volume file commands

ONTAP commands

NetApp
February 04, 2026

This PDF was generated from https://docs.netapp.com/us-en/ontap-cli-97/volume-file-compact-data.html on February 04, 2026. Always check docs.netapp.com for the latest.

Table of Contents

volume file commands . . . 1
volume file compact-data . . . 1
Description . . . 1
Parameters . . . 1
Examples . . . 1
volume file modify . . . 1
Description . . . 2
Parameters . . . 2
Examples . . . 2
volume file privileged-delete . . . 3
Description . . . 3
Parameters . . . 3
Examples . . . 3
volume file reservation . . . 3
Description . . . 3
Parameters . . . 3
Examples . . . 4
volume file show-disk-usage . . . 4
Description . . . 4
Parameters . . . 4
Examples . . . 5
volume file show-filehandle . . . 6
Description . . . 6
Parameters . . . 6
Examples . . . 7
volume file show-inode . . . 7
Description . . . 7
Parameters . . . 8
Examples . . . 9
volume file clone autodelete . . . 10
Description . . . 11
Parameters . . . 11
Examples . . . 11
volume file clone create . . . 12
Description . . . 12
Parameters . . . 12
Examples . . . 14
volume file clone show-autodelete . . . 15
Description . . . 15
Parameters . . . 15
Examples . . . 15
volume file clone deletion add-extension . . . 16
Description . . . 16

Parameters . . . 16
Examples . . . 16
volume file clone deletion modify . . . 16
Description . . . 16
Parameters . . . 16
Examples . . . 17
volume file clone deletion remove-extension . . . 17
Description . . . 17
Parameters . . . 17
Examples . . . 17
volume file clone deletion show . . . 17
Description . . . 18
Parameters . . . 18
Examples . . . 18
volume file clone split load modify . . . 19
Description . . . 19
Parameters . . . 19
Examples . . . 19
volume file clone split load show . . . 19
Description . . . 19
Parameters . . . 20
Examples . . . 20
volume file fingerprint abort . . . 21
Description . . . 21
Parameters . . . 21
Examples . . . 21
volume file fingerprint dump . . . 22
Description . . . 22
Parameters . . . 25
Examples . . . 25
volume file fingerprint show . . . 27
Description . . . 27
Parameters . . . 27
Examples . . . 28
volume file fingerprint start . . . 28
Description . . . 28
Parameters . . . 28
Examples . . . 29
volume file retention show . . . 29
Description . . . 29
Parameters . . . 29
Examples . . . 30

volume file commands

volume file compact-data

Apply Adaptive Data Compaction to a Snapshot copy of a file

Availability: This command is available to *cluster* administrators at the *admin* privilege level.

Description

The `volume file compact-data` command applies the Adaptive Data Compaction feature to the Snapshot copy of a file such that partially filled blocks from that file will merge and consume less storage space.

Parameters

`-node <nodename>` - Node

This parameter indicates the node name that the AWA instance runs on.

`-vserver <vserver name>` - Vserver Name

This specifies the Vserver in which the target file is located.

`-file </vol/<volume name>/<file path>>` - File Path

This specifies the complete file path. The Snapshot copy name can be specified as part of the path or by specifying the `-snapshot` parameter.

`[-volume <volume name>]` - Volume Name

This specifies the volume in which the targeted file is located.

`[-snapshot <snapshot name>]` - Snapshot Copy Name

This specifies the Snapshot copy name in which the file will be compacted.

Examples

The following command applies the Adaptive Data Compaction feature to the Snapshot copy *`snap1`* of the file */`file1`* in volume *`vol1`* :

```
cluster1::> volume file compact-data -vserver vs1 -volume vol1 -file
/vol/vol1/file1 -snapshot snap1
```

volume file modify

Manage the association of a QoS policy group with a file

Availability: This command is available to *cluster* administrators at the *admin* privilege level.

Description

This command adds and removes files from QoS policy groups. QoS policy groups define measurable service level objectives (SLOs) that apply to the storage objects with which the policy group is associated. A QoS policy group associated with this file can be created, modified, and deleted. You cannot associate a file to a QoS policy group if a LUN was created from the file.

Parameters

`-vserver <vserver name>` - Vserver Managing Volume

This specifies the Vserver on which the volume (containing the file) resides.

`-volume <volume name>` - Volume Name

This specifies the name of the volume. The name must be unique within the hosting Vserver.

`-file <text>` - File Path

This specifies the actual path of the file with respect to the volume.

{ `[-qos-policy-group <text>]` - QoS Policy Group Name

This option associates the file with a QoS policy group. This policy group manages storage system resources to deliver your desired level of service. If you do not assign a policy to a file, the system will not monitor and control the traffic to it. To remove this file from a QoS policy group, enter the reserved keyword “none”.

| `[-qos-adaptive-policy-group <text>]` - QoS Adaptive Policy Group Name }

This optional parameter specifies which QoS adaptive policy group to apply to the file. This policy group defines measurable service level objectives (SLOs) and Service Level Agreements (SLAs) that adjust based on the file’s allocated space or used space. To remove this file from an adaptive policy group, enter the reserved keyword "none".

`[-caching-policy <text>]` - Caching Policy Name

This optionally specifies the caching policy to apply to the file. A caching policy defines how the system caches this volume’s data in Flash Cache modules. If a caching policy is not assigned to this file, the system uses the caching policy that is assigned to the containing volume. If a caching policy is not assigned to the containing volume, the system uses the caching policy that is assigned to the containing Vserver. If a caching policy is not assigned to the containing Vserver, the system uses the default cluster-wide policy. The available caching policies are:

- none - Does not cache any user data or metadata blocks.
- auto - Read caches all metadata and randomly read user data blocks, and write caches all randomly overwritten user data blocks.

Default caching-policy is auto.

Examples

```
cluster1::> vol file modify -vserver vs0 -volume vs0_vol56 -file 1.txt
-qos-policy-group fast -cache all-read
```

Associates the file *`1.txt`* with the *`fast`* QoS policy group and *`all-read`* caching policy.

volume file privileged-delete

Perform a privileged-delete operation on unexpired WORM files on a SnapLock enterprise volume

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

The ` volume file privileged-delete ` command is used to perform a privileged-delete operation on unexpired WORM files on a SnapLock enterprise volume. The only built-in role that has access to the command is *"vsadmin-snaplock"* .

Parameters

-vserver <vserver name> - Vserver

Specifies the Vserver which hosts the SnapLock enterprise volume.

-file </vol/<volume name>/<file path>> - File Path

Specifies the absolute path of the file to be deleted. The value begins with /vol/<volumename>.

Examples

The following example deletes the unexpired WORM file *"/vol/vol1/wormfile"* . The file wormfile is stored in volume vol1 under Vserver *vserver1* .

```
vserver1::> volume file privileged-delete -file /vol/vol1/wormfile
     [Job 76] Job succeeded: Privileged-delete of File
"vs1:/vol/sle_vol1/wormfile" Completed.
```

volume file reservation

Get/Set the space reservation info for the named file.

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

The `volume file reservation` command can be used to query the space reservation settings for the named file, or to modify those settings. With no further modifiers, the command will report the current setting of the space reservation flag for a file. This tells whether or not space is reserved to fill holes in the file and to overwrite existing portions of the file that are also stored in a snapshot. For symlinks, the link is followed and the command operates on the link target.

Parameters

-vserver <vserver name> - Vserver Name

Specifies the Vserver on which the volume is located. If only one data Vserver exists, you do not need to specify this parameter.

-path </vol/<volume name>/<file path>> - File Name

Specifies the complete file path for which we want to get/set the space reservation settings.

[-is-enabled <text>] - enable | disable

Specifying enable or disable will turn the reservation setting on or off accordingly for the file.

Examples

The following example enables the file reservation setting for the file named file1. The file file1 is stored in volume testvol on Vserver vs0.

```
node::> file reservation -vserver vs0 /vol/testvol/file1 enable
space reservations for file /vol/testvol/file1: on.
```

volume file show-disk-usage

Show disk usage of file

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

This command requires a path to a file in a volume and displays the following information:

- Vserver name
- Total bytes used by the file in kilobytes
- Full Path to the file

If not logged in as Vserver administrator, the command also requires a Vserver name.

The "-instance" option provides the same result as the default as there are no extra fields to display.

Parameters

{ [-fields <fieldname>,…]

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

| [-h]

If this parameter is specified, the command displays total bytes used by the file in human readable form.

| [-k]

If this parameter is specified, the command displays total bytes used by the file in kilobytes.

| [-m]

If this parameter is specified, the command displays total bytes used by the file in megabytes.

| [-u]

If this parameter is specified, the command displays the unique bytes used by the file (bytes that are not shared with any other file in the volume due to deduplication or FlexClone files) in kilobytes.

| [-uh]

If this parameter is specified, the command displays the unique bytes used by the file in human readable form.

| [-uk]

If this parameter is specified, the command displays the unique bytes used by the file in kilobytes.

| [-um]

If this parameter is specified, the command displays the unique bytes used by the file in megabytes.

| [-instance] }

If you specify the `-instance` parameter, the command displays detailed information about all fields.

-vserver <vserver name> - Vserver

This parameter is used to specify the Vserver that contains the file for which the command displays the total bytes used. It is required if not logged in as Vserver administrator.

-path </vol/<volume name>/<file path>> - Full Path

This required parameter is used to specify the path of the file for which the command displays the total bytes used.

[-r, -range [start offset>:<end offset]] - Block Range

If this parameter is specified, the command displays the total bytes used by the file in the specified block range.

Examples

The following example displays the disk-usage of the file *file1.txt* in volume */vol/root_vs0* .

```
cluster1::> volume file show-disk-usage -vserver vs0 -path
/vol/root_vs0/file1.txt

Vserver          Total                Path
--------         ------               -----
vs0              1408KB               /vol/root_vs0/file1.txt
cluster1::> volume file show-disk-usage -m -vserver vs0 -path
/vol/root_vs0/file1.txt

Vserver          Total                Path
--------         ------               -----
vs0              1MB                  /vol/root_vs0/file1.txt
vs0::> volume file show-disk-usage -um -path /vol/root_vs0/file1.txt

Vserver       Total             Unique              Path
--------      -----             ------              -------
vs0           1MB               1MB
/vol/root_vs0/file1.txt
```

volume file show-filehandle

Show the file handle of a file

Availability: This command is available to *cluster* and *Vserver* administrators at the *advanced* privilege level.

Description

This command requires a path to a file in a volume and displays the file handle information described below:

- Vserver name
- Path to the file
- File handle flags
- Snapshot ID of the file (snapid)
- File ID
- File handle generation number
- File system ID (fsid)
- Master data set ID (msid)
- Data set ID (dsid)

If not logged in as a Vserver administrator, the command also requires a Vserver name.

Parameters

{ [-fields <fieldname>,…]

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

| [-instance] }

If you specify the `-instance` parameter, the command displays detailed information about all fields.

[-vserver <vserver name>] - Vserver Managing Volume (privilege: advanced)

This specifies the Vserver where the file resides.

[-path <text>] - Path to File (privilege: advanced)

This specifies the path to the file.

Examples

The following example displays the file handle information of a file named *`file1.txt`* in the volume *`/vol/vol1`*.

```
cluster1::> volume file show-filehandle -vserver vs0 -path
/vol/vol1/file1.txt
             Vserver                   Path
             ----------------------    ---------------------------
             vs0                        /vol/vol1/file1.txt
flags   snapid  fileid     generation  fsid        msid          dsid
             ------- ------  ---------  ----------  ----------
------------  ------------
             0x0     0       0x60       0x206b6     0x402       0x80000402
0x402
```

volume file show-inode

Display file paths for a given inode

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

This command displays information about all the files having a given inode in a volume of a Vserver. If the `-snapshot-id` or `-snapshot-name` parameter is specified, the command displays file information from the Snapshot copy; otherwise, it displays the information from the active file system. The `-vserver` , `-volume` and `-inode-number` are mandatory parameters.

If no optional parameter is specified, the command displays the following fields for all the files having the given inode:

- Vserver Name
- Volume Name

- Inode Number
- File Path

The `volume file show-inode` command is only supported on flexible volumes and FlexGroup constituents.

Parameters

{ [-fields <fieldname>,…]

If you specify the -fields parameter, the command output also includes the specified field or fields.

| [-snapshot]

If this parameter is specified, the command displays the following information:

- Vserver Name
- Volume Name
- Inode Number
- Snapshot Name
- Snapshot ID
- File Path

| [-instance] }

If this parameter is specified, the command displays detailed information about the files matching the specified inode number. The following information is displayed:

- Vserver Name
- Volume Name
- Inode Number
- File Path
- Snapshot Name
- Snapshot ID
- File Name
- Parent Inode Number
- Parent Directory Cookie

-vserver <vserver name> - Vserver Name

This specifies the Vserver in which the volume or Snapshot copy is located.

-volume <volume name> - Volume Name

This specifies the volume in which the inode number is located.

-inode-number <integer> - Inode Number

This specifies the inode number whose information has to be retrieved.

{ [-snapshot-name <snapshot name>] - Snapshot Name

If this parameter or `-snapshot-id` is specified, information about the files is retrieved from the Snapshot copy instead of the active file system.

| [-snapshot-id <integer>] - Physical Snapshot ID }

If this parameter or `-snapshot-name` is specified, information about the files is retrieved from the Snapshot copy instead of the active file system.

[-file-path <text>] - File Path

If this parameter is specified, the command displays information only about the files that match the specified file path.

[-file-name <text>] - File Name

If this parameter is specified, the command displays information only about the files that match the specified file name.

[-parent-inode-number <integer>] - Parent Inode Number

The inode number of the parent directory of the file associated with the inode. If this parameter is specified, the command displays information only about the files that match the specified parent inode number.

[-parent-dir-cookie <integer>] - Parent Directory Cookie

The index of the directory entry of the file in its parent directory tree. If this parameter is specified, the command displays information only about the files that match the specified parent directory cookie.

Examples

The following example displays information about all the files having the inode number 96 in the active file system of a volume named vol1 on a Vserver named vs1:

```
cluster1::> volume file show-inode -vserver vs1 -volume vol1 -inode-number
96
                              Inode
        Vserver   Volume      Number   File Path
        --------- ----------- -------- ---------------------------------
        vs1       vol1        96       /vol/vol1/file1
        vs1       vol1        96       /vol/vol1/file2
        vs1       vol1        96       /vol/vol1/A/file2
        3 entries were displayed.
```

The following example displays information about all the files with inode number 96 in a Snapshot copy named mysnap. The Snapshot copy is present in a volume named vol1 on a Vserver named vs1:

```
cluster1::> volume file show-inode -vserver vs1 -volume vol1 -inode-number
96 -snapshot-name mysnap -snapshot
                              Inode    Snapshot Snapshot
        Vserver   Volume      Number   Name     ID        File Path
        --------- ----------- -------- -------- --------
-------------------------
        vs1       vol1        96       mysnap   131
/vol/vol1/.snapshot/mysnap/file1
        vs1       vol1        96       mysnap   131
/vol/vol1/.snapshot/mysnap/file2
        2 entries were displayed.
```

The following example displays detailed information about all the files with inode number 96 in a Snapshot copy named mysnap. The Snapshot copy is present in a volume named vol1 on a Vserver named vs1:

```
cluster1::> volume file show-inode -vserver vs1 -volume vol1 -inode-number
96 -snapshot-name mysnap -instance
Vserver Name: vs1
                 Volume Name: vol1
                Inode number: 96
                   File Path: /vol/vol1/.snapshot/mysnap/file1
               Snapshot Name: mysnap
        Physical Snapshot ID: 131
                   File Name: file1
         Parent Inode Number: 64
     Parent Directory Cookie: 2
Vserver Name: vs1
                 Volume Name: vol1
                Inode number: 96
                   File Path: /vol/vol1/.snapshot/mysnap/file2
               Snapshot Name: mysnap
        Physical Snapshot ID: 131
                   File Name: file2
         Parent Inode Number: 64
     Parent Directory Cookie: 3
        2 entries were displayed.
```

volume file clone autodelete

Enable/Disable autodelete

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

The `volume file clone autodelete` command enables or disables the automatic deletion of file, LUN or NVMe namespace clones. Newly created file, LUN and NVMe namespace clones are disabled for automatic deletion by default.

Parameters

`-vserver <vserver name>` - Vserver Name

This specifies the Vserver on which the volume resides. If only one data Vserver exists, you do not need to specify this parameter.

`[-volume <volume name>]` - Volume Name

This specifies the name of the volume in which the file, LUN or NVMe namespace is present.

`-clone-path <text>` - Clone Path

This specifies the path where clone resides. If you use the volume parameter, then specify the relative path to the file, LUN or NVMe namespace clone. Otherwise, specify the absolute path.

`-enable {true|false}` - Enable or Disable Autodelete

This parameter enables or disables the autodelete feature for the file, LUN or NVMe namespace clones in a specified volume if the clones are already added for automatic deletion. If you set the parameter to true, the specified file, LUN or NVMe namespace clones get automatically deleted in the 'try' or 'disrupt' mode. If the value is false, the clones get automatically deleted only in the 'destroy' mode.

`[-force <true>]` - Force Enable or Disable Autodelete

If `-enable` is *`true`* then this parameter forces automatic deletion of a specified file, LUN or NVMe namespace, or a file, LUN or NVMe namespace clone. If `-enable` is *`false`* then specifying this parameter disables autodeletion on a file, LUN or NVMe namespace - or a file, LUN or NVMe namespace clone - even if `-commitment` *`destroy`* is specified.

Examples

The following command enables for automatic deletion a LUN Clone named *`lun_clone`* contained in a volume named *`volume1`* . This volume is present on a Vserver named *`vs1`* .

```
cluster1::> volume file clone autodelete /vol/volume1/lun_clone -enable
true -vserver vs1
```

The following command specifies the relative clone path when the volume parameter is specified in the command.

```
cluster1::> volume file clone autodelete lun_clone -enable true
      -vserver vs1 -volume volume1
```

volume file clone create

Create file or LUN full or sub file clone

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

The `volume file clone create` command creates a clone of a file, a LUN or an NVMe namespace. You can optionally specify the following parameters for the clone file creation process:

- Vserver in which the volume resides
- Name of the parent snapshot
- The range of blocks to be cloned
- The option to avoid space reservations for the new file or LUN clone
- The option to assign a QoS policy group to the new file or LUN clone
- The option to assign a caching policy to the new file or LUN clone
- The option to mark the new file, LUN or NVMe namespace clone created for auto deletion
- The option to overwrite an existing file, LUN or NVMe namespace clone

File, LUN or NVMe namespace clones create a duplicate copy of another file, LUN or NVMe namespace, but don’t require copying the data itself. This allows the clone operation to occur in constant time, taking the same amount of time to complete no matter the size of the file being cloned. This also means that clones require only a small amount of additional storage space because the clone shares the data with the source file, LUN or NVMe namespace.

Parameters

`-vserver <vserver name>` - Vserver Name

This specifies the Vserver in which the parent volume resides. If only one data Vserver exists, you do not need to specify this parameter.

`[-volume <volume name>]` - Volume

This specifies the name of volume in which a file, LUN or NVMe namespace is going to be cloned.

`-source-path <text>` - Source Path

This specifies the path to the file, LUN or NVMe namespace to be cloned relative to the specified volume.

`-destination-path <text>` - Destination Path

This specifies the path for the newly-created cloned file, LUN or NVMe namespace relative to the specified volume. If the file, LUN or NVMe namespace clone to be created is a whole file, LUN or NVMe namespace, the destination file, LUN or NVMe namespace must not exist. If the `range` parameter is specified, the destination file or LUN must exist. If the `snapshot-name` parameter is specified, this option is mandatory.

`[-s, -snapshot-name <snapshot name>]` - Source Snapshot

The name of the Snapshot copy to use as the source for the clone operation. If this value is not specified, the active filesystem will be used instead.

{ [-r, -range [source start block>:<destination start block>:<block length],…] - Block Range

This specifies the block range to be cloned. If the range is not specified, the entire file, LUN, or NVMe namespace is cloned. The block range should be specified in the format s:d:n where s is the source start block number, d is the destination start block number, and n is the length in blocks to be cloned. The range of n should be from 1 to 32768 or 1 to 16777216 in case of clone from Active File System or Snapshot copy respectively. If this parameter is used in the path provided by the `destination-path` , the parameter must refer to a file, LUN, or NVMe namespace which already exists. If either the source or destination is a LUN or NVMe namespace, then the block size is measured in LBA blocks. The source object block size and destination block size must be equal. If neither the source nor destination is a LUN or NVMe namespace, then the block size will be 4KB. If 512-byte sectors are used, the source and destination offsets must have the same offset within 4KB blocks. + This option is most likely to be used by external automated systems in managing virtual disk configurations and not by human administrators.

| [-o, -no-reserve <true>] - Do not reserve clone }

If this option is used, the clone file or LUN will not be guaranteed space in the underlying aggregate. While this out-of-space condition persists, writes to the clone file or LUN would fail. This option may be useful if few writes to the clone are expected to be needed, or to allow a file or LUN clone to be created under space-constrained conditions for recovery purposes. If this option is not specified the clone will inherit the space reservation properties from the source.

[-i, -ignore-streams <true>] - Ignore streams

This parameter specifies whether streams should be ignored during cloning of files, LUNs, or NVMe namespaces. If you set this parameter to false, the streams are ignored; otherwise, they are included in the clones. The default value is false.

[-k, -ignore-locks <true>] - Ignore locks

This parameter specifies whether byte-range locks and shared-mode locks on files, LUNs or NVMe namespaces should be ignored during cloning. If you set this parameter to true, the locks are ignored; otherwise, clone operation fails if locks are present on files, LUNs or NVMe namespaces. The default value is false.

[-d, -overwrite-destination <true>] - Overwrite Destination

Specify this parameter to overwrite the destination file, LUN, or NVMe namespace, if it exists. The default is not to overwrite the destination file. The command will fail if the destination file exists.

{ [-qos-policy-group <text>] - QoS Policy Group Name

This optionally specifies which QoS policy group to apply to the file or LUN. This policy group defines measurable service level objectives (SLOs) that apply to the storage objects with which the policy group is associated. If you do not assign a policy group to a file or LUN, the system will not monitor and control the traffic to it. You cannot associate a file to a QoS policy group if a LUN was created from the file.

| [-qos-adaptive-policy-group <text>] - QoS Adaptive Policy Group Name }

This optionally specifies which QoS adaptive policy group to apply to the file or LUN. This policy group defines measurable service level objectives (SLOs) and Service Level Agreements (SLAs) that adjust based on the file or LUN’s allocated space or used space.

[-caching-policy <text>] - Caching Policy Name

This optionally specifies the caching policy to apply to the file. A caching policy defines how the system caches this volume’s data in Flash Cache modules. If a caching policy is not assigned to this file, the system uses the caching policy that is assigned to the containing volume. If a caching policy is not assigned to the containing volume, the system uses the caching policy that is assigned to the containing Vserver. If a

caching policy is not assigned to the containing Vserver, the system uses the default cluster-wide policy. The available caching policies are:

- none - Does not cache any user data or metadata blocks.
- auto - Read caches all metadata and randomly read user data blocks, and write caches all randomly overwritten user data blocks.

Default caching-policy is auto.

[-autodelete {true|false}] - Mark Clone for Autodeletion

This parameter marks the file, LUN or NVMe namespace clones created for auto deletion. When set to true, the file, LUN or NVMe namespace clones get automatically deleted when the volume runs out of space. The default value is false.

[-bypass-throttle {true|false}] - Bypass Throttle Checks (privilege: advanced)

This parameter specifies whether clone throttle checks should be skipped during clone creation. When set to true, clones are created without enforcing any clone throttle checks. The default value is false.

[-is-backup {true|false}] - Is a Clone for Backup

This parameter is used to mark the destination file as a backup clone, where divergence is expected on the source file and no divergence is expected on the destination file. It is applicable only for full-file clones created from Active File System volumes. The default value is *false* .

[-destination-volume <volume name>] - Destination Volume

This specifies the name of the volume where the destination file resides. This can be different from `volume` , whereas parameter `volume` specifies the volume on which source file resides. This is an optional argument that applies only to a MetaWAFL volume where the source and destination volumes for the clone operation can be different. If this parameter is not given, the destination file will be created in the volume where source_file resides.

Examples

The following command creates a FlexClone file of the file named *myfile* contained in a volume named *vol* . The file *myfile* is located in the root directory of that volume. The cloned file *myfile_copy* resides in the root directory same volume.

```
cluster1::> volume file clone create -volume vol -source-path /myfile
-destination-path /myfile_copy
```

The following command optionally associates the FlexClone file named *myfile_copy* with the *fast* QoS policy group and the caching policy named random-read.

```
cluster1::> volume file clone create -volume vol -source-path /myfile
-destination-path /myfile_copy -qos-policy-group fast -caching-policy
random-read
```

volume file clone show-autodelete

Show the autodelete status for a file or LUN clone

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

The `volume file clone show-autodelete` command displays the autodelete details of a file, LUN or NVMe namespace clone. The command displays the following information about a file, LUN or NVMe namespace clone:

- Vserver Name
- Clone Path
- Whether auto deletion of file, LUN or NVMe namespace clone is enabled

Parameters

{ [-fields <fieldname>,…]

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

| [-instance] }

If you specify the `-instance` parameter, the command displays detailed information about all fields.

-vserver <vserver name> - Vserver Name

This specifies the Vserver to which the file, LUN or NVMe namespace clone belongs.

-clone-path <text> - Clone Path

This specifies the path of the file, LUN or NVMe namespace clone.

[-autodelete-enabled {true|false}] - Autodelete Enabled

If this parameter is true, the file, LUN or NVMe namespace clone gets automatically deleted in the 'try' or 'disrupt' mode. If the value is false, the clones get automatically deleted only in the 'destroy' mode.

Examples

The following example displays the autodelete information about a file, LUN or NVMe namespace clone.

```
cluster1::> volume file clone show-autodelete -vserver vs1 -clone-path
/vol/v1/f1
Vserver Name: vs1
        Clone Path: /vol/v1/f1
Autodelete Enabled: true
```

volume file clone deletion add-extension

Add new supported file extensions to be deleted with clone delete

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

The `volume file clone deletion add-extension` command can be used to add new supported file extensions for clone delete.

Parameters

`-vserver <vserver name>` - Vserver Name

Name of the vserver.

`-volume <volume name>` - Volume Name

Name of the volume.

`-extensions <text>` - Supported Extensions for Clone Delete

List of supported file extensions for clone delete.

Examples

The following example adds the new supported *`vmdk,vhd`* file extensions to volume *`vol1`* of vserver *`vs1`* .

```
cluster1::> volume file clone deletion add-extension -vserver vs1 -volume
vol1 -extensions vmdk,vhd
```

volume file clone deletion modify

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

The `volume file clone deletion modify` command can be used to change the required minimum clone file size of a volume for clone delete.

Parameters

`-vserver <vserver name>` - Vserver Name

Name of the vserver.

`-volume <volume name>` - Volume Name

Name of the volume.

[-minimum-size {<integer>[KB|MB|GB|TB|PB]}] - Minimum Size Required for Clone delete

Minimum clone file size required for clone delete.

Examples

The following example changes the required minimum file size to 100M for volume *vol1* of vserver _ vs1 _ .

```
cluster1::> volume file clone deletion modify -volume vol1 -vserver vs1
-minimum-size 100M
```

volume file clone deletion remove-extension

Remove unsupported file extensions for clone delete

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

The `volume file clone deletion remove-extension ` command can be used to remove the existing file extensions that are no longer supported for clone delete.

Parameters

-vserver <vserver name> - Vserver Name

Name of the vserver.

-volume <volume name> - Volume Name

Name of the volume.

[-extensions <text>] - Unsupported Extensions for Clone Delete

List of unsupported file extensions for clone delete.

Examples

The following example removes the existing unsupported *vmdk, vhd* file extensions to volume *vol1* of vserver *vs1* .

```
cluster1::> volume file clone deletion remove-extension -vserver vs1
-volume vol1 -extensions vmdk,vhd
```

volume file clone deletion show

Show the supported file extensions for clone delete

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

The `volume file clone deletion show` command displays the following information for clone delete:

- Vserver Name
- Volume Name
- Minimum File Size Required for Clone Delete
- List of Supported File Extensions for Clone Delete

Parameters

{ [-fields <fieldname>,…]

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

| [-instance] }

If you specify the `-instance` parameter, the command displays detailed information about all fields.

[-vserver <vserver name>] - Vserver Name

Name of the vserver.

[-volume <volume name>] - Volume Name

Name of the volume.

[-extensions <text>,…] - Supported Extensions for Clone Delete

List of supported file extensions for Clone Delete.

[-minimum-size {<integer>[KB|MB|GB|TB|PB]}] - Minimum Size Required for Clone delete

Minimum file size required for Clone Delete.

Examples

The following example displays the clone deletion information for all volumes of all vservers.

```
cluster1::> volume file clone deletion show
        Vserver            Volume                Minimum    Extensions
                                                    Size
       -----------------   -------------------   -------
----------------------
       vs0                 testvol                 100B      vmdk, vhd,
vhdx, vswp
                           vs0_root                  0B      -
       vs1                 testvol                 100G      vmdk, vhd,
vhdx, vswp
                           vs1_root                  0B      -
```

The following example displays the clone deletion information for volume *vol1* of vserver *vs1* .

```
cluster1::> volume file clone deletion show -vserver vs0 -volume testvol
                          Vserver Name: vs0
                           Volume Name: testvol
 Supported Extensions for Clone Delete: vmdk, vhd, vhdx, vswp
Minimum Size Required for Clone delete: 100B
```

volume file clone split load modify

Modify maximum split load on a node

Availability: This command is available to *cluster* and *Vserver* administrators at the *advanced* privilege level.

Description

The `volume file clone split load modify` command can be used to change the maximum split load (file, LUN or NVMe namespace clones) of a node.

Parameters

`-node {<nodename>|local}` - Node Name

Node name on which the new maximum split load is being applied.

`[-max-split-load {<integer>[KB|MB|GB|TB|PB]}]` - Maximum Clone Split Load

This specifies the new maximum split load of a node. This is the amount of clone create load, the node can take at any point of time. If it crosses this limit, then the clone create requests will not be allowed, till the split load is less than maximum split load

Examples

The following example changes the new maximum limit to 10TB on node1.

```
cluster1::*> volume file clone split load*> modify -node clone-01 -max
-split-load 100KB
```

volume file clone split load show

Show split load on a node

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

The `volume file clone split load show` command displays the corresponding file, LUN or NVMe namespace clone split loads on nodes.If no parameters are specified, the command displays the following information:

- Node
- Max Split Load
- Current Split Load
- Token Reserved Load
- Allowable Split Load

Parameters

{ [-fields <fieldname>,…]

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

| [-instance] }

If you specify the `-instance` parameter, the command displays detailed information about all fields.

[-node {<nodename>|local}] - Node Name

Node on which the file, LUN or NVMe namespace clone split load is displayed.

[-max-split-load {<integer>[KB|MB|GB|TB|PB]}] - Maximum Clone Split Load

This specifies the maximum allowable split load on the node.

[-current-split-load {<integer>[KB|MB|GB|TB|PB]}] - Current Clone Split Load

This specifies the current on going split load on the node.

[-token-reserved-load {<integer>[KB|MB|GB|TB|PB]}] - Load Reserved for Clone Creation

This specifies the reserved split load of the node using the tokens.

[-allowable-split-load {<integer>[KB|MB|GB|TB|PB]}] - Allowable Clone Split Load

This specifies the available split load of the node.

Examples

The following example displays the current and allowable file, LUN or NVMe namespace clone split load on a node.

```
cluster1::> volume file clone split load show
Node                            Max        Current    Token
Allowable
                                Split Load Split Load Reserved Load Split
Load
------------------------------ ---------- ---------- -------------
----------
clone-01                         15.97TB         0B         100MB
15.97TB
clone-02                         15.97TB         0B         100MB
15.97TB
2 entries were displayed.
cluster1::> volume file clone split load show -node clone-01 -instance
Node Name: clone-01
        Maximum Clone Split Load: 15.97TB
        Current Clone Split Load: 0B
Load Reserved for Clone Creation: 100MB
      Allowable Clone Split Load: 15.97TB
```

volume file fingerprint abort

Abort a file fingerprint operation

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

The `volume file fingerprint abort` command aborts an in-progress fingerprint operation. This command only aborts the fingerprint operations that have not yet completed. This command takes session-id as input and aborts the fingerprint operation that is associated with that particular session-id.

Parameters

`-session-id <integer>` - Session ID of Fingerprint Operation

Specifies the session-id of the fingerprint operation that needs to be aborted. It is an unique identifier for the fingerprint operation. This session-id is returned when the fingerprint operation is started on a file.

Examples

The following example aborts the fingerprint operation identified by *17039361* :

```
cluster1::> volume file fingerprint abort -session-id 17039361
```

volume file fingerprint dump

Display fingerprint of a file

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

The `volume file fingerprint dump` command displays the following information given the `-session -id` of the fingerprint operation:

- Vserver:

The Vserver on which the file exists.
* Session-ID:

A unique identifier for the fingerprint operation. This session-id is returned when the fingerprint operation is started on a file. The session-id of the fingerprint operation can be used to get the progress of an ongoing fingerprint operation as well as the complete fingerprint output for the file once the operation is completed.
* Volume:

The name of the volume on which the file resides.
* Path:

The absolute path of the file on which the fingerprint is calculated. The value begins with /vol/<volumename>.
* Data Fingerprint:

The digest value of data of the file. The fingerprint is base64 encoded. This field is not included if the scope is *`metadata-only`* .
* Metadata Fingerprint:

The digest value of metadata of the file. The metadata fingerprint is calculated for file size, file ctime, file mtime, file crtime, file retention time, file uid, file gid, and file type. The fingerprint is base64 encoded. This field is not included if the scope is *`data-only`* .
* Fingerprint Algorithm:

The digest algorithm which is used for the fingerprint computation. Fingerprint is computed using *`md5`* or *`sha-256`* digest algorithm.
* Fingerprint Scope:

The scope of the file which is used for the fingerprint computation. Fingerprint is computed over *`data-only`* , *`metadata-only`* , or *`data-and-metadata`* .
* Fingerprint Start Time:

The start time of the fingerprint computation in seconds since 1 January 1970 00:00:00 in GMT timezone.
* Formatted Fingerprint Start Time:

The start time of the fingerprint computation in a human-readable format <day> <month> <day of month> <hour>:<min>:<sec><year> in GMT timezone.
* Fingerprint Version:

The version of the fingerprint output format.
* SnapLock License:

The status of the SnapLock license.
* Vserver UUID:

A universal unique identifier for the Vserver on which the file exists.
* Volume MSID:

The mirror set identifier of the volume where the file resides.
* Volume DSID:

The data set identifier of the volume where the file resides.
* Hostname:

The name of the storage system where the fingerprint operation is performed.
* Filer ID:

The NVRAM identifier of the storage system.
* Volume Containing Aggregate:

The name of the aggregate in which the volume resides.
* Aggregate ID:

A universal unique identifier for the aggregate containing the volume.
* SnapLock System ComplianceClock:

The System ComplianceClock time in seconds since 1 January 1970 00:00:00 in GMT timezone if it is initialized.
* Formatted SnapLock System ComplianceClock:

The System ComplianceClock time in a human-readable format <day> <month> <day of month> <hour>:<min>:<sec><year> in GMT timezone if it is initialized.
* Volume SnapLock Type:

The type of the SnapLock volume. This value is only given for SnapLock volumes. Possible values are *`compliance`* and *`enterprise`*.
* Volume ComplianceClock:

The volume ComplianceClock time in seconds since 1 January 1970 00:00:00 in GMT timezone. This has a value only for SnapLock volumes.
* Formatted Volume ComplianceClock:

The volume ComplianceClock time in a human-readable format <day> <month> <day of month> <hour>:<min>:<sec><year> in GMT timezone. This has a value only for SnapLock volumes.
* Volume Expiry Date:

The expiry date of the SnapLock volume in seconds since 1 January 1970 00:00:00 in GMT timezone. The volume expiry date can be in wraparound format.
* Is Volume Expiry Date Wraparound:

The value is *`true`* if the volume expiry date is in wraparound format. The wraparound format indicates that dates after 19 January 2038 are mapped from 1 January 1970 through 31 December 2002 to 19 January 2038 through 19 January 2071.
* Formatted Volume Expiry Date:

The expiry date of the SnapLock volume in a human-readable format <day> <month> <day of month>

<hour>:<min>:<sec><year> in GMT timezone. The volume expiry date can be in wraparound format.
* Filesystem ID:

The filesystem identifier of the volume on which the file resides.
* File ID:

A unique number within the filesystem identifying the file.
* File Type:

The type of the file. Possible values include: *worm* , *worm_appendable* , *worm_active_log* , *worm_log* , and *regular* .
* File Size:

The size of the file in bytes.
* Creation Time:

The creation time of the file in seconds since 1 January 1970 00:00:00 in GMT timezone.
* Formatted Creation Time:

The creation time of the file in a human-readable format <day> <month> <day of month> <hour>:<min>:<sec><year> in GMT timezone.
* Modification Time:

The last modification time of the file in seconds since 1 January 1970 00:00:00 in GMT timezone.
* Formatted Modification Time:

The last modification time of the file in a human-readable format <day> <month> <day of month> <hour>:<min>:<sec><year> in GMT timezone.
* Changed Time:

The last changed time of the file attributes in seconds since 1 January 1970 00:00:00 in GMT timezone. Time is taken from the system clock for regular files and from the volume ComplianceClock for WORM files when they are committed. The changed time can be in wraparound format.
* Is Changed Time Wraparound:

The value is *true* if the last changed time of the file attributes is in wraparound format. The wraparound format indicates that dates after 19 January 2038 are mapped from 1 January 1970 through 31 December 2002 to 19 January 2038 through 19 January 2071.
* Formatted Changed Time:

The last changed time of the file attributes in a human-readable format <day> <month> <day of month> <hour>:<min>:<sec><year> in GMT timezone. The changed time can be in wraparound format.
* Retention Time:

The retention time of the files committed to WORM on SnapLock volumes in seconds since 1 January 1970 00:00:00 in GMT timezone. The retention time can be in wraparound format.
* Is Retention Time Wraparound:

The value is *true* if the retention time of the file is in wraparound format. The wraparound format indicates that dates after 19 January 2038 are mapped from 1 January 1970 through 31 December 2002 to 19 January 2038 through 19 January 2071.
* Formatted Retention Time:

The retention time of the files protected by SnapLock in a human-readable format <day> <month> <day of month> <hour>:<min>:<sec><year> in GMT timezone. The retention time can be in wraparound format.

* Access Time:

The last access time of the regular files on SnapLock volumes and files on non-SnapLock volumes attributes in seconds since 1 January 1970 00:00:00 in GMT timezone.
* Formatted Access Time:

The last access time of the regular files on SnapLock volumes and files on non-SnapLock volumes attributes in a human-readable format <day> <month> <day of month> <hour>:<min>:<sec><year> in GMT timezone.
* Owner ID:

The integer identifier of the owner of the file.
* Group ID:

The integer identifier of the group owning the file.
* Owner SID:

The security identifier of the owner of the file when it has NTFS security style.
* Fingerprint End Time:

The end time of the fingerprint computation in seconds since 1 January 1970 00:00:00 in GMT timezone.
* Formatted Fingerprint End Time:

The end time of the fingerprint computation in a human-readable format <day> <month> <day of month> <hour>:<min>:<sec><year> in GMT timezone.
* Litigation Count:

The number of litigations on the file.

Parameters

{ [-fields <fieldname>,…]

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

| [-instance] }

If you specify the `-instance` parameter, the command displays detailed information about all fields.

-session-id <integer> - Session ID of Fingerprint Operation

Specifies the session-id of the fingerprint operation whose output is to be displayed. It is a unique identifier for the fingerprint operation. This session-id is returned when the fingerprint operation is started on a file.

Examples

The following example displays the fingerprint information of the fingerprint session identified by session-id *17039367*:

```
cluster1::> volume file fingerprint dump -session-id 17039367
Vserver:vs1
                                Session-ID:17039367
                                    Volume:nfs_slc
                                      Path:/vol/nfs_slc/worm
```

```
                          Data
Fingerprint:MOFJVevxNSJm3C/4Bn5oEEYH51CrudOzZYK4r5Cfy1g=
                      Metadata
Fingerprint:8iMjqJXiNcqgXT5XuRhLiEwIrJEihDmwS0hrexnjgmc=
                    Fingerprint Algorithm:SHA256
                        Fingerprint Scope:data-and-metadata
                   Fingerprint Start Time:1460612586
         Formatted Fingerprint Start Time:Thu Apr 14 05:43:06 GMT 2016
                      Fingerprint Version:3
                         SnapLock License:available
                             Vserver UUID:acf7ae64-00d6-11e6-a027-
0050569c55ae
                              Volume MSID:2152884007
                              Volume DSID:1028
                                 Hostname:cluster1
                                 Filer ID:5f18eda2-00b0-11e6-914e-
6fb45e537b8d
               Volume Containing Aggregate:slc_aggr
                             Aggregate ID:c84634aa-c757-4b98-8f07-
eefe32565f67
           SnapLock System ComplianceClock:1460610635
 Formatted SnapLock System ComplianceClock:Thu Apr 14 05:10:35 GMT 2016
                     Volume SnapLock Type:compliance
                   Volume ComplianceClock:1460610635
          Formatted Volume ComplianceClock:Thu Apr 14 05:10:35 GMT 2016
                      Volume Expiry Date:1465880998
          Is Volume Expiry Date Wraparound:false
              Formatted Volume Expiry Date:Tue Jun 14 05:09:58 GMT 2016
                           Filesystem ID:1028
                                 File ID:96
                                File Type:worm
                                File Size:1048576
                           Creation Time:1460612515
                  Formatted Creation Time:Thu Apr 14 05:41:55 GMT 2016
                        Modification Time:1460612515
               Formatted Modification Time:Thu Apr 14 05:41:55 GMT 2016
                            Changed Time:1460610598
                Is Changed Time Wraparound:false
                    Formatted Changed Time:Thu Apr 14 05:09:58 GMT 2016
                          Retention Time:1465880998
              Is Retention Time Wraparound:false
                  Formatted Retention Time:Tue Jun 14 05:09:58 GMT 2016
                             Access Time:-
                   Formatted Access Time:-
                                Owner ID:0
                                Group ID:0
```

```
                              Owner SID:-
                    Fingerprint End Time:1460612586
          Formatted Fingerprint End Time:Thu Apr 14 05:43:06 GMT 2016
                         Litigation Count:0
```

volume file fingerprint show

Display fingerprint operation status

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

The `volume file fingerprint show` command returns information for one or several fingerprint operations. This command requires either `-session-id` or `-vserver` and `-volume` .

Parameters

{ [-fields <fieldname>,…]

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

| [-instance] }

If you specify the `-instance` parameter, the command displays detailed information about all fields.

[-session-id <integer>] - Session ID of Fingerprint Operation

If this parameter is specified, the command returns the progress of the fingerprint operation of the specified session-id. The session-id is a unique identifier for the fingerprint operation that is returned when the fingerprint operation is started on a file.

[-vserver <vserver name>] - Vserver

If this parameter is specified, `-volume` must also be specified. When queried with `-vserver` and `-volume` , the command returns the progress of all the fingerprint operations running on that particular volume.

[-volume <volume name>] - Volume Name

If this parameter is specified, `-vserver` must also be specified. When queried with `-vserver` and `-volume` , the command returns the progress of all the fingerprint operations running on that particular volume.

[-file </vol/<volume name>/<file path>>] - File Path

If this parameter is specified, the command returns the progress of all fingerprint operations on the specified file.

[-operation-status {Unknown|In-Progress|Failed|Aborting|Completed}] - Operation Status

If this parameter is specified, the command returns the progress of all fingerprint operations with matching status value.

[-progress-percentage <integer>] - Progress Percentage

If this parameter is specified, the command returns the progress of all fingerprint operations with matching progress percentage value.

Examples

The following example displays the progress of all the fingerprint operations running on volume *nfs_slc* :

```
cluster1::> volume file fingerprint show -vserver vs0 -volume nfs_slc

Progress
File-Path                                Session-ID         Status
Percentage
---------------------------------------- ------------------ ---------
----------
/vol/nfs_slc/worm                        17104897           Completed
100
/vol/nfs_slc/worm_appedable              17104898           Completed
100
/vol/nfs_slc/regular                     17104899           In-Progress
30
3 entries were displayed.
```

volume file fingerprint start

Start a file fingerprint computation on a file

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

The `volume file fingerprint start` command starts the fingerprint computation on a file. The fingerprint computation is started on the file, and a session-id is returned. This session-id is an unique identifier for the fingerprint operation and can be used to get the progress of an ongoing fingerprint operation as well as the complete fingerprint output for the file once the operation is completed.

Parameters

-vserver <vserver name> - Vserver

Specifies the name of the vserver which owns the volume on which the file resides.

-file </vol/<volume name>/<file path>> - Path

Specifies the absolute path of the file on which fingerprint needs to be calculated. The value begins with /vol/<volumename>.

[-algorithm {MD5|SHA256}] - Fingerprint Algorithm

Specifies the digest algorithm which is used for the fingerprint computation.

Fingerprint can be computed using one of the following digest algorithms:

- md5
- sha-256

[-scope {data-and-metadata|data-only|metadata-only}] - Fingerprint Scope

Specifies the scope of the file which is used for the fingerprint computation.

Fingerprint can be computed using one of the following scope:

- data-only
- metadata-only
- data-and-metadata

Examples

The following example starts computing fingerprint over data and metadata for file */vol/nfs_slc/worm* using *md5* hash algorithm. The file */vol/nfs_slc/worm* is stored in volume *nfs_slc* on Vserver *vs0* .

```
cluster1::> volume file fingerprint start -vserver vs0 -scope data-and-
metadata -algorithm md5 -file /vol/nfs_slc/worm
File fingerprint operation is queued. Run "volume file fingerprint show
-session-id 16973825" to view the fingerprint session status.
```

volume file retention show

Display retention time of a file protected by SnapLock.

Availability: This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

Description

The `volume file retention show` command displays the retention time of a file protected by SnapLock given `-vserver` and `-file` .

Parameters

{ [-fields <fieldname>,…]

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

| [-instance] }

If you specify the `-instance` parameter, the command displays detailed information about all fields.

[-vserver <vserver name>] - Vserver Name

Specifies the name of the Vserver which has the file.

[-file </vol/<volume name>/<file path>>] - File path

Specifies the absolute path of the file. The value begins with /vol/<volumename>.

[-retention-time <integer>] - Retention Time of the File

If this paramater is specified, the command returns the retention time of the file protected by SnapLock if its retention time in seconds since 1 January 1970 00:00:00 matches the specified value.

[-formatted-retention-time <text>] - Formatted Retention Period

If this paramater is specified, the command returns the retention time of the file protected by SnapLock if its expiry date matches the specified value. The expiry date format is <day> <month> <day of month> <hour>:<min>:<sec><year> in GMT timezone taking care of wraparound. A value of *infinite* indicates that this file has infinite retention time. A value of *indefinite* indicates that this file has indefinite retention time.

[-is-wraparound {true|false}] - Is Retention Time Wraparound

If this paramater is specified, the command returns the retention time of the file protected by SnapLock if it has a matching `-is-wraparound` value. The value is `true` if the date represented in retention time is in wraparound format. The wraparound format indicates that dates after 19 January 2038 are mapped from 1 January 1970 through 31 December 2002 to 19 January 2038 through 19 January 2071.

Examples

The following example displays the retention time of the file */vol/nfs_sle/f12* :

```
cluster1::> volume file retention show -vserver vs0 -file /vol/nfs_sle/f12
Vserver : vs0
                                    Path : /vol/nfs_sle/f12
          Retention Time (Secs from Epoch) : 1439111404
                  Formatted Retention Time : Sun Aug  9 09:10:04 GMT 2015
              Is Retention Time Wraparound : false
```

Copyright information

Copyright © 2026 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.