



Drive API methods

Element Software

NetApp
June 11, 2021

Table of Contents

- Drive API methods 1
 - Find more information 1
 - AddDrives 1
 - GetDriveHardwareInfo 4
 - GetDriveStats 5
 - ListDrives 8
 - ListDriveStats 10
 - RemoveDrives 13
 - SecureEraseDrives 14

Drive API methods

You can use drive API methods to add and manage drives that are available to a storage cluster. When you add a storage node to the storage cluster or install new drives in an existing storage node, the drives are available to be added to the storage cluster.

- [AddDrives](#)
- [GetDriveHardwareInfo](#)
- [GetDriveStats](#)
- [ListDrives](#)
- [ListDriveStats](#)
- [RemoveDrives](#)
- [SecureEraseDrives](#)

Find more information

- [NetApp SolidFire Resources Page](#)
- [Documentation for earlier versions of NetApp SolidFire and Element products](#)

AddDrives

You can use the `AddDrives` method to add one or more available drives to the cluster, enabling the drives to host a portion of the data for the cluster.

When you add a storage node to the cluster or install new drives in an existing node, the new drives are marked as available and must be added via `AddDrives` before they can be utilized. Use the [ListDrives](#) method to display drives that are available to be added. When you add a drive, the system automatically determines the type of drive it should be.

The method is asynchronous and returns as soon as the processes for rebalancing the drives in the cluster are started. However, it might take more time for the data in the cluster to be rebalanced using the newly added drives; the rebalancing continues even after the `AddDrives` method call is complete. You can use the [GetAsyncResult](#) method to query the method's returned `asyncHandle`. After the `AddDrives` method returns, you can use the [ListSyncJobs](#) method to see the progress of the rebalancing of data with the new drives.



When you add multiple drives, it is more efficient to add them in a single `AddDrives` method call rather than multiple individual methods with a single drive each. This reduces the amount of data balancing that must occur to stabilize the storage load on the cluster.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
drives	<p>Information about each drive to be added to the cluster. Possible values:</p> <ul style="list-style-type: none"> • driveID: The ID of the drive to add (integer). • type: The type of drive to add (string). Valid values are "slice", "block", or "volume". If omitted, the system assigns the correct type. 	JSON object array	None	Yes (type is optional)

Return value

This method has the following return value:

Name	Description	Type
asyncHandle	Handle value used to obtain the operation result.	integer

Request example

Requests for this method are similar to the following example:

```
{
  "id": 1,
  "method": "AddDrives",
  "params": {
    "drives": [
      {
        "driveID": 1,
        "type": "slice"
      },
      {
        "driveID": 2,
        "type": "block"
      },
      {
        "driveID": 3,
        "type": "block"
      }
    ]
  }
}
```

Response example

This method returns a response similar to the following example:

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

New since version

9.6

Find more information

- [GetAsyncResult](#)
- [ListDrives](#)
- [ListSyncJobs](#)

GetDriveHardwareInfo

You can use the `GetDriveHardwareInfo` method to get all the hardware information for the given drive. This generally includes manufacturers, vendors, versions, and other associated hardware identification information.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
driveID	ID of the drive for the request.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
result	Returned hardware information for the specified driveID.	hardwareInfo

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetDriveHardwareInfo",
  "params": {
    "driveID": 5
  },
  "id" : 100
}
```

Response example

This method returns a response similar to the following example:

```

{
  "id" : 100,
  "result" : {
    "driveHardwareInfo" : {
      "description" : "ATA Drive",
      "dev" : "8:80",
      "devpath" :
"/devices/pci0000:40/0000:40:01.0/0000:41:00.0/host6/port-6:0/expander-
6:0/port-6:0:4/end_device-6:0:4/target6:0:4/6:0:4:0/block/sdf",
      "driveSecurityAtMaximum" : false,
      "driveSecurityFrozen" : false
      "driveSecurityLocked" : false,
      "logicalname" : "/dev/sdf",
      "product" : "INTEL SSDSA2CW300G3",
      "securityFeatureEnabled" : false,
      "securityFeatureSupported" : true,
      "serial" : "CVPR121400NT300EGN",
      "size" : "300069052416",
      "uuid" : "7e1fd5b9-5acc-8991-e2ac-c48f813a3884",
      "version" : "4PC10362"
    }
  }
}

```

New since version

9.6

Find more information

[ListDrives](#)

GetDriveStats

You can use the `GetDriveStats` method to get high-level activity measurements for a single drive. Values are cumulative from the addition of the drive to the cluster. Some values are specific to block drives. Statistical data is returned for either block or metadata drive types when you run this method.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
driveID	ID of the drive for the request.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
driveStats	Drive activity information for the specified driveID.	driveStats

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetDriveStats",
  "params": {
    "driveID": 3
  },
  "id" : 1
}
```

Response example (block drive)

This method returns a response similar to the following example for a block drive:


```
{
  "id": 1,
  "result": {
    "driveStats": {
      "driveID": 10,
      "failedDieCount": 0,
      "lifeRemainingPercent": 99,
      "lifetimeReadBytes": 26471661830144,
      "lifetimeWriteBytes": 13863852441600,
      "powerOnHours": 33684,
      "readBytes": 10600432105,
      "readOps": 5101025,
      "reallocatedSectors": 0,
      "reserveCapacityPercent": 100,
      "timestamp": "2016-10-17T20:23:45.456834Z",
      "totalCapacity": 300069052416,
      "usedCapacity": 6112226545,
      "usedMemory": 114503680,
      "writeBytes": 53559500896,
      "writeOps": 25773919
    }
  }
}
```

Response example (volume metadata drive)

This method returns a response similar to the following example for a volume metadata drive:

```
{
  "id": 1,
  "result": {
    "driveStats": {
      "activeSessions": 8,
      "driveID": 12,
      "failedDieCount": 0,
      "lifeRemainingPercent": 100,
      "lifetimeReadBytes": 2308544921600,
      "lifetimeWriteBytes": 1120986464256,
      "powerOnHours": 16316,
      "readBytes": 1060152152064,
      "readOps": 258826209,
      "reallocatedSectors": 0,
      "reserveCapacityPercent": 100,
      "timestamp": "2016-10-17T20:34:52.456130Z",
      "totalCapacity": 134994670387,
      "usedCapacity": null,
      "usedMemory": 22173577216,
      "writeBytes": 353346510848,
      "writeOps": 86266238
    }
  }
}
```

New since version

9.6

Find more information

[ListDrives](#)

ListDrives

You can use the [ListDrives](#) method to list the drives that exist in the active nodes of the cluster. This method returns drives that have been added as volume metadata or block drives as well as drives that have not been added and are available.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
drives	List of drives in the cluster.	drive array

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```

{
  "id" : 1,
  "result" : {
    "drives" : [
      {
        "attributes" : {},
        "capacity" : 299917139968,
        "driveID" : 35,
        "nodeID" : 5,
        "serial" : "scsi-SATA_INTEL_SSDSA2CW6CVPR141502R3600FGN-part2",
        "slot" : 0,
        "status" : "active",
        "type" : "volume"
      },
      {
        "attributes" : {},
        "capacity" : 600127266816,
        "driveID" : 36,
        "nodeID" : 5,
        "serial" : "scsi-SATA_INTEL_SSDSA2CW6CVPR1415037R600FGN",
        "slot" : 6,
        "status" : "active",
        "type" : "block"
      }
    ]
  }
}

```

New since version

9.6

ListDriveStats

You can use the `ListDriveStats` method to list high-level activity measurements for multiple drives in the cluster. By default, this method returns statistics for all drives in the cluster, and these measurements are cumulative from the addition of the drive to the cluster. Some values this method returns are specific to block drives, and some are specific to metadata drives.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
drives	List of drive IDs (driveID) for which to return drive statistics. If you omit this parameter, measurements for all drives are returned.	integer array	None	No

Return values

This method has the following return values:

Name	Description	Type
driveStats	List of drive activity information for each drive.	driveStats array
errors	This list contains the driveID and associated error message. It is always present, and empty if there are no errors.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
  "id": 1,
  "method": "ListDriveStats",
  "params": {
    "drives": [22,23]
  }
}
```

Response example

This method returns a response similar to the following example:

```

{
  "id": 1,
  "result": {
    "driveStats": [
      {
        "driveID": 22,
        "failedDieCount": 0,
        "lifeRemainingPercent": 84,
        "lifetimeReadBytes": 30171004403712,
        "lifetimeWriteBytes": 103464755527680,
        "powerOnHours": 17736,
        "readBytes": 14656542,
        "readOps": 3624,
        "reallocatedSectors": 0,
        "reserveCapacityPercent": 100,
        "timestamp": "2016-03-01T00:19:24.782735Z",
        "totalCapacity": 300069052416,
        "usedCapacity": 1783735635,
        "usedMemory": 879165440,
        "writeBytes": 2462169894,
        "writeOps": 608802
      }
    ],
    "errors": [
      {
        "driveID": 23,
        "exception": {
          "message": "xStatCheckpointDoesNotExist",
          "name": "xStatCheckpointDoesNotExist"
        }
      }
    ]
  }
}

```

New since version

9.6

Find more information

[GetDriveStats](#)

RemoveDrives

You can use the `RemoveDrives` method to proactively remove drives that are part of the cluster. You might use this method when reducing cluster capacity or preparing to replace drives nearing the end of their service life. Any data on the drives is removed and migrated to other drives in the cluster before the drive is removed from the cluster.

`RemoveDrives` is an asynchronous method. Depending on the total capacity of the drives being removed, it might take several minutes to migrate all of the data.

When removing multiple drives, use a single `RemoveDrives` method call rather than multiple individual methods with a single drive each. This reduces the amount of data balancing that must occur to evenly stabilize the storage load on the cluster.

You can also remove drives with a "failed" status using `RemoveDrives`. When you remove a drive with a "failed" status, the drive is not returned to an "available" or "active" status. The drive is unavailable for use in the cluster.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
drives	List of driveIDs to remove from the cluster.	integer array	None	Yes

Return value

This method has the following return value:

Name	Description	Type
asyncHandle	Handle value used to obtain the operation result.	integer

Request example

Requests for this method are similar to the following example:

```
{
  "method": "RemoveDrives",
  "params": {
    "drives" : [3, 4, 5]
  },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

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

New since version

9.6

Find more information

- [GetAsyncResult](#)
- [ListDrives](#)

SecureEraseDrives

You can use the `SecureEraseDrives` method to remove any residual data from drives that have a status of "available". You might use this method when replacing a drive nearing the end of its service life that contained sensitive data. This method uses a Security Erase Unit command to write a predetermined pattern to the drive and resets the encryption key on the drive. This asynchronous method might take several minutes to complete.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
drives	List of drive IDs to secure erase.	integer array	None	Yes

Return value

This method has the following return value:

Name	Description	Type
asyncHandle	Handle value used to obtain the operation result.	integer

Request example

Requests for this method are similar to the following example:

```
{
  "method": "SecureEraseDrives",
  "params": {
    "drives" : [3, 4, 5]
  },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

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

New since version

9.6

Find more information

- [GetAsyncResult](#)
- [ListDrives](#)

Copyright Information

Copyright © 2021 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means-graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system- without prior written permission of the copyright owner.

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

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

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

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

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark Information

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