



Replication API methods

Element Software

NetApp
March 05, 2025

Table of Contents

- Replication API methods 1
 - Find more information 1
 - Cluster pairing order of operations 1
 - Find more information 2
 - Volume pairing order of operations 2
 - Find more information 2
 - Supported modes of replication for paired clusters 2
- CompleteClusterPairing 3
 - Parameters 3
 - Return value 3
 - Request example 3
 - Response example 4
 - New since version 4
 - Find more information 4
- CompleteVolumePairing 4
 - Parameters 4
 - Return value 4
 - Request example 4
 - Response example 5
 - New since version 5
 - Find more information 5
- ListClusterPairs 5
 - Parameter 5
 - Return value 6
 - Request example 6
 - Response example 6
 - New since version 7
- ListActivePairedVolumes 7
 - Parameters 7
 - Return value 7
 - Request example 8
 - Response example 8
 - New since version 9
- ModifyVolumePair 10
 - Parameters 10
 - Return value 11
 - Request example 11
 - Response example 12
 - New since version 12
- RemoveClusterPair 12
 - Parameter 12
 - Return value 12
 - Request example 13

Response example	13
New since version	13
RemoveVolumePair	13
Parameter	13
Return value	13
Request example	14
Response example	14
New since version	14
StartClusterPairing	14
Parameter	14
Return values	14
Request example	15
Response example	15
New since version	15
Find more information	15
StartVolumePairing	16
Parameters	16
Return value	17
Request example	17
Response example	17
New since version	18
Find more information	18

Replication API methods

Replication API methods enable you to connect two clusters for continuous data protection (CDP). When you connect two clusters, active volumes within a cluster can be continuously replicated to a second cluster to provide data recovery. By pairing volumes for replication, you can protect your data from events that might render it inaccessible.

- [Cluster pairing order of operations](#)
- [Volume pairing order of operations](#)
- [Supported modes of replication for paired clusters](#)
- [CompleteClusterPairing](#)
- [CompleteVolumePairing](#)
- [ListClusterPairs](#)
- [ListActivePairedVolumes](#)
- [ModifyVolumePair](#)
- [RemoveClusterPair](#)
- [RemoveVolumePair](#)
- [StartClusterPairing](#)
- [StartVolumePairing](#)

Find more information

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

Cluster pairing order of operations

You must establish a connection between a pair of storage clusters running Element software before remote replication can be used.

Use the following set of API methods to establish a cluster connection:

- [StartClusterPairing](#):

This API method creates and returns a pairing key that is used to establish a cluster pair. The key is encoded and contains information that is used to establish communications between clusters. A single cluster can be paired with up to four other clusters. However, a new key must be generated for each cluster pairing. The [StartClusterPairing](#) method generates a new key each time the method is called. Use each unique key with the [CompleteClusterPairing](#) method to pair each additional cluster.



For security reasons, the pairing key should not be sent to other users via email. The key contains a user name and password.

- [CompleteClusterPairing](#):

This method uses the pairing key created with the [StartClusterPairing](#) API method to create a cluster pair. Issue the [CompleteClusterPairing](#) API method with the `clusterPairingKey` parameter to the destination. The origination cluster is the cluster that created the key.

Find more information

- [StartClusterPairing](#)
- [CompleteClusterPairing](#)

Volume pairing order of operations

You must create a cluster pair between two corresponding clusters before volumes can be paired.

Use the following set of API methods to establish a cluster connection:

- [StartVolumePairing](#):

This API method creates and returns a volume pairing key that is used to create a volume pair. The key contains information that is used to establish communications between volumes.

- [CompleteVolumePairing](#):

This method uses the pairing key created with the [StartVolumePairing](#) API method to create a volume pair. Issue the [CompleteVolumePairing](#) API method with the `volumeID` and `volumePairingKey` parameters to the destination volume.

Only one of the paired volumes can be identified as a replication target volume. Use the [ModifyVolumePair](#) API method to establish the direction of the volume's data replication by identifying which volume is the target. Data is replicated from the source volume to the target volume.

Find more information

- [StartVolumePairing](#)
- [CompleteVolumePairing](#)
- [ModifyVolumePair](#)

Supported modes of replication for paired clusters

The following modes of replication are supported on the paired clusters:

- Asynchronous replication of data: The data sent to the replication target volume is sent asynchronously. The system does not wait for an acknowledgment to be sent before writing data.
- Synchronous replication of data: The data sent to the replication target volume is sent synchronously. When the I/O operations sent from the host are acknowledged by the system, the system acknowledgment is sent back to the host and the data is sent to the replication target volume.
- Snapshots-only replication of data: Only volume snapshots are replicated to the target cluster.

CompleteClusterPairing

The CompleteClusterPairing method is the second step in the cluster pairing process. Use this method with the encoded key received from the StartClusterPairing method to complete the cluster pairing process.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
clusterPairingKey	A string of characters that is returned from the StartClusterPairing API method.	string	None	Yes

Return value

This method has the following return value:

Name	Description	Type
clusterPairID	Unique identifier for the cluster pair.	integer

Request example

Requests for this method are similar to the following example:

```
{
  "method": "CompleteClusterPairing",
  "params": {
    "clusterPairingKey" :
    "7b22636c7573746572506169724944223a312c22636c75737465725061697255554944223
    a2231636561313336322d346338662d343631612d626537322d37343536366139353364326
    6222c22636c7573746572556e697175654944223a2278736d36222c226d766970223a22313
    9322e3136382e3133392e313232222c226e616d65223a224175746f54657374322d6330755
    2222c2270617373776f7264223a22695e59686f20492d64774d7d4c67614b222c227270634
    36f6e6e656374696f6e4944223a3931333134323634392c22757365726e616d65223a225f5
    f53465f706169725f50597a796647704c7246564432444a42227d"
  },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

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

New since version

9.6

Find more information

[StartClusterPairing](#)

CompleteVolumePairing

You can use `CompleteVolumePairing` to complete the pairing of two volumes.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeID	The ID of volume that will complete the volume pair.	integer	None	Yes
volumePairingKey	The key returned from the StartVolumePairing API method.	string	None	Yes

Return value

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "CompleteVolumePairing",
  "params": {
    "volumeID" : 12,
    "volumePairingKey" :
"7b22636c7573746572506169724944223a312c22636c75737465725061697255554944223
a2231636561313336322d346338662d343631612d626537322d37343536366139353364326
6222c22636c75737465725556e697175654944223a2278736d36222c226d766970223a22313
9322e3136382e3133392e313232222c226e616d65223a224175746f54657374322d6330755
2222c2270617373776f7264223a22695e59686f20492d64774d7d4c67614b222c227270634
36f6e6e656374696f6e4944223a3931333134323634392c22757365726e616d65223a225f5
f53465f706169725f50597a796647704c7246564432444a42227d"
    },
  "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

[StartVolumePairing](#)

ListClusterPairs

You can use the `ListClusterPairs` method to list all clusters that are paired with the current cluster. This method returns information about active and pending cluster pairings, such as statistics about the current pairing as well as the connectivity and latency (in milliseconds) of the cluster pairing.

Parameter

This method has no input parameter:

Return value

This method has the following return value:

Name	Description	Type
clusterPairs	Information about each paired cluster.	clusterPair array

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```

{
  "id": 1,
  "result": {
    "clusterPairs": [
      {
        "clusterName": "cluster2",
        "clusterPairID": 3,
        "clusterPairUUID": "9866fbbeb-c2f8-4df3-beb9-58a5c4e49c9b",
        "clusterUUID": 5487,
        "latency": 1,
        "mvip": "172.1.1.5",
        "status": "Connected"
        "version": "8.0.0.1361"
      },
      {
        "clusterName": "cluster3",
        "clusterPairID": 2,
        "clusterPairUUID": "8132a699-ce82-41e0-b406-fb914f976042",
        "clusterUUID": 1383,
        "latency": 1,
        "mvip": "172.1.1.6",
        "status": "Connected"
        "version": "8.0.0.1361"
      }
    ]
  }
}

```

New since version

9.6

ListActivePairedVolumes

You can use the `ListActivePairedVolumes` method to list all of the active volumes paired with a volume. This method returns information about volumes with active and pending pairings.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
volumes	Volume information for the paired volumes.	volumePair array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListActivePairedVolumes",
  "params": {
    },
  "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": "2016-06-24T15:21:59Z",
        "deleteTime": "",
        "enable512e": true,
        "iqn": "iqn.2010-01.com.solidfire:0oto.bk.24",
        "name": "BK",
        "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": "306f746f00000018f47acc0100000000",
"scsiNAADeviceID": "6f47acc100000000306f746f00000018",
"sliceCount": 1,
"status": "active",
"totalSize": 10737418240,
"virtualVolumeID": null,
"volumeAccessGroups": [],
"volumeID": 24,
"volumePairs": [
    {
        "clusterPairID": 2,
        "remoteReplication": {
            "mode": "Async",
            "pauseLimit": 3145728000,
            "remoteServiceID": 14,
            "resumeDetails": "",
            "snapshotReplication": {
                "state": "Idle",
                "stateDetails": ""
            },
            "state": "Active",
            "stateDetails": ""
        },
        "remoteSliceID": 8,
        "remoteVolumeID": 8,
        "remoteVolumeName": "PairingDoc",
        "volumePairUUID": "229fcbf3-2d35-4625-865a-
d04bb9455cef"
    }
]
}
}
}

```

New since version

9.6

ModifyVolumePair

You can use the `ModifyVolumePair` method to pause or restart replication between a pair of volumes. This method is set on the source volume (the volume with read/write access).

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
<code>volumeID</code>	Identification number of the volume to be modified.	integer	None	Yes
<code>pausedManual</code>	Remote replication can be paused or restarted on the source (read/write) volume. Possible values: <ul style="list-style-type: none">• <code>true</code>: Pause volume replication.• <code>false</code>: Restart volume replication. If no value is specified, no change in replication is performed.	boolean	None	No

mode	<p>Volume replication mode. Possible values:</p> <ul style="list-style-type: none"> • Async: Writes are acknowledged when they complete locally. The cluster does not wait for writes to be replicated to the target cluster. • Sync: The source acknowledges the write when the data is stored locally and on the remote cluster. • SnapshotsOnly: Only snapshots created on the source cluster are replicated. Active writes from the source volume are not replicated. 	string	None	No
------	--	--------	------	----

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ModifyVolumePair",
  "params": {
    "pausedManual": false,
    "volumeID": 5,
    "mode": "sync"
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

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

New since version

9.6

RemoveClusterPair

You can use the `RemoveClusterPair` method to close the open connections between two paired clusters.

Parameter



Before you remove a cluster pair, you must first remove all volume pairing to the clusters with the `RemoveVolumePair` API method.

This method has the following input parameter:

Name	Description	Type	Default value	Required
clusterPairID	Unique identifier used to pair two clusters.	integer	None	Yes

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "RemoveClusterPair",
  "params": {
    "clusterPairID": 1
  },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

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

New since version

9.6

RemoveVolumePair

You can use the `RemoveVolumePair` method to remove the remote pairing between two volumes. Use this method on both the source and target volumes that are paired together. When you remove the volume pairing information, data is no longer replicated to or from the volume.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
volumeID	ID of the volume on which to stop the replication process.	integer	None	Yes

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "RemoveVolumePair",
  "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

StartClusterPairing

You can use the `StartClusterPairing` method to create an encoded key from a cluster that is used to pair with another cluster. The key created from this API method is used in the `CompleteClusterPairing` method to establish a cluster pairing. You can pair a cluster with a maximum of four other clusters.

Parameter

This method has no input parameter.

Return values

This method has the following return values:

Name	Description	Type
clusterPairingKey	A string of characters that is used by the CompleteClusterPairing API method.	string
clusterPairID	Unique identifier for the cluster pair.	integer

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "clusterPairID": 1,
    "clusterPairingKey":
"7b22636c7573746572506169724944223a312c22636c75737465725061697255554944223
a2231636561313336322d346338662d343631612d626537322d37343536366139353364326
6222c22636c7573746572556e697175654944223a2278736d36222c226d766970223a22313
9322e3136382e3133392e31323222c226e616d65223a224175746f54657374322d6330755
2222c2270617373776f7264223a22695e59686f20492d64774d7d4c67614b222c227270634
36f6e6e656374696f6e4944223a3931333134323634392c22757365726e616d65223a225f5
f53465f706169725f50597a796647704c7246564432444a42227d"
  }
}
```

New since version

9.6

Find more information

[CompleteClusterPairing](#)

StartVolumePairing

You can use the `StartVolumePairing` method to create an encoded key from a volume that is used to pair with another volume. The key that this method creates is used in the `CompleteVolumePairing` method to establish a volume pairing.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
mode	<p>The mode of the volume on which to start the pairing process. The mode can only be set if the volume is the source volume. Possible values:</p> <ul style="list-style-type: none">• <code>Async</code>: Writes are acknowledged when they complete locally. The cluster does not wait for writes to be replicated to the target cluster. (Default if no mode parameter specified.)• <code>Sync</code>: Source acknowledges write when the data is stored locally and on the remote cluster.• <code>SnapshotsOnly</code>: Only snapshots created on the source cluster are replicated. Active writes from the source volume are not replicated.	string	None	No

Name	Description	Type	Default value	Required
volumeID	The ID of the volume on which to start the pairing process.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
volumePairingKey	A string of characters that is used by the CompleteVolumePairing API method.	string

Request example

Requests for this method are similar to the following example:

```
{
  "method": "StartVolumePairing",
  "params": {
    "mode": "Async",
    "volumeID" : 14
  },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {
    "volumePairingKey" :
    "7b226d766970223a223139322e3136382e3133392e31323222c22766f6c756d654944223
    a312c22766f6c756d654e616d65223a2254657374222c22766f6c756d6550616972555494
    4223a2236393632346663622d323032652d343332352d613536392d6563396336353563376
    23561227d"
  }
}
```

New since version

9.6

Find more information

[CompleteVolumePairing](#)

Copyright information

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

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

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

Trademark information

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