Volume API methods

Element Software

NetApp
August 18, 2025

Table of Contents

Volume API methods

Element software volume API methods enable you to manage volumes that reside on a storage node. You can create, modify, clone, and delete volumes with these methods. You can also use volume API methods to gather and display data measurements for a volume.

- CancelClone
- CancelGroupClone
- CloneMultipleVolumes
- CloneVolume
- CopyVolume
- CreateQoSPolicy
- CreateVolume
- CreateBackupTarget
- DeleteQoSPolicy
- DeleteVolume
- DeleteVolumes
- GetBackupTarget
- GetVolumeStats
- GetDefaultQoS
- GetQoSPolicy
- GetVolumeCount
- GetVolumeEfficiency
- ListActiveVolumes
- ListBackupTargets
- ListBulkVolumeJobs
- ListDeletedVolumes
- ListQoSPolicies
- ListSyncJobs
- ListVolumeQoSHistograms
- ListVolumes
- ListVolumeStats
- ListVolumesForAccount
- ListVolumeStatsByAccount
- ListVolumeStatsByVirtualVolume
- ListVolumeStatsByVolume
- ListVolumeStatsByVolumeAccessGroup

- ModifyBackupTarget
- ModifyQoSPolicy
- ModifyVolume
- ModifyVolumes
- PurgeDeletedVolume
- PurgeDeletedVolumes
- RemoveBackupTarget
- RestoreDeletedVolume
- SetDefaultQoS
- StartBulkVolumeRead
- StartBulkVolumeWrite
- UpdateBulkVolumeStatus

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

CancelClone

You can use the `CancelClone` method to stop an ongoing volume clone or volume copy process. When you cancel a group clone operation, the system completes and removes the operation's associated asyncHandle.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
cloneID	The cloneID for the ongoing clone process.	integer	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CancelClone",
   "params": {
      "cloneID" : 5,
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {}
}
```

New since version

9.6

CancelGroupClone

You can use the `CancelGroupClone` method to stop an ongoing clone process occurring on a group of volumes. When you cancel a group clone operation, the system completes and removes the operation's associated asyncHandle.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
groupCloneID	The cloneID for the ongoing clone process.	integer	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CancelGroupClone",
   "params": {
      "cloneID" : 5,
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {}
}
```

New since version

9.6

CloneMultipleVolumes

You can use the `CloneMultipleVolumes` method to create a clone of a group of specified volumes. You can assign a consistent set of characteristics to a group of multiple volumes when they are cloned together.

Before using the groupSnapshotID parameter to clone the volumes in a group snapshot, you must first create the group snapshot using the CreateGroupSnapshot API method or the web UI. Using groupSnapshotID is optional when cloning multiple volumes.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
access	New default access method for the new volumes if not overridden by information passed in the volume's array.	string	None	No

Name	Description	Type	Default value	Required
enableSnapMirrorReplication	Determines whether the volume can be used for replication with SnapMirror endpoints. Possible values: • true • false	boolean	false	No
groupSnapshotID	ID of the group snapshot to use as a basis for the clone.	integer	None	No
newAccountID	New account ID for the volumes if not overridden by information passed in the volumes array.	integer	None	No

Name	Description	Type	Default value	Required
volumes	Collection of members that you specify for the new volumes. Members: • volumeID: (Required) • access: (Optional) Can be one of readOnly, readWrite, locked, or replicationTarget. • attributes: (Optional) List of name-value pairs in JSON object format. • name: (Optional) New name for the clone. • newAccountID: (Optional) Account ID for the new volumes. • newSize: (Optional) Total size of the volume, in bytes. Size is rounded up to the nearest megabyte. If optional members are not specified, the values are inherited from the source volumes.	JSON object array	None	Yes (volumeID)

Return values

This method has the following return values:

Name	Description	Type

asyncHandle	A value returned from an asynchronous method call.	integer
groupCloneID	Unique ID of the new group clone.	integer
members	List of volumeIDs for the source and destination volume pairs.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CloneMultipleVolumes",
   "params": {
       "volumes": [
           {
              "volumeID": 5
              "name":"foxhill",
              "access":"readOnly"
              },
           {
              "volumeID": 18
              },
           {
             "volumeID": 20
              }
     ]
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "asyncHandle": 12,
    "groupCloneID": 4,
    "members": [
     {
      "srcVolumeID": 5,
      "volumeID": 29
     },
     {
      "srcVolumeID": 18,
      "volumeID": 30
     },
     {
      "srcVolumeID": 20,
      "volumeID": 31
      }
    ]
  }
}
```


New since version

9.6

CloneVolume

You can use the `CloneVolume` method to create a copy of a volume. This method is asynchronous and might take a variable amount of time to complete.

The cloning process begins immediately when you make the `CloneVolume` request and is representative of the state of the volume when the API method is issued. You can use the GetAsyncResult method to determine when the cloning process is complete and the new volume is available for connections. You can use ListSyncJobs to see the progress of creating the clone. The initial attributes and quality of service settings for the volume are inherited from the volume being cloned. You can change these settings with ModifyVolume.

Cloned volumes do not inherit volume access group membership from the source volume.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
access	Access allowed for the new volume. If a value is not specified, the access value does not change. Possible values: • `readOnly`: (Optional) Only read operations are allowed. • `readWrite`: (Optional) Reads and writes are allowed. • `locked`: (Optional) No reads or writes are allowed. If not specified, the access value of the volume being cloned is used. • `replicationTarget`: (Optional) Identify a volume as the target volume for a paired set of volumes. If the volume is not paired, the access status is locked.	string	None	No
attributes	List of name-value pairs in JSON object format.	JSON object	None	No

Name	Description	Type	Default value	Required
enable512e	Specifies whether the new volume should use 512-byte sector emulation. If unspecified, the setting of the volume being cloned is used.	boolean	Setting of original volume	No
enableSnapMirrorReplication	Determines whether the volume can be used for replication with SnapMirror endpoints. Possible values: • true • false	boolean	false	No
name	Name of the new cloned volume; must be 1 to 64 characters in length.	string	None	Yes
newAccountID	AccountID for the owner of the new volume. If unspecified, the accountID of the owner of the volume being cloned is used.	integer	AccountID of the owner of original volume	No
newSize	New size of the volume, in bytes. Might be greater or less than the size of the volume being cloned. If not specified, the volume size is not changed. Size is rounded up to the nearest 1MB in size.	integer	None	No

Name	Description	Type	Default value	Required
snapshotID	ID of the snapshot that is used as the source of the clone. If no ID is provided, the current active volume is used.	integer	None	No
volumeID	VolumeID for the volume to be cloned.	integer	None	Yes

Return values

This method has the following return values:

Name	Description	Type
asyncHandle	The handle value used to obtain the operation result.	integer
cloneID	The cloneID for the newly cloned volume.	integer
curve	The QoS curve values applied to the clone.	JSON object
volume	An object containing information about the newly cloned volume.	volume
volumeID	VolumeID for the newly cloned volume.	integer

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CloneVolume",
   "params": {
      "volumeID" : 5,
      "name"  : "mysqldata-snapshot1",
      "access" : "readOnly"
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:


```
{
  "id": 1,
  "result": {
      "asyncHandle": 42,
      "cloneID": 37,
      "volume": {
          "access": "readOnly",
          "accountID": 1,
          "attributes": {},
          "blockSize": 4096,
          "createTime": "2016-03-31T22:26:03Z",
          "deleteTime": "",
          "enable512e": true,
          "iqn": "iqn.2010-01.com.solidfire:jyay.mysqldata-snapshot1.680",
          "name": "mysqldata-snapshot1",
          "purgeTime": "",
          "qos": {
              "burstIOPS": 100,
              "burstTime": 60,
              "curve": {
                  "4096": 100,
                  "8192": 160,
                  "16384": 270,
                  "32768": 500,
                  "65536": 1000,
                  "131072": 1950,
                  "262144": 3900,
                  "524288": 7600,
                  "1048576": 15000
              },
              "maxIOPS": 100,
              "minIOPS": 50
          },
          "scsiEUIDeviceID": "6a796179000002a8f47acc0100000000",
          "scsiNAADeviceID": "6f47acc1000000006a796179000002a8",
          "sliceCount": 0,
          "status": "init",
          "totalSize": 1000341504,
          "virtualVolumeID": null,
          "volumeAccessGroups": [],
          "volumeID": 680,
          "volumePairs": []
```

```
        },
        "volumeID": 680
    }
}
```

New since version

9.6

Find more information

- GetAsyncResult
- ListSyncJobs
- ModifyVolume

CopyVolume

You can use the `CopyVolume` method to overwrite the data contents of an existing volume with the data contents of another volume (or snapshot). Attributes of the destination volume such as IQN, QoS settings, size, account, and volume access group membership are not changed. The destination volume must already exist and must be the same size as the source volume.

It is best if clients unmount the destination volume before the operation begins. If the destination volume is modified during the operation, the changes are lost. This operation can take a variable amount of time to complete. You can use the GetAsyncResult method to determine when the process has finished, and ListSyncJobs to see the progress of the copy.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
dstVolumeID	VolumeID of the volume to overwrite.	integer	None	Yes
volumeID	VolumeID of the volume to be read from.	integer	None	Yes
snapshotID	ID of the snapshot that is used as the source of the clone. If no ID is provided, the current active volume is used.	integer	None	No

Return values

This method has the following return values:

Name	Description	Type
asyncHandle	Handle value used to obtain the operation result.	integer
cloneID	CloneID for the newly cloned volume.	integer

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CopyVolume",
   "params": {
      "volumeID" : 3,
      "dstVolumeID" : 2
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "asyncHandle": 9,
    "cloneID": 5
  }
}
```

New since version

9.6

Find more information

- GetAsyncResult
- ListSyncJobs

CreateQoSPolicy

You can use the `CreateQoSPolicy` method to create a QoSPolicy object that you can later apply to a volume upon creation or modification. A QoS policy has a unique ID, a name, and QoS settings.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
name	The name of the QoS policy; for example, gold, platinum, or silver.	string	None	Yes
qos	The QoS settings that this policy represents.	QoS	None	Yes

Return value

This method has the following return value:

Name	Description	Type
qosPolicy	The newly created QoSPolicy object.	QoSPolicy

Request example

Requests for this method are similar to the following example:

```
{
  "id": 68,
  "method": "CreateQoSPolicy",
  "params": {
    "name": "bronze",
    "qos": {
      "minIOPS": 50,
      "maxIOPS": 15000,
      "burstIOPS": 15000
    }
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 68,
  "result": {
    "qosPolicy": {
      "name": "bronze",
      "qos": {
        "burstIOPS": 15000,
        "burstTime": 60,
        "curve": {
          "4096": 100,
          "8192": 160,
          "16384": 270,
          "32768": 500,
          "65536": 1000,
          "131072": 1950,
          "262144": 3900,
          "524288": 7600,
          "1048576": 15000
        },
        "maxIOPS": 15000,
        "minIOPS": 50
      },
      "qosPolicyID": 2,
      "volumeIDs": []
    }
  }
}
```

New since version

10.0

CreateVolume

You can use the `CreateVolume` method to create a new, empty volume on the cluster. As soon as the volume is created, the volume is available for connection via iSCSI.

Volumes created without specified QoS values use the default values. You can view default values for a volume by using the `GetDefaultQoS` method.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
`access`	The access mode for the volume. If this parameter is included, the only supported value is `snapMirrorTarget`.	string	None	No
`accountID`	The ID of the account that owns this volume.	integer	None	Yes
`associateWithQoSPolicy`	Associate the volume with the specified QoS policy. Possible values: • `true`: Associate the volume with the QoS policy specified in the QoSPolicyID parameter. • `false`: Do not associate the volume with the QoS policy specified in the QoSPolicyID parameter. When false, any existing policy association is removed, regardless of whether you specify a QoS policy in the QoSPolicy parameter.	boolean	true	No

Name	Description	Type	Default value	Required
`attributes`	List of name-value pairs in JSON object format. The total attribute size must be less than 1000B, or 1KB, including JSON formatting characters.	JSON object	None	No
`dontMoveforIopsRebalance`	Prevent a volume from moving when load balancing with actual IOPS. This setting is available beginning with Element 12.8 and is only effective when VolumeLoadBalanceOnActualIOPS is enabled. Possible values: • `true`: Volume does not load balance based on actual IOPS. • `false`: Volume does load balance based on actual IOPS. **Note**: Balancing for high availability (node failure) takes precedence over `VolumeLoadBalanceOnActualIOPS` and `dontMoveforIopsRebalance`.	boolean	false	No

Name	Description	Type	Default value	Required
`enable512e`	Enable 512-byte sector emulation. Possible values: • `true`: The volume provides 512-byte sector emulation. • `false`: 512e emulation is not enabled.	boolean	None	Yes
`enableSnapMirrorReplication`	Determines whether the volume can be used for replication with SnapMirror endpoints. Possible values: • `true` • `false`	boolean	false	No
`fifoSize`	Specifies the maximum number of First-In-First-Out (FIFO) snapshots supported by the volume. Note that FIFO and non-FIFO snapshots both use the same pool of available snapshot slots on a volume. Use this option to limit FIFO snapshot consumption of the available snapshot slots. If omitted, the value defaults to 24.	integer	24	No

Name	Description	Type	Default value	Required
`minFifoSize`	Specifies the minimum number of First-In-First-Out (FIFO) snapshot slots reserved by the volume. This guarantees that if you are using both FIFO snapshots and non-FIFO snapshots on a volume that the non-FIFO snapshots do not unintentionally consume too many FIFO slots. It also ensures that at least this many FIFO snapshots are always available. Since FIFO and non-FIFO snapshots share the same pool, the `minFifoSize` reduces the total number of possible non-FIFO snapshots by the same amount. If omitted, the value defaults to 0.	integer	0	No
`name`	Name of the volume access group (may be user-specified). Not required to be unique, but recommended. Must be 1 to 64 characters in length.	string	None	Yes
`qos`	The initial quality of service settings for this volume. Default values are used if none are specified. Possible values: • `minIOPS` • `maxIOPS` • `burstIOPS`	QoS object	None	No

Name	Description	Type	Default value	Required
`qosPolicyID`	The ID for the policy whose QoS settings should be applied to the specified volumes. This parameter is mutually exclusive with the `qos` parameter.	integer	None	No
`totalSize`	Total size of the volume, in bytes. Size is rounded up to the nearest megabyte.	integer	None	Yes

Return values

This method has the following return values:

Name	Description	Type
volume	Object containing information about the newly created volume.	volume
volumeID	The volumeID for the newly created volume.	integer
curve	The curve is a set of key-value pairs. The keys are the I/O sizes in bytes. The values represent the cost of performing an IOP at a specific I/O size. The curve is calculated relative to a 4096 byte operation set at 100 IOPS.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CreateVolume",
   "params": {
      "name": "testit",
      "accountID": 22,
      "dontMoveForIopsRebalance": true,
      "totalSize": 100000000000,
      "enable512e": false,
      "attributes": {},
      "qos": {
         "minIOPS": 500,
         "maxIOPS": 27000,
         "burstIOPS": 27000,
         "burstTime": 60
      }
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "curve": {
            "1048576": 15000,
            "131072": 1950,
            "16384": 270,
            "262144": 3900,
            "32768": 500,
            "4096": 100,
            "524288": 7600,
            "65536": 1000,
            "8192": 160
        },
        "volume": {
            "access": "readWrite",
            "accountID": 22,
            "attributes": {},
            "blockSize": 4096,
            "createTime": "2024-04-02T13:03:02Z",
            "currentProtectionScheme": "doubleHelix",
            "deleteTime": "",
```

```
            "dontMoveForIopsRebalance": true,
            "enable512e": false,
            "enableSnapMirrorReplication": false,
            "fifoSize": 24,
            "iqn": "iqn.2010-01.com.solidfire:mysqldata.677",
            "lastAccessTime": null,
            "lastAccessTimeIO": null,
            "minFifoSize": 0,
            "name": "testit",
            "previousProtectionScheme": null,
            "purgeTime": "",
            "qos": {
                "burstIOPS": 27000,
                "burstTime": 60,
                "curve": {
                    "1048576": 15000,
                    "131072": 1950,
                    "16384": 270,
                    "262144": 3900,
                    "32768": 500,
                    "4096": 100,
                    "524288": 7600,
                    "65536": 1000,
                    "8192": 160
                },
                "maxIOPS": 27000,
                "minIOPS": 500
            },
            "qosPolicyID": null,
            "scsiEUIDeviceID": "3365657500000140f47acc0100000000",
            "scsiNAADeviceID": "6f47acc100000000336565750000140",
            "sliceCount": 0,
            "status": "active",
            "totalSize": 1000000716800,
            "virtualVolumeID": null,
            "volumeAccessGroups": [],
            "volumeConsistencyGroupUUID": "8ed68e57-13ee-47df-8381-
29b125142718",
            "volumeID": 320,
            "volumePairs": [],
            "volumeUUID": "e0e2c938-4ecd-4de9-a1be-f6b17c93ce5d"
        },
        "volumeID": 320
    }
}
```

New since version

9.6

Find more information

GetDefaultQoS

CreateBackupTarget

You can use `CreateBackupTarget` to create and store backup target information so that you do not need to re-enter it each time a backup is created.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
name	Name for the backup target.	string	None	Yes
attributes	List of name-value pairs in JSON object format.	JSON object	None	Yes (but can be empty)

Return value

This method has the following return value:

Name	Description	Type
backupTargetID	Unique identifier assigned to the new backup target.	integer

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CreateBackupTarget",
   "params": {
      "name": "mytargetbackup"
   },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
      "backupTargetID": 1
   }
}
```

New since version

9.6

DeleteQoSPolicy

You can use the `DeleteQoSPolicy` method to delete a QoS policy from the system. The QoS settings for all volumes created or modified with this policy are unaffected.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
qosPolicyID	The ID of the QoS policy to be deleted.	integer	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "id": 663,
  "method": "DeleteQoSPolicy",
  "params": {
    "qosPolicyID": 4
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 663,
  "result": {}
}
```

New since version

9.6

DeleteVolume

You can use the `DeleteVolume` method to mark an active volume for deletion. When marked, the volume is purged (permanently deleted) after the cleanup interval elapses.

After making a request to delete a volume, any active iSCSI connections to the volume are immediately terminated and no further connections are allowed while the volume is in this state. A marked volume is not returned in target discovery requests.

Any snapshots of a volume that has been marked for deletion are not affected. Snapshots are kept until the volume is purged from the system. If a volume is marked for deletion and has a bulk volume read or bulk volume write operation in progress, the bulk volume read or write operation is stopped.

If the volume you delete is paired with a volume, replication between the paired volumes is suspended and no data is transferred to it or from it while in a deleted state. The remote volume the deleted volume was paired with enters into a PausedMisconfigured state and data is no longer sent to it or from the deleted volume. Until the deleted volume is purged, it can be restored and data transfers resume. If the deleted volume gets purged from the system, the volume it was paired with enters into a StoppedMisconfigured state and the volume pairing status is removed. The purged volume becomes permanently unavailable.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
volumeID	The ID of the volume to delete.	integer	None	Yes

Return values

This method has the following return values:

Name	Description	Type

volume	Object containing information about the deleted volume.	volume
volumeID	The volumeID of the deleted volume.	integer
curve	The curve is a set of key-value pairs. The keys are the I/O sizes in bytes. The values represent the cost of performing an IOP at a specific I/O size. The curve is calculated relative to a 4096 byte operation set at 100 IOPS.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "DeleteVolume",
   "params": {
      "volumeID"  : 5
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volume": {
      "access": "readWrite",
      "accountID": 1,
      "attributes": {
        "name1": "value1",
        "name2": "value2",
        "name3": "value3"
      },
      "blockSize": 4096,
      "createTime": "2016-03-28T16:16:13Z",
      "deleteTime": "2016-03-31T22:59:42Z",
      "enable512e": true,
```

```
      "iqn": "iqn.2010-01.com.solidfire:jyay.1459181777648.5",
      "name": "1459181777648",
      "purgeTime": "2016-04-01T06:59:42Z",
      "qos": {
        "burstIOPS": 150,
        "burstTime": 60,
        "curve": {
          "4096": 100,
          "8192": 160,
          "16384": 270,
          "32768": 500,
          "65536": 1000,
          "131072": 1950,
          "262144": 3900,
          "524288": 7600,
          "1048576": 15000
        },
        "maxIOPS": 100,
        "minIOPS": 60
      },
      "scsiEUIDeviceID": "6a79617900000005f47acc0100000000",
      "scsiNAADeviceID": "6f47acc1000000006a79617900000005",
      "sliceCount": 1,
      "status": "deleted",
      "totalSize": 1000341504,
      "virtualVolumeID": null,
      "volumeAccessGroups": [
        1
      ],
      "volumeID": 5,
      "volumePairs": []
    }
  }
}
```


New since version

9.6

DeleteVolumes

You can use the `DeleteVolumes` method to mark multiple (up to 500) active volumes for deletion. When marked, the volume is purged (permanently deleted) after the cleanup interval elapses.

After making a request to delete volumes, any active iSCSI connections to the volumes are immediately

terminated and no further connections are allowed while the volumes are in this state. A marked volume is not returned in target discovery requests.

Any snapshots of a volume that has been marked for deletion are not affected. Snapshots are kept until the volume is purged from the system. If a volume is marked for deletion and has a bulk volume read or bulk volume write operation in progress, the bulk volume read or write operation is stopped.

If the volumes you delete are paired with a volume, replication between the paired volumes is suspended and no data is transferred to them or from them while in a deleted state. The remote volumes the deleted volumes were paired with enter into a PausedMisconfigured state and data is no longer sent to them or from the deleted volumes. Until the deleted volumes are purged, they can be restored and data transfers resume. If the deleted volumes are purged from the system, the volumes they were paired with enter into a StoppedMisconfigured state and the volume pairing status is removed. The purged volumes become permanently unavailable.

Parameters

This method has the following input parameters.

At least one of the following parameters are required, and you must use only one of the parameters (they are all mutually exclusive with one another).

Name	Description	Type	Default value	Required
volumeIDs	The list of IDs of the volumes to delete from the system.	integer array	None	See Note.
volumeAccessGroupIDs	A list of volume access group IDs. All of the volumes from all of the volume access groups you specify in this list are deleted from the system.	integer array	None	See Note.
accountIDs	A list of account IDs. All volumes from these accounts are deleted from the system.	integer array	None	See Note.

Return values

This method has the following return values:

Name	Description	Type
volumes	Information about the newly deleted volume.	volume

curve	The curve is a set of key-value pairs. The keys are the I/O sizes in bytes. The values represent the cost of performing an IOP at a specific I/O size. The curve is calculated relative to a 4096 byte operation set at 100 IOPS.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "DeleteVolumes",
   "params": {
      "accountIDs" : [1, 2, 3]
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result": {
    "volumes" : [ {
      "access": "readWrite",
      "accountID": 1,
      "attributes": {},
      "blockSize": 4096,
      "createTime": "2015-03-06T18:50:56Z",
      "deleteTime": "",
      "enable512e": False,
      "iqn": "iqn.2010-01.com.solidfire:pzsr.vclient-030-v00001.1",
      "name": "vclient-030-v00001",
      "qos": {
        "burstIOPS": 15000,
        "burstTime": 60,
        "curve": {},
        "maxIOPS": 15000,
        "minIOPS": 100
      },
      "purgeTime": "",
      "sliceCount": 1,
      "scsiEUIDeviceID": "707a737200000001f47acc0100000000",
      "scsiNAADeviceID": "6f47acc100000000707a737200000001",
      "status": "active",
      "totalSize": 10000003072,
      "virtualVolumeID": 5,
      "volumeAccessGroups": [],
      "volumePairs": [],
      "volumeID": 1
    } ]
  }
}
```


New since version

9.6

GetBackupTarget

You can use the `GetBackupTarget` method to return information about a specific backup target that you have created.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
attributes	List of name-value pairs in JSON object format.	JSON object	None	No
backupTargetID	Unique identifier assigned to the backup target.	integer	None	Yes
name	Name of the backup target.	string	None	No

Return value

This method has the following return value:

Name	Description	Type
backupTarget	List of name-value pairs in JSON object format.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
  "id": 1,
  "method": "GetBackupTarget",
  "params": {
      "backupTargetID": 1
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
     "backupTarget": {
     "attributes" : {
        "size" : 100
     },
     "backupTargetID" : 1,
     "name" : "mytargetbackup"
    }
  }
}
```

New since version

9.6

GetVolumeStats

You can use the `GetVolumeStats` method to get high-level activity measurements for a single volume. Values are cumulative from the creation of the volume.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
volumeID	Specifies the volume for which statistics are gathered.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
volumeStats	Volume activity information.	volumeStats

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetVolumeStats",
   "params": {
      "volumeID": 32
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volumeStats": [
        {
            "accountID": 1,
            "actualIOPS": 4,
            "asyncDelay": null,
            "averageIOPSize": 5970,
            "burstIOPSCredit": 0,
            "clientQueueDepth": 0,
            "desiredMetadataHosts": null,
            "latencyUSec": 474,
            "metadataHosts": {
                "deadSecondaries": [],
                "liveSecondaries": [
                    13
                ],
                "primary": 25
            },
            "nonZeroBlocks": 34931222,
            "normalizedIOPS": 4,
            "readBytes": 1282491003392,
            "readBytesLastSample": 0,
            "readLatencyUSec": 0,
            "readLatencyUSecTotal": 4581669750,
            "readOps": 15592933,
            "readOpsLastSample": 0,
            "samplePeriodMSec": 500,
            "sliceIopsStats": {
                "largeStatistics": {
                    "averageReadIops": 17,
                    "averageTotalIops": 43,
```

```
                    "averageWriteIops": 26,
                    "nSamples": 24,
                    "peakReadIops": 19,
                    "peakTotalIops": 47,
                    "peakWriteIops": 30,
                    "sliceID": 1
                },
                "smallStatistics": {
                    "averageReadIops": 17,
                    "averageTotalIops": 42,
                    "averageWriteIops": 25,
                    "nSamples": 120,
                    "peakReadIops": 173,
                    "peakTotalIops": 249,
                    "peakWriteIops": 77,
                    "sliceID": 1
            }
        },
        "throttle": 0,
        "timestamp": "2025-02-03T21:18:38.880100Z",
        "unalignedReads": 167319,
        "unalignedWrites": 90836,
        "volumeAccessGroups": [
            1
        ],
        "volumeID": 1,
        "volumeSize": 2147483648000,
        "volumeUtilization": 0.0002666666666666667,
        "writeBytes": 1385173585408,
        "writeBytesLastSample": 12288,
        "writeLatencyUSec": 474,
        "writeLatencyUSecTotal": 11233350905,
        "writeOps": 157060458,
        "writeOpsLastSample": 2,
        "zeroBlocks": 489356778
    }
  }
}
```


New since version

9.6

GetDefaultQoS

You can use the `GetDefaultQoS` method to get the default quality of service (QoS)

values for a newly created volume.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
QoS	The default QoS values.	QoS

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetDefaultQoS",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result" : {
      "burstIOPS" : 15000,
      "burstTime" : 60,
      "curve" : {
         "1048576" : 15000,
         "131072" : 1900,
         "16384" : 270,
         "262144" : 3000,
         "32768" : 500,
         "4096" : 100,
         "524288" : 7500,
         "65536" : 1000,
         "8192" : 160
      },
      "maxIOPS" : 15000,
      "minIOPS" : 100
   }
}
```


New since version

9.6

GetQoSPolicy

You can use the `GetQoSPolicy` method to get details about a specific QoS policy from the system.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
qosPolicyID	The ID of the policy to be retrieved.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
qosPolicy	Details of the requested QoS policy.	QoSPolicy

Request example

Requests for this method are similar to the following example:

```
{
    "method": "GetQoSPolicy",
    "params": {
        "qosPolicyID": 2
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "qosPolicy": {
      "name": "bronze",
      "qos": {
        "burstIOPS": 15002,
        "burstTime": 60,
        "curve": {
          "4096": 100,
          "8192": 160,
          "16384": 270,
          "32768": 500,
          "65536": 1000,
          "131072": 1950,
          "262144": 3900,
          "524288": 7600,
          "1048576": 15000
        },
        "maxIOPS": 15002,
        "minIOPS": 51
      },
      "qosPolicyID": 2,
      "volumeIDs": [
          2
      ]
    }
  }
}
```

New since version

10.0

GetVolumeCount

You can use the `GetVolumeCount` method to get the number of volumes currently in the system.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
count	The number of volumes currently in the system.	integer

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetVolumeCount",
    "params": {
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "count": 7
  }
}
```

New since version

9.6

GetVolumeEfficiency

You can use the `GetVolumeEfficiency` method to get information about a volume. Only the volume you give as a parameter in this API method is used to compute the capacity.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
volumeID	Specifies the volume for which capacity is computed.	integer	None	Yes

Return values

This method has the following return values:

Name	Description	Type
compression	The amount of space being saved by compressing data on a single volume. Stated as a ratio, where 1 means data has been stored without being compressed.	float
deduplication	The amount of space being saved on a single volume by not duplicating data. Stated as a ratio.	float
missingVolumes	The volumes that could not be queried for efficiency data. Missing volumes can be caused by Garbage Collection (GC) being less than an hour old, temporary network loss or restarted services since the GC cycle.	integer array
thinProvisioning	The ratio of space used to the amount of space allocated for storing data. Stated as a ratio.	float
timestamp	The last time efficiency data was collected after GC.	ISO 8601 data string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetVolumeEfficiency",
   "params": {
      "volumeID": 606
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
       "compression": 2.001591240821456,
       "deduplication": 1,
       "missingVolumes": [],
       "thinProvisioning": 1.009861932938856,
       "timestamp": "2014-03-10T16:06:33Z"
   }
}
```

New since version

9.6

ListActiveVolumes

You can use the `ListActiveVolumes` method to get the list of active volumes currently in the system. The list of volumes is sorted in VolumeID order and can be returned in multiple parts (pages).

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
includeVirtualVolumes	Virtual volumes are included in the response, by default. To exclude virtual volumes, set to false.	boolean	true	No

Name	Description	Type	Default value	Required
startVolumeID	Starting VolumeID to return. If no volume exists with this VolumeID, the next volume by VolumeID order is used as the start of the list. To page through the list, pass the VolumeID of the last volume in the previous response + 1.	integer	0	No
limit	Maximum number of volume info objects to return. 0 (zero) returns all volumes (unlimited).	integer	(unlimited)	No

Return value

This method has the following return value:

Name	Description	Type
volumes	List of active volumes.	volume array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListActiveVolumes",
   "params": {
      "startVolumeID"  : 0,
      "limit" : 1000
   },
   "id" : 1
}
```

Response example

Due to the length of this response example, it is documented in a supplementary topic.

New since version

9.6

ListBackupTargets

You can use the `ListBackupTargets` method to get information about all backup targets that have been created.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
backupTargets	Objects returned for each backup target. Included objects: • attributes: List of name-value pairs in JSON object format. (JSON object) • backupTargetID: Unique identifier assigned to the backup target. (integer) • name: Name of the backup target. (string)	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListBackupTargets",
   "params": {},
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
     "backupTargets": [
       {
        "attributes" : {},
        "backupTargetID" : 1,
        "name" : "mytargetbackup"
       }
     ]
   }
}
```

New since version

9.6

ListBulkVolumeJobs

You can use the `ListBulkVolumeJobs` method to get information about each bulk volume read or write operation that is occurring in the system.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
bulkVolumeJobs	An array of information for each bulk volume job.	bulkVolumeJob array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListBulkVolumeJobs",
   "params": {
      },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:


```
{
  "id": 1,
  "result": {
    "bulkVolumeJobs": [
        {
          "attributes": {
            "blocksPerTransfer": 1024,
            "firstPendingLba": 216064,
            "nLbas": 2441472,
            "nextLba": 226304,
            "pendingLbas": "[220160, 223232, 221184, 224256, 217088,
225280, 222208, 218112, 219136, 216064]",
            "percentComplete": 8,
            "startLba": 0
          },
          "bulkVolumeID": 2,
          "createTime": "2015-05-07T14:52:17Z",
          "elapsedTime": 44,
          "format": "native",
          "key": "eaffb0526d4fb47107061f09bfc9a806",
          "percentComplete": 8,
          "remainingTime": 506,
          "script": "bv_internal.py",
          "snapshotID": 509,
          "srcVolumeID": 3,
          "status": "running",
          "type": "read"
        }
      ]
   }
}
```


New since version

9.6

ListDeletedVolumes

You can use the `ListDeletedVolumes` method to retrieve the list of volumes that have been marked for deletion and purged from the system.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
includeVirtualVolumes	Virtual volumes are included in the response, by default. To exclude virtual volumes, set to false.	boolean	true	No

Return value

This method has the following return value:

Name	Description	Type
volumes	List of deleted volumes.	volume array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListDeletedVolumes",
   "params": {},
   "id" : 1
}
```

Response example

Responses for this method are similar to the following example:

```
{
    "id": 1,
    "result": {
        "volumes": [
            {
                "access": "readWrite",
                "accountID": 2,
                "attributes": {},
                "blockSize": 4096,
                "createTime": "2018-06-24T03:13:13Z",
                "deleteTime": "2018-07-22T16:12:39Z",
                "enable512e": true,
                "iqn": "iqn.2010-01.com.solidfire:0oto.deletethis.23",
                "name": "deleteThis",
                "purgeTime": "2016-07-23T00:12:39Z",
                "qos": {
                    "burstIOPS": 15000,
                    "burstTime": 60,
                    "curve": {
                        "4096": 100,
                        "8192": 160,
                        "16384": 270,
                        "32768": 500,
                        "65536": 1000,
                        "131072": 1950,
                        "262144": 3900,
                        "524288": 7600,
                        "1048576": 15000
                    },
                    "maxIOPS": 15000,
                    "minIOPS": 50
                },
                "scsiEUIDeviceID": "306f746f00000017f47acc0100000000",
                "scsiNAADeviceID": "6f47acc100000000306f746f00000017",
                "sliceCount": 1,
                "status": "deleted",
                "totalSize": 1396703232,
                "virtualVolumeID": null,
                "volumeAccessGroups": [],
                "volumeID": 23,
                "volumePairs": []
            }
        ]
    }
}
```

New since version

9.6

ListQoSPolicies

You can use the `ListQoSPolicies` method to list the settings of all QoS policies on the system.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
qosPolicies	A list of details about each QoS policy.	QoSPolicy array

Request example

Requests for this method are similar to the following example:

```
{
  "id": 231,
  "method": "ListQoSPolicies",
  "params": {}
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 231,
  "result": {
    "qosPolicies": [
      {
        "name": "silver",
        "qos": {
          "burstIOPS": 15000,
          "burstTime": 60,
          "curve": {
```

```
            "4096": 100,
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
          },
          "maxIOPS": 14000,
          "minIOPS": 50
        },
        "qosPolicyID": 1,
        "volumeIDs": [
          1
        ]
      },
      {
        "name": "bronze",
        "qos": {
          "burstIOPS": 15000,
          "burstTime": 60,
          "curve": {
            "4096": 100,
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
          },
          "maxIOPS": 15000,
          "minIOPS": 50
        },
        "qosPolicyID": 2,
        "volumeIDs": [
          2
        ]
      }
    ]
  }
}
```

New since version

10.0

ListSyncJobs

You can use the `ListSyncJobs` method to get information about synchronization jobs that are running on an Element storage cluster. This method returns information about slice, clone, block, and remote synchronization jobs.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
syncJobs	List of objects describing synchronization processes that are currently running in the system.	syncJob array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListSyncJobs",
   "params": { },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id":1,
    "result":{
      "syncJobs":[
        {
           "bytesPerSecond":275314.8834458956,
           "currentBytes":178257920,
           "dstServiceID":36,
```

```
            "elapsedTime":289.4568382049871,
            "percentComplete":8.900523560209423,
            "remainingTime":2962.675921065957,
            "sliceID":5,
            "srcServiceID":16,
            "stage":"whole",
            "totalBytes":2002780160,
            "type":"slice"
        },
        {
            "bytesPerSecond":305461.3198607744,
            "cloneID":1,
            "currentBytes":81788928,
            "dstServiceID":16,
            "dstVolumeID":6,
            "elapsedTime":291.7847648200743,
            "nodeID":1,
            "percentComplete":8.167539267015707,
            "remainingTime":3280.708270981153,
            "sliceID":6,
            "srcServiceID":16,
            "srcVolumeID":5,
            "stage":"whole",
            "totalBytes":1001390080,
            "type":"clone"
         },
         {
            "blocksPerSecond":0,
            "branchType": "snapshot",
            "dstServiceID":8,
            "dstVolumeID":2,
            "elapsedTime":0,
            "percentComplete":0,
            "remainingTime":0,
            "sliceID":2,
            "stage":"metadata",
            "type":"remote"
        }
      ]
    }
}
```


New since version

9.6

ListVolumeQoSHistograms

You can use the `ListVolumeQoSHistograms` method to generate a histogram of volume QoS usage for one volume or multiple volumes. This enables you to better understand how volumes are using QoS.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeIDs	An optional list of volume IDs specifying which volumes should have QoS histograms generated.	integer array	None	No

Return value

This method has the following return value:

Name	Description	Type
qosHistograms	A list of objects describing volume usage for one or more volumes.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
    "method": "ListVolumeQoSHistograms",
        "params": {
            "volumeIDs": [1]
        },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
```

```
    "id": 1,
    "result": {
        "qosHistograms": [
        {
            "histograms": {
            "belowMinIopsPercentages": {
                "Bucket1To19": 2406,
                "Bucket20To39": 3,
                "Bucket40To59": 0,
                "Bucket60To79": 4,
                "Bucket80To100": 0
            },
                "minToMaxIopsPercentages": {
                "Bucket101Plus": 0,
                "Bucket1To19": 0,
                "Bucket20To39": 0,
                "Bucket40To59": 2,
                "Bucket60To79": 0,
                "Bucket80To100": 0
            },
                "readBlockSizes": {
                "Bucket131072Plus": 0,
                "Bucket16384To32767": 0,
                "Bucket32768To65535": 0,
                "Bucket4096To8191": 0,
                "Bucket65536To131071": 0,
                "Bucket8192To16383": 0
            },
                "targetUtilizationPercentages": {
                "Bucket0": 134943,
                "Bucket101Plus": 0,
                "Bucket1To19": 2409,
                "Bucket20To39": 4,
                "Bucket40To59": 0,
                "Bucket60To79": 2,
                "Bucket80To100": 0
            },
                "throttlePercentages": {
                "Bucket0": 137358,
                "Bucket1To19": 0,
                "Bucket20To39": 0,
                "Bucket40To59": 0,
                "Bucket60To79": 0,
                "Bucket80To100": 0
            },
                "writeBlockSizes": {
```

```
                "Bucket131072Plus": 0,
                "Bucket16384To32767": 0,
                "Bucket32768To65535": 0,
                "Bucket4096To8191": 0,
                "Bucket65536To131071": 0,
                "Bucket8192To16383": 0
            }
        },
            "timestamp": "2018-06-21T18:45:52.010844Z",
            "volumeID": 1
        }
    ]
}
}
```

ListVolumes

You can use the `ListVolumes` method to get a list of volumes that are in a cluster. You can specify the volumes you want to return in the list by using the available parameters.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
accounts	Only volumes owned by the accounts you specify here are returned. Mutually exclusive with the volumeIDs parameter.	integer array	None	No
includeVirtualVolumes	Virtual volumes are included in the response by default. To exclude virtual volumes, set to false.	boolean	true	No

Name	Description	Type	Default value	Required
isPaired	Returns volumes that are paired or not paired. Possible values: • true: Returns all paired volumes. • false: Returns all volumes not paired.	boolean	None	No
limit	Enables you to set the maximum number of volume results that are returned. Mutually exclusive with the volumeIDs parameter.	integer	10000	No
startVolumeID	Only volumes with an ID greater than or equal to this value are returned. Mutually exclusive with the volumeIDs parameter.	integer	None	No
volumeIDs	A list of volume IDs. If you specify this parameter, other parameters operate only on this set of volumes. Mutually exclusive with the accounts, startVolumeID, and limit parameters.	integer array	No	No
volumeName	Only volume object information matching the volume name is returned.	string	No	No

Name	Description	Type	Default value	Required
volumeStatus	Only volumes with a status equal to the status value are returned. Possible values: • creating • snapshotting • active • deleted	string	No	No

Return value

This method has the following return value:

Name	Description	Type
volumes	List of volumes.	volume array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListVolumes",
    "params": {
        "volumeIDs": [1],
        "volumeStatus": "active",
        "isPaired": "false"
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "volumes": [
            {
                "access": "readWrite",
                "accountID": 1,
                "attributes": {},
                "blockSize": 4096,
                "createTime": "2016-03-28T14:39:05Z",
                "deleteTime": "",
                "enable512e": true,
                "iqn": "iqn.2010-01.com.solidfire:testvolume1.1",
                "name": "testVolume1",
                "purgeTime": "",
                "qos": {
                    "burstIOPS": 15000,
                    "burstTime": 60,
                    "curve": {
                        "4096": 100,
                        "8192": 160,
                        "16384": 270,
                        "32768": 500,
                        "65536": 1000,
                        "131072": 1950,
                        "262144": 3900,
                        "524288": 7600,
                        "1048576": 15000
                    },
                    "maxIOPS": 15000,
                    "minIOPS": 50
                },
                "scsiEUIDeviceID": "6a79617900000001f47acc0100000000",
                "scsiNAADeviceID": "6f47acc1000000006a79617900000001",
                "sliceCount": 1,
                "status": "active",
                "totalSize": 5000658944,
                "virtualVolumeID": null,
                "volumeAccessGroups": [],
                "volumeID": 1,
                "volumePairs": []
            }
        ]
    }
}
```

New since version

9.6

ListVolumeStats

You can use the `ListVolumeStats` method to get high-level activity measurements for a single volume, list of volumes, or all volumes (if you omit the volumeIDs parameter). Measurement values are cumulative from the creation of the volume.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
includeVirtualVolumes	Virtual volumes are included in the response by default. To exclude virtual volumes, set to false.	boolean	true	No
volumeIDs	A list of volumes from which to retrieve activity information.	integer array	No	No

Return value

This method has the following return value:

Name	Description	Type
volumeStats	List of volume activity information.	volumeStats array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListVolumeStats",
    "params": {
        "volumeIDs": [1]
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:


```
{
  "id": 1,
  "result": {
    "volumeStats": [
      {
        "accountID": 1,
        "actualIOPS": 0,
        "asyncDelay": null,
        "averageIOPSize": 0,
        "burstIOPSCredit": 30000,
        "clientQueueDepth": 0,
        "desiredMetadataHosts": null,
        "latencyUSec": 0,
        "metadataHosts": {
          "deadSecondaries": [],
          "liveSecondaries": [
            47
          ],
          "primary": 33
        },
        "nonZeroBlocks": 22080699,
        "readBytes": 657262370816,
        "readBytesLastSample": 0,
        "readLatencyUSec": 0,
        "readOps": 160464446,
        "readOpsLastSample": 0,
        "samplePeriodMSec": 500,
        "throttle": 0,
        "timestamp": "2016-03-09T19:39:15.771697Z",
        "unalignedReads": 0,
        "unalignedWrites": 0,
        "volumeAccessGroups": [
          1
        ],
        "volumeID": 1,
        "volumeSize": 107374182400,
        "volumeUtilization": 0,
        "writeBytes": 219117547520,
        "writeBytesLastSample": 0,
        "writeLatencyUSec": 0,
        "writeOps": 53495495,
        "writeOpsLastSample": 0,
```

```
        "zeroBlocks": 4133701
      }
    ]
  }
}
```

New since version

9.6

ListVolumesForAccount

You can use the `ListVolumesForAccount` method to list active and (pending) deleted volumes for an account.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
includeVirtualVolumes	Virtual volumes are included in the response by default. To exclude virtual volumes, set to false.	boolean	true	No
accountID	All volumes owned by this accountID are returned.	integer	No	Yes

Return value

This method has the following return value:

Name	Description	Type
volumes	List of volume information.	volume array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListVolumesForAccount",
   "params": {
      "accountID"  : 1
   },
   "id" : 1
}
```

Response example

Responses for this method are similar to the following example:

```
{
    "id": 1,
    "result": {
        "volumes": [
            {
                "access": "readWrite",
                "accountID": 1,
                "attributes": {},
                "blockSize": 4096,
                "createTime": "2018-07-22T16:15:25Z",
                "deleteTime": "",
                "enable512e": false,
                "iqn": "iqn.2010-01.com.solidfire:0oto.test1.25",
                "name": "test1",
                "purgeTime": "",
                "qos": {
                    "burstIOPS": 15000,
                    "burstTime": 60,
                    "curve": {
                        "4096": 100,
                        "8192": 160,
                        "16384": 270,
                        "32768": 500,
                        "65536": 1000,
                        "131072": 1950,
                        "262144": 3900,
                        "524288": 7600,
                        "1048576": 15000
                    },
                    "maxIOPS": 15000,
                    "minIOPS": 50
                },
                "scsiEUIDeviceID": "306f746f00000019f47acc0100000000",
                "scsiNAADeviceID": "6f47acc100000000306f746f00000019",
                "sliceCount": 1,
                "status": "active",
                "totalSize": 1000341504,
                "virtualVolumeID": null,
                "volumeAccessGroups": [],
                "volumeID": 25,
                "volumePairs": []
            }
        ]
    }
}
```

New since version

9.6

ListVolumeStatsByAccount

You can use the `ListVolumeStatsByAccount` method to list high-level volume activity measurements for every account. Values are summed from all volumes owned by the account.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
includeVirtualVolumes	Virtual volumes are included in the response by default. To exclude virtual volumes, set to false.	boolean	true	No
accounts	A list of account IDs for which to return volume statistics. If omitted, statistics for all accounts are returned.	integer array	None	No

Return value

This method has the following return value:

Name	Description	Type
volumeStats	List of volume activity information for each account.**Note:** The volumeID member is 0 for each entry, as the values represent the summation of all volumes owned by the account.	volumeStats array

Request example

Requests for this method are similar to the following example:

```
{
"method": "ListVolumeStatsByAccount",
  "params": {"accounts": [3]},
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volumeStats": [
      {
        "accountID": 3,
        "nonZeroBlocks": 155040175,
        "readBytes": 3156273328128,
        "readBytesLastSample": 0,
        "readOps": 770574543,
        "readOpsLastSample": 0,
        "samplePeriodMSec": 500,
        "timestamp": "2016-10-17T20:42:26.231661Z",
        "unalignedReads": 0,
        "unalignedWrites": 0,
        "volumeAccessGroups": [],
        "volumeID": 0,
        "volumeSize": 1127428915200,
        "writeBytes": 1051988406272,
        "writeBytesLastSample": 0,
        "writeOps": 256833107,
        "writeOpsLastSample": 0,
        "zeroBlocks": 120211025
      }
    ]
  }
}
```

New since version

9.6

ListVolumeStatsByVirtualVolume

You can use the `ListVolumeStatsByVirtualVolume` method to list volume statistics for any volumes in the system that are associated with virtual volume. Statistics are cumulative from the creation of the volume.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
virtualVolumeIDs	A list of one or more virtual volume IDs for which to retrieve information. If you specify this parameter, the method returns information about only these virtual volumes.	UUID string array	No	No

Return value

This method has the following return value:

Name	Description	Type
volumeStats	A list of objects containing activity information for each virtual volume in the system.	volumeStats array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListVolumeStatsByVirtualVolume",
  "params": {},
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volumeStats": [
      {
        "accountID": 17,
        "actualIOPS": 0,
        "asyncDelay": null,
        "averageIOPSize": 1074265444,
        "burstIOPSCredit": 0,
        "clientQueueDepth": 0,
        "desiredMetadataHosts": null,
        "latencyUSec": 0,
        "metadataHosts": {
          "deadSecondaries": [],
          "liveSecondaries": [
            26
          ],
          "primary": 56
        },
        "nonZeroBlocks": 36,
        "readBytes": 18366464,
        "readBytesLastSample": 0,
        "readLatencyUSec": 0,
        "readOps": 156,
        "readOpsLastSample": 0,
        "samplePeriodMSec": 500,
        "throttle": 0,
        "timestamp": "2016-10-10T17:46:35.914642Z",
        "unalignedReads": 156,
        "unalignedWrites": 185,
        "virtualVolumeID": "070ac0ba-f344-4f4c-b79c-142efa3642e8",
        "volumeAccessGroups": [],
        "volumeID": 12518,
        "volumeSize": 91271200768,
        "volumeUtilization": 0,
        "writeBytes": 23652213248,
        "writeBytesLastSample": 0,
        "writeLatencyUSec": 0,
        "writeOps": 185,
        "writeOpsLastSample": 0,
        "zeroBlocks": 22282972
      }
    ]
  }
}
```

New since version

9.6

ListVolumeStatsByVolume

You can use the `ListVolumeStatsByVolume` method to list high-level activity measurements for every volume, by volume. Values are cumulative from the creation of the volume.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
includeVirtualVolumes	Virtual volumes are included in the response by default. To exclude virtual volumes, set to false.	boolean	true	No

Return value

This method has the following return value:

Name	Description	Type
volumeStats	List of volume activity information.	volumeStats array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListVolumeStatsByVolume",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
```

```
  "result": {
    "volumeStats": [
        {
            "accountID": 1,
            "actualIOPS": 4,
            "asyncDelay": null,
            "averageIOPSize": 5970,
            "burstIOPSCredit": 0,
            "clientQueueDepth": 0,
            "desiredMetadataHosts": null,
            "latencyUSec": 474,
            "metadataHosts": {
                "deadSecondaries": [],
                "liveSecondaries": [
                    13
                ],
                "primary": 25
            },
            "nonZeroBlocks": 34931222,
            "normalizedIOPS": 4,
            "readBytes": 1282491003392,
            "readBytesLastSample": 0,
            "readLatencyUSec": 0,
            "readLatencyUSecTotal": 4581669750,
            "readOps": 15592933,
            "readOpsLastSample": 0,
            "samplePeriodMSec": 500,
            "sliceIopsStats": {
                "largeStatistics": {
                    "averageReadIops": 17,
                    "averageTotalIops": 43,
                    "averageWriteIops": 26,
                    "nSamples": 24,
                    "peakReadIops": 19,
                    "peakTotalIops": 47,
                    "peakWriteIops": 30,
                    "sliceID": 1
                },
                "smallStatistics": {
                    "averageReadIops": 17,
                    "averageTotalIops": 42,
                    "averageWriteIops": 25,
                    "nSamples": 120,
                    "peakReadIops": 173,
                    "peakTotalIops": 249,
                    "peakWriteIops": 77,
```

```
                    "sliceID": 1
                }
            },
            "throttle": 0,
            "timestamp": "2025-02-03T21:18:38.880100Z",
            "unalignedReads": 167319,
            "unalignedWrites": 90836,
            "volumeAccessGroups": [
                1
            ],
            "volumeID": 1,
            "volumeSize": 2147483648000,
            "volumeUtilization": 0.0002666666666666667,
            "writeBytes": 1385173585408,
            "writeBytesLastSample": 12288,
            "writeLatencyUSec": 474,
            "writeLatencyUSecTotal": 11233350905,
            "writeOps": 157060458,
            "writeOpsLastSample": 2,
            "zeroBlocks": 489356778
        }
    ]
  }
}
```


New since version

9.6

ListVolumeStatsByVolumeAccessGroup

You can use the `ListVolumeStatsByVolumeAccessGroup` method to list total activity measurements for all of the volumes that are members of the specified volume access groups.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
includeVirtualVolumes	Virtual volumes are included in the response by default. To exclude virtual volumes, set to false.	boolean	true	No
volumeAccessGroups	An array of VolumeAccessGroupIDs for which volume activity is returned. If omitted, statistics for all volume access groups are returned.	integer array	None	No

Return value

This method has the following return value:

Name	Description	Type
volumeStats	List of volume activity information for all volumes in the specified volume access group.**Note:** The volumeID member is 0 for each entry, because the values represent the summation of all volumes owned by the account.	volumeStats

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListVolumeStatsByVolumeAccessGroup",
    "params": {"volumeAccessGroups": [1]},
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volumeStats": [
      {
        "accountID": 0,
        "nonZeroBlocks": 149366393,
        "readBytes": 3156273328128,
        "readBytesLastSample": 0,
        "readOps": 770574543,
        "readOpsLastSample": 0,
        "samplePeriodMSec": 500,
        "timestamp": "2016-10-17T21:04:10.712370Z",
        "unalignedReads": 0,
        "unalignedWrites": 0,
        "volumeAccessGroups": [
            1
        ],
        "volumeID": 0,
        "volumeSize": 1073741824000,
        "writeBytes": 1051988406272,
        "writeBytesLastSample": 0,
        "writeOps": 256833107,
        "writeOpsLastSample": 0,
        "zeroBlocks": 112777607
      }
    ]
  }
}
```


New since version

9.6

ModifyBackupTarget

You can use the `ModifyBackupTarget` method to change attributes of a backup target.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
backupTargetID	Unique target ID for the target to modify.	integer	None	Yes
attributes	List of name-value pairs in JSON object format.	JSON object	None	No
name	New name for the backup target.	string	None	No

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ModifyBackupTarget",
   "params": {
      "backupTargetID" : 1,
      "name": "yourtargetS3"
      "attributes" : {
          "size" : 500,
      }
   },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {}
}
```

New since version

9.6

ModifyQoSPolicy

You can use the `ModifyQoSPolicy` method to modify an existing QoS policy on the system.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
qosPolicyID	The ID of the policy to be modified.	integer	None	Yes
name	If supplied, the name of the QoS policy (e.g. gold, platinum, silver) is changed to this value.	string	None	No
qos	If supplied, the QoS settings for this policy are changed to these settings. You can supply partial QoS values and only change some of the QoS settings.	QoS object	None	No

Return values

This method has the following return values:

Name	Description	Type
qosPolicy	Details of the newly modified QoS policy.	QoSPolicy

Request example

Requests for this method are similar to the following example:

```
{
  "id": 1950,
  "method": "ModifyQoSPolicy",
  "params": {
    "qosPolicyID": 2,
    "qos": {
      "minIOPS": 51,
      "maxIOPS": 15002,
      "burstIOPS": 15002
    }
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1950,
  "result": {
    "qosPolicy": {
      "name": "bronze",
      "qos": {
        "burstIOPS": 15002,
        "burstTime": 60,
        "curve": {
          "4096": 100,
          "8192": 160,
          "16384": 270,
          "32768": 500,
          "65536": 1000,
          "131072": 1950,
          "262144": 3900,
          "524288": 7600,
          "1048576": 15000
        },
        "maxIOPS": 15002,
        "minIOPS": 51
      },
      "qosPolicyID": 2,
      "volumeIDs": [
        2
      ]
    }
  }
}
```


New since version

10.0

ModifyVolume

You can use the `ModifyVolume` method to modify settings on an existing volume. You can make modifications to one volume at a time and changes take place immediately.

If you do not specify QoS values when you modify a volume, they remain the same as before the modification. You can retrieve default QoS values for a newly created volume by running the `GetDefaultQoS` method.

When you need to increase the size of a volume that is being replicated, do so in the following order to prevent replication errors:

1. Increase the size of the volume with replicationTarget access.

2. Increase the size of the source or the volume with readWrite access.

Ensure that both the target and source volumes are the same size.

If you change the access status to locked or replicationTarget, all existing iSCSI connections are terminated.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeID	The volumeID for the volume to be modified.	integer	None	Yes

Name	Description	Type	Default value	Required
access	Access allowed for the volume. Possible values: • `readOnly`: Only read operations are allowed. • `readWrite`: Reads and writes are allowed. • `locked`: No reads or writes are allowed. If not specified, the access value does not change. • `replicationTarget`: Identify a volume as the target volume for a paired set of volumes. If the volume is not paired, the access status is locked. If a value is not specified, the access value does not change. • `snapMirrorTarget`: Identify a volume as the target volume for SnapMirror replication.	string	None	No
accountID	The accountID to which the volume is reassigned. If none is specified, the previous account name is used.	integer	None	No

Name	Description	Type	Default value	Required
associateWithQoSPolicy	Associate the volume with the specified QoS policy. Possible values: • `true`: Associate the volume with the QoS policy specified in the QoSPolicyID parameter. • `false`: Do not associate the volume with the QoS policy specified in the QoSPolicyID parameter. When false, any existing policy association is removed, regardless of whether you specify a QoS policy in the QoSPolicy parameter.	boolean	None	No
attributes	List of name-value pairs in JSON object format.	JSON object	None	No
createTime	An ISO 8601 date string to set as the new volume creation date. Required if setCreateTime is set to true.	ISO 8601 string	None	No

Name	Description	Type	Default value	Required
dontMoveforIopsRebalance	Prevent a volume from moving when load balancing with actual IOPS. This setting is available beginning with Element 12.8 and is only effective when VolumeLoadBalanceOnActualIOPS is enabled. Possible values: • `true`: Volume does not load balance based on actual IOPS. • `false`: Volume does load balance based on actual IOPS. **Note**: Balancing for high availability (node failure) takes precedence over `VolumeLoadBalanceOnActualIOPS` and `dontMoveforIopsRebalance`.	boolean	false	No
enableSnapMirrorReplication	Determines whether the volume can be used for replication with SnapMirror endpoints. Possible values: • `true` • `false`	boolean	false	No

Name	Description	Type	Default value	Required
fifoSize	Specifies the maximum number of First-In-First-Out (FIFO) snapshots supported by the volume. Note that FIFO and non-FIFO snapshots both use the same pool of available snapshot slots on a volume. Use this option to limit FIFO snapshot consumption of the available snapshot slots. Note that you cannot modify this value to be less than the current FIFO snapshot count.	integer	None	No
minFifoSize	Specifies the number of snapshot slots that are reserved for only First-In-First-Out (FIFO) snapshots. Since FIFO and non-FIFO snapshots share the same pool, the minFifoSize parameter reduces the total number of possible non-FIFO snapshots by the same amount. Note that you cannot modify this value so that it conflicts with the current non-FIFO snapshot count.	integer	None	No

Name	Description	Type	Default value	Required
mode	Volume replication mode. Possible values: • `asynch`: Waits for system to acknowledge that data is stored on source before writing to the target. • `sync`: Does not wait for data transmission acknowledgment from source to begin writing data to the target.	string	None	No
qos	The new quality of service settings for this volume. If not specified, the QoS settings are not changed. Possible values: • `minIOPS` • `maxIOPS` • `burstIOPS`	QoS	None	No
qosPolicyID	The ID for the policy whose QoS settings should be applied to the specified volumes. This parameter is mutually exclusive with the qos parameter.	integer	None	No
setCreateTime	Set to true to change the recorded date of volume creation.	boolean	None	No

Name	Description	Type	Default value	Required
totalSize	The new size of the volume in bytes. 1000000000 is equal to 1GB. Size is rounded up to the nearest megabyte in size. This parameter can only be used to increase the size of a volume.	integer	None	No

Return value

This method has the following return value:

Name	Description	Type
volume	Object containing information about the newly modified volume.	volume

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ModifyVolume",
  "params": {
     "volumeID": 319,
     "access": "readWrite",
     "dontMoveForIopsRebalance": false
     },
     "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
      "volume": {
          "access": "readWrite",
          "accountID":22,
          "attributes": {},
```

```
          "blockSize": 4096,
          "createTime": "2024-04-01T19:39:40Z",
          "currentProtectionScheme": "doubleHelix",
          "deleteTime": "",
          "dontMoveForIopsRebalance": false,
          "enable512e": false,
          "enableSnapMirrorReplication": false,
          "fifoSize": 24,
          "iqn": "iqn.2010-01.com.solidfire:3eeu.suite40.319",
          "lastAccessTime": "2024-04-02T12:41:34Z",
          "lastAccessTimeIO": "2024-04-01T20:41:19Z",
          "minFifoSize": 0,
          "name": "suite40",
          "previousProtectionScheme": null,
          "purgeTime": "",
          "qos": {
              "burstIOPS": 27000,
              "burstTime": 60,
              "curve": {
                  "1048576": 15000,
                  "131072": 1950,
                  "16384": 270,
                  "262144": 3900,
                  "32768": 500, "4096": 100,
                  "524288": 7600,
                  "65536": 1000,
                  "8192": 160
              },
              "maxIOPS": 27000,
              "minIOPS": 500
          },
          "qosPolicyID": null,
          "scsiEUIDeviceID": "336565750000013ff47acc0100000000",
          "scsiNAADeviceID": "6f47acc1000000003365657500000013f",
          "sliceCount": 1,
          "status": "active",
          "totalSize": 1000000716800,
          "virtualVolumeID": null,
          "volumeAccessGroups": [
          22
          ],
          "volumeConsistencyGroupUUID": "3003109e-6e75-444c-8cee-
470d641a09c3",
          "volumeID": 319,
          "volumePairs": [],
          "volumeUUID": "78203136-b0eb-454b-9f67-2c867ec7d7bb"
```

```
        }
    }
}
```

New since version

9.6

Find more information

GetDefaultQoS

ModifyVolumes

You can use the `ModifyVolumes` method to configure up to 500 existing volumes at one time. Changes take place immediately. If `ModifyVolumes` fails to modify any of the specified volumes, none of the specified volumes are changed.

If you do not specify QoS values when you modify volumes, the QoS values for each volume remain unchanged. You can retrieve default QoS values for a newly created volume by running the `GetDefaultQoS` method.

When you need to increase the size volumes that are being replicated, do so in the following order to prevent replication errors:

1. Increase the size of the volume with replicationTarget access.
2. Increase the size of the source or the volume with readWrite access.

Ensure that both the target and source volumes are the same size.

> If you change the access status to locked or replicationTarget, all existing iSCSI connections are terminated.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
access	Access allowed for the volumes. Possible values: • `readOnly`: Only read operations are allowed. • `readWrite`: Reads and writes are allowed. • `locked`: No reads or writes are allowed. If not specified, the access value does not change. • `replicationTarget`: Identify a volume as the target volume for a paired set of volumes. If the volume is not paired, the access status is locked. If a value is not specified, the access value does not change.	string	None	No
accountID	The accountID to which the volumes are reassigned. If none is specified, the previous account name is used.	integer	None	No

Name	Description	Type	Default value	Required
associateWithQoSPolicy	Associate the volume with the specified QoS policy. Possible values: • `true`: Associate the volume with the QoS policy specified in the QoSPolicyID parameter. • `false`: Do not associate the volume with the QoS policy specified in the QoSPolicyID parameter. When false, any existing policy association is removed, regardless of whether you specify a QoS policy in the QoSPolicy parameter.	boolean	None	No
attributes	List of name-value pairs in JSON object format.	JSON object	None	No
createTime	An ISO 8601 date string to set as the new volume creation date. Required if setCreateTime is set to true.	ISO 8601 string	None	No

Name	Description	Type	Default value	Required
dontMoveforIopsRebalance	Prevent a volume from moving when load balancing with actual IOPS. This setting is available beginning with Element 12.8 and is only effective when VolumeLoadBalanceOnActualIOPS is enabled. Possible values: • `true`: Volume does not load balance based on actual IOPS. • `false`: Volume does load balance based on actual IOPS. **Note**: Balancing for high availability (node failure) takes precedence over `VolumeLoadBalanceOnActualIOPS` and `dontMoveforIopsRebalance`.	boolean	false	No
enableSnapMirrorReplication	Determines whether the volume can be used for replication with SnapMirror endpoints. Possible values: • `true` • `false`	boolean	false	No

Name	Description	Type	Default value	Required
fifoSize	Specifies the maximum number of First-In-First-Out (FIFO) snapshots supported by the volume. Note that FIFO and non-FIFO snapshots both use the same pool of available snapshot slots on a volume. Use this option to limit FIFO snapshot consumption of the available snapshot slots. Note that you cannot modify this value to be less than the current FIFO snapshot count.	integer	None	No
minFifoSize	Specifies the number of snapshot slots that are reserved for only First-In-First-Out (FIFO) snapshots. Since FIFO and non-FIFO snapshots share the same pool, the minFifoSize parameter reduces the total number of possible non-FIFO snapshots by the same amount. Note that you cannot modify this value so that it conflicts with the current non-FIFO snapshot count.	integer	None	No

Name	Description	Type	Default value	Required
mode	Volume replication mode. Possible values: • `asynch`: Waits for system to acknowledge that data is stored on source before writing to the target. • `sync`: Does not wait for data transmission acknowledgment from source to begin writing data to the target.	string	None	No
qos	The new quality of service settings for the volumes. If not specified, the QoS settings are not changed. Possible values: • `minIOPS` • `maxIOPS` • `burstIOPS`	QoS	None	No
qosPolicyID	The ID for the policy whose QoS settings should be applied to the specified volumes. This parameter is mutually exclusive with the qos parameter.	integer	None	No
setCreateTime	Set to true to change the recorded date of volume creation.	boolean	None	No

Name	Description	Type	Default value	Required
totalSize	The new size of the volumes in bytes. 1000000000 is equal to 1GB. Size is rounded up to the nearest megabyte in size. This parameter can only be used to increase the size of a volume.	integer	None	No
volumeIDs	A list of volumeIDs for the volumes to be modified.	integer array	None	Yes

Return value

This method has the following return value:

Name	Description	Type
volume	An array of objects containing information about each newly modified volume.	volume array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ModifyVolumes",
  "params": {
    "volumeIDs": [319,22],
    "access": "readWrite",
    "dontMoveForIopsRebalance": false
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
```

```
  "result": {
    "volumes":
      {
        "access":  "readWrite,
        "accountID": 22,
        "attributes": {},
        "blockSize": 4096,
        "createTime": "2024-04-01T19:39:40Z",
        "currentProtectionScheme": "doubleHelix",
        "deleteTime": "",
        "dontMoveForIopsRebalance": false,
        "enable512e": false,
        "enableSnapMirrorReplication": false,
        "fifoSize": 24,
        "iqn": "iqn.2010-01.com.solidfire:3eeu.suite40.319",
        "lastAccessTime": "2024-04-02T12:41:34Z",
        "lastAccessTimeIO": "2024-04-01T20:41:19Z",
        "minFifoSize": 0,
        "name": "suite40",
        "previousProtectionScheme": null,
        "purgeTime": "",
        "qos": {
          "burstIOPS": 27000,
          "burstTime": 60,
          "curve": {
            "1048576": 15000,
            "131072": 1950,
            "16384": 270,
            "262144": 3900,
            "32768": 500,
            "4096": 100,
            "524288": 7600,
            "65536": 1000,
            "8192": 160
          },
          "maxIOPS": 27000,
          "minIOPS": 500
        },
        "qosPolicyID": null,
        "scsiEUIDeviceID": "336565750000013ff47acc0100000000",
        "scsiNAADeviceID": "6f47acc1000000003365657500000013f",
        "sliceCount": 1,
        "status": "active",
        "totalSize": 1000000716800,
        "virtualVolumeID": null,
        "volumeAccessGroups": [
```

```
        22
        ],
        "volumeConsistencyGroupUUID": "3003109e-6e75-444c-8cee-
470d641a09c3",
        "volumeID": 319,
        "volumePairs": [],
        "
      }
  }
}
```

New since version

9.6

Find more information

GetDefaultQoS

PurgeDeletedVolume

You can use the `PurgeDeletedVolume` method to immediately and permanently purge a volume that has been deleted. You must delete a volume using `DeleteVolume` before it can be purged.

Volumes are purged automatically after a period of time, so usage of this method is not typically required.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
volumeID	The volumeID of the volume to be purged.	integer	No	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "PurgeDeletedVolume",
   "params": {
      "volumeID"  : 5
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result": {}
}
```

New since version

9.6

Find more information

DeleteVolume

PurgeDeletedVolumes

You can use the `PurgeDeletedVolumes` method to immediately and permanently purge volumes that have been deleted; you can use this method to purge up to 500 volumes at one time.

You must delete volumes using `DeleteVolumes` before they can be purged. Volumes are purged automatically after a period of time, so usage of this method is not typically required.

If you purge a large number of volumes at one time, or if the volumes you purge each have many associated snapshots, the method might fail and return the error "xDBConnectionLoss". If this happens, retry the method call again with fewer volumes.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeIDs	A list of volumeIDs of volumes to be purged from the system.	integer array	No	No
accountIDs	A list of accountIDs. All of the volumes from all of the specified accounts are purged from the system.	integer array	No	No
volumeAccessGroupIDs	A list of volumeAccessGroupIDs. All of the volumes from all of the specified volume access groups are purged from the system.	integer array	No	No

Note: You can specify only one of the above parameters per method call. Specifying more than one, or none, results in an error.

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "PurgeDeletedVolumes",
   "params": {
      "accountIDs" : [1, 2, 3]
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result": {}
}
```

New since version

9.6

Find more information

DeleteVolumes

RemoveBackupTarget

You can use the `RemoveBackupTarget` method to remove backup targets.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
backupTargetID	Unique target ID of the target to remove.	integer	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "RemoveBackupTarget",
   "params": {
        "backupTargetID" : 1
   },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {}
}
```

New since version

9.6

RestoreDeletedVolume

You can use the `RestoreDeletedVolume` method to mark a deleted volume as active again. This action makes the volume immediately available for iSCSI connection.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
volumeID	The volumeID of the deleted volume to restore.	integer	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "RestoreDeletedVolume",
   "params": {
      "volumeID"  : 5
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result": {}
}
```

New since version

9.6

SetDefaultQoS

You can use the `SetDefaultQoS` method to configure the default Quality of Service (QoS) values (measured in inputs and outputs per second, or IOPS) for a volume.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
minIOPS	The minimum number of sustained IOPS that are provided by the cluster to a volume.	integer	None	No
maxIOPS	The maximum number of sustained IOPS that are provided by the cluster to a volume.	integer	None	No
burstIOPS	The maximum number of IOPS allowed in a short burst scenario.	integer	None	No

Return values

This method has the following return values:

Name	Description	Type
minIOPS	The minimum number of sustained IOPS that are provided by the cluster to a volume.	integer

Name	Description	Type
maxIOPS	The maximum number of sustained IOPS that are provided by the cluster to a volume.	integer
burstIOPS	The maximum number of IOPS allowed in a short burst scenario.	integer

Request example

Requests for this method are similar to the following example:

```
{
   "method": "SetDefaultQoS",
    "params": {
        "burstIOPS":8000,
        "maxIOPS":1000,
        "minIOPS":200
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id":1,
    "result": {
        "burstIOPS":8000,
        "maxIOPS":1000,
        "minIOPS":200
    }
}
```

New since version

9.6

StartBulkVolumeRead

You can use the `StartBulkVolumeRead` method to start a bulk volume read session on a specified volume.

Only two bulk volume processes can run simultaneously on a volume. When you initialize the session, data is read from a SolidFire storage volume to be stored on an external backup source. The external data is accessed by a web server running on an Element storage node. Server interaction information for external data access is passed by a script running on the storage system.

At the start of a bulk volume read operation, a snapshot of the volume is made and the snapshot is deleted when the read has completed. You can also read a snapshot of the volume by entering the ID of the snapshot as a parameter. When you read a previous snapshot, the system does not create a new snapshot of the volume, nor does it delete the previous snapshot when the read completes.

This process creates a new snapshot if the ID of an existing snapshot is not provided. Snapshots can be created if cluster fullness is at stage 2 or 3. Snapshots are not created when cluster fullness is at stage 4 or 5.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
format	The format of the volume data. Can be either: • `uncompressed`: Every byte of the volume is returned without any compression. • `native`: Opaque data is returned that is smaller and more efficiently stored and written on a subsequent bulk volume write.	string	None	Yes
volumeID	The ID of the volume to be read.	integer	None	Yes
snapshotID	The ID of a previously created snapshot used for bulk volume reads. If no ID is entered, a snapshot of the current active volume image is made.	integer	None	No

Name	Description	Type	Default value	Required
script	The name of an executable script. If no script name is given, the key and URL are necessary to access Element storage nodes. The script is run on the primary node, and the key and URL are returned to the script so the local web server can be contacted.	string	None	No
scriptParameters	JSON parameters to pass to the script.	JSON object	None	No
attributes	List of name-value pairs in the JSON object format. Learn more.	JSON object	None	No

Return values

This method has the following return values:

Name	Description	Type
asyncHandle	The ID of the asynchronous process to be checked for completion.	integer
key	Opaque key uniquely identifying the session.	string
url	URL to access the node's web server.	string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "StartBulkVolumeRead",
   "params": {
      "volumeID" : 5,
      "format"  : "native",
      "snapshotID" : 2
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
      "id" : 1,
   "result" : {
      "asyncHandle" : 1,
      "key" : "11eed8f086539205beeaadd981aad130",
      "url" : "https://127.0.0.1:44000/"
   }
}
```

New since version

9.6

StartBulkVolumeWrite

You can use the `StartBulkVolumeWrite` method to start a bulk volume write session on a specified volume.

Only two bulk volume processes can run simultaneously on a volume. When you initialize the session, data is written to an Element storage volume from an external backup source. The external data is accessed by a web server running on an Element storage node. Server interaction information for external data access is passed by a script running on the storage system.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
format	The format of the volume data. Can be either: • `uncompressed`: Every byte of the volume is returned without any compression. • `native`: Opaque data is returned that is smaller and more efficiently stored and written on a subsequent bulk volume write.	string	None	Yes
volumeID	The ID of the volume to be written to.	integer	None	Yes
script	The name of an executable script. If no script name is given, the key and URL are necessary to access Element storage nodes. The script is run on the primary node, and the key and URL are returned to the script so the local web server can be contacted.	string	None	No
scriptParameters	JSON parameters to pass to the script.	JSON object	None	No
attributes	List of name-value pairs in the JSON object format. Learn more.	JSON object	None	No

Return values

This method has the following return values:

Name	Description	Type
asyncHandle	The ID of the asynchronous process to be checked for completion.	integer
key	Opaque key uniquely identifying the session.	string
url	URL to access the node's web server.	string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "StartBulkVolumeWrite",
   "params": {
      "volumeID" : 5,
      "format"  : "native",
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result" : {
      "asyncHandle" : 1,
      "key" : "11eed8f086539205beeaadd981aad130",
      "url" : "https://127.0.0.1:44000/"
   }
}
```

New since version

9.6

UpdateBulkVolumeStatus

You can use the `UpdateBulkVolumeStatus` method to update the status of a bulk volume job that you started with the `StartBulkVolumeRead` or `StartBulkVolumeWrite` methods.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
key	The key assigned during initialization of a StartBulkVolumeRead or StartBulkVolumeWrite session.	string	None	Yes
status	The system sets the status of the given bulk volume job. Possible values: • running: Jobs that are still active. • complete: Jobs that are done. • failed: Jobs that have failed.	string	None	Yes
percentComplete	The completed progress of the bulk volume job as a percentage.	string	None	No
message	Returns the status of the bulk volume job when the job has completed.	string	None	No
attributes	JSON attributes; updates what is on the bulk volume job.	JSON object	None	No

Return values

This method has the following return values:

Name	Description	Type
status	Status of the session requested. Returned status: • preparing • active • done • failed	string
attributes	Returns attributes that were specified in the method call. Values are returned whether they have changed or not.	string
url	The URL to access the node's web server; provided only if the session is still active.	string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "UpdateBulkVolumeStatus",
   "params": {
          "key": "0b2f532123225febda2625f55dcb0448",
          "status": "running"
      },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result": {
      "status" : "running",
      "url" : "https://10.10.23.47:8443/"
    }
}
```

New since version

9.6

Find more information

- StartBulkVolumeRead
- StartBulkVolumeWrite

Copyright information

Trademark information