	Throughput bytes observed at the storage object. This should be used along with delta time to calculate the rate of throughput bytes per unit of time.
timestamp	string	The timestamp of the performance data.

svm

Required in POST

Name	Type	Description
name	string	The name of the SVM.
uuid	string	The unique identifier of the SVM.

user

The user set as owner of the qtree.

Name	Type	Description
id	string	The numeric ID of the user who owns the qtree. Valid in POST or PATCH.
name	string	Alphanumeric username of user who owns the qtree. Valid in POST or PATCH.

volume

Required in POST

Name	Type	Description
name	string	The name of the volume.
uuid	string	Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move. <ul style="list-style-type: none"> example: 028baa66-41bd-11e9-81d5-00a0986138f7 Introduced in: 9.6

qtree

A qtree is a directory at the top level of a volume to which a custom export policy (for fine-grained access control) and a quota rule can be applied, if required.

Name	Type	Description
export_policy	export_policy	Export Policy
filesystem_path	string	Path of the qtree directory. This path is relative to the volume root directory.
group	group	The user set as owner of the qtree.
id	integer	The identifier for the qtree, unique within the qtree's volume.
name	string	The name of the qtree. Required in POST; optional in PATCH.

Name	Type	Description
nas	nas	
path	string	Client visible path to the qtree. This field is not available if the volume does not have a junction-path configured. Not valid in POST or PATCH. This field is to be deprecated and replaced with nas.path.
qos_policy	qos_policy	
security_style	string	Security style. Valid in POST or PATCH.
unix_permissions	integer	The UNIX permissions for the qtree. Valid in POST or PATCH.
user	user	The user set as owner of the qtree.

job_link

Name	Type	Description
uuid	string	The UUID of the asynchronous job that is triggered by a POST, PATCH, or DELETE operation.

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

Name	Type	Description
arguments	array[error_arguments]	Message arguments
code	string	Error code
message	string	Error message

Name	Type	Description
target	string	The target parameter that caused the error.

Copyright information

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—with prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.