5 Using Exascale Block Storage with EDV

This topic outlines the procedure for using Exascale block storage with Exascale Direct Volumes (EDV).

The Exascale block store provides capabilities to create and manage arbitrary-sized raw block volumes based on Exascale storage.

EDV is recommended for use-cases where clients want to use storage volumes inside the Exadata RDMA Network Fabric. For example:

- You can use EDV to support Oracle Advanced Cluster File System (ACFS) or local Linux file systems (such as XFS, EXT4, and so on) on the Exadata compute nodes.
- You can use EDV raw devices to support earlier Oracle Database versions, not natively supported on Exascale. For example, you can create an Oracle ASM disk group based on EDV raw devices and use the disk group to contain the Oracle Database data files. Or, you can create a file system on an EDV volume and use it to contain the Oracle Database software and data files.

Before you can use Exascale block storage with EDV, note the following:

- The block store manager (BSM) and block store worker (BSW) services must be running in the Exascale cluster.

  To display information about Exascale storage services running across the Exascale cluster, use the ESCLI lsservice command. For example:

  ```
  @> lsservice --detail
  ```

- The Exascale Direct Volume (EDV) service must be running on each Exadata compute node that you want to host an EDV attachment. If you are planning to use a cluster-wide attachment, then the EDV service must be running on every node in the Oracle Grid Infrastructure (GI) cluster.

  To display information about client-side Exascale services, including Exascale Node Proxy (ESNP) and Exascale Direct Volume (EDV), use the DBMCLI LIST DBSERVER command, as follows:

  ```
  DBMCLI> list dbserver detail
  ```

  The command output displays status information for all of the client-side Exascale services running on the Exadata compute node. You must run the command separately on each compute node.

- During initial system deployment with Oracle Exadata Deployment Assistant (OEDA), the Exascale Direct Volume (EDV) service is configured on each Exadata compute node (bare-metal or VM) and runs with the permissions of the Exascale user that manages the Oracle Grid Infrastructure (GI) cluster. To create an EDV attachment, you must use the Exascale user linked with the EDV service.

  If the GI cluster uses a non-role-separated user configuration with one Oracle OS user account, then the associated Exascale user is related to the EDV service. If the GI cluster uses a role-separated configuration with a Grid OS user account and an Oracle OS user account, then the EDV service is linked to the Exascale user associated with the Grid OS account.

To find the Exascale user linked with the EDV service, use the ESCLI lsinitiator command with the --detail option and examine the user attribute.

To begin, you can create an Exascale block volume and EDV attachment.

For example:

```
@>mkvault my-vault
Vault @my-vault created.

@>mkvolume 1g --attributes name=edv1 --vault my-vault
Created volume with id vol0002_7eb2b5cc5d1a47f09abed0fa83514a36

@>lsvolume vol0002_7eb2b5cc5d1a47f09abed0fa83514a36 --detail
id                         vol0002_7eb2b5cc5d1a47f09abed0fa83514a36
name                         edv1
bandwidthProvisioned                unlimited
contentType                   DATA
creationTime                   2025-06-06T05:06:46+00:00
filePath                     @my-vault/vol.e2173df2021a4368bbf986483f8332b8
iopsProvisioned                  unlimited
mediaType                     HC
numAttachments                    0
owners                        exa01
redundancy                      high
size                        1G
state                       AVAILABLE
vault                        @my-vault

@>lsinitiator --detail
id                         e7e0db8c-9a2a-0279-e7e0-db8c9a2a0279
edvBaseDriverVersion                25.2.0.0.0.250602
edvEffectiveDriverVersion            25.2.0.0.0.250602
giClusterId                   deadbeef-badc-0fee-dead-beefbadc0fee
giClusterName                   edvTestCluster
hostName                      exa01
lastHeartbeat                   2025-06-06T05:08:20+00:00
registerTime                   2025-06-03T07:41:10+00:00
state                       ONLINE
user                        exa01
version                       25.2.0.0.0.250602

@>mkvolumeattachment vol0002_7eb2b5cc5d1a47f09abed0fa83514a36 my-edv1 --attributes
giClusterId=deadbeef-badc-0fee-dead-beefbadc0fee
Created edv attachment with id att0001_9deed3d3f7f944838f5767151d9f06de

@>lsvolumeattachment att0001_9deed3d3f7f944838f5767151d9f06de --detail
id                         att0001_9deed3d3f7f944838f5767151d9f06de
attachTime                   2025-06-06T05:11:20+00:00
deviceName                     my-edv1
devicePath                    /dev/exc/my-edv1
giClusterId                   deadbeef-badc-0fee-dead-beefbadc0fee
giClusterName                   edvTestCluster
hostName
```

```
initiator
kernelDeviceName                exc-dev1
logicalSectorSize             512
volume                      vol0002_7eb2b5cc5d1a47f09abed0fa83514a36
volumeSnapshot

@>
```

The EDV attachment creates an association between the volume and an EDV device file, which resides on the Exadata compute nodes hosting the attachment. A node hosting an EDV attachment is also known as an EDV initiator. If you create a cluster-wide attachment, then the EDV device file is created on every node in the Oracle Grid Infrastructure (GI) cluster. If you create a node-specific attachment, then the corresponding EDV device is only created on that node.

In the example, the EDV attachment is a cluster-wide attachment, and the EDV device name is `my-edv1`, so the corresponding device file is located at `/dev/exc/my-edv1` on each cluster node. The volume identifier (`vol0002_7eb2b5cc5d1a47f09abed0fa83514a36`) was reported to the user during volume creation. Volume identifiers can also be discovered using the ESCLI lsvolume command. The GI cluster identifier (`deadbeef-badc-0fee-dead-beefbadc0fee`) was found by using the ESCLI lsinitiator command.

Alternatively, for a node-specific attachment you must specify the node-specific EDV initiator identifier instead of the GI cluster identifier. For example:

```
@>lsinitiator --detail
id                          e7e0db8c-9a2a-0279-e7e0-db8c9a2a0279
edvBaseDriverVersion                25.2.0.0.0.250602
edvEffectiveDriverVersion          25.2.0.0.0.250602
giClusterId                    deadbeef-badc-0fee-dead-beefbadc0fee
giClusterName                    edvTestCluster
hostName                      exa01
lastHeartbeat                 2025-06-06T05:21:18+00:00
registerTime                  2025-06-03T07:41:10+00:00
state                     ONLINE
user                      exa01
version                     25.2.0.0.0.250602

@>mkvolumeattachment vol0002_7eb2b5cc5d1a47f09abed0fa83514a36 my-edv1 --attributes
initiator=e7e0db8c-9a2a-0279-e7e0-db8c9a2a0279
Created edv attachment with id att0001_4118794cebae49d6836139b34df08b6b

@>lsvolumeattachment att0001_4118794cebae49d6836139b34df08b6b --detail
id                          att0001_4118794cebae49d6836139b34df08b6b
attachTime                    2025-06-06T05:22:47+00:00
deviceName                     my-edv1
devicePath                    /dev/exc/my-edv1
giClusterId
giClusterName
hostName                      exa01
initiator                   e7e0db8c-9a2a-0279-e7e0-db8c9a2a0279
kernelDeviceName                exc-dev1
logicalSectorSize             512
volume                      vol0002_7eb2b5cc5d1a47f09abed0fa83514a36
volumeSnapshot
```

```
@>
```

After attachment, the EDV device can be used on the Exadata compute node or GI cluster hosting the attachment.

For example, you could use the following command sequence to create and mount an ACFS file system using the EDV device defined in the previous examples (/dev/exc/my-edv1):

```
# # Confirmation of the EDV device at /dev/exc/my-edv1.
# lsblk /dev/exc/my-edv1
NAME     MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
exc-dev1 251:1    0   1G  0 disk
# # Note the default ownership and permissions for the EDV device.
# ls -l /dev/exc/my-edv1
brw-rw---- 2 root disk 251, 1 Jun  6 05:11 /dev/exc/my-edv1
# # Now use the device to support ACFS.
# mkfs -t acfs /dev/exc/my-edv1
mkfs.acfs: version                   = 23.0.0.0.0
mkfs.acfs: ACFS compatibility        = 23.0.0.0.0
mkfs.acfs: on-disk version           = 53.0
mkfs.acfs: volume                    = /dev/exc/my-edv1
mkfs.acfs: volume size               = 1073741824  (   1.00 GB )
mkfs.acfs: file system size          = 1073741824  (   1.00 GB )
mkfs.acfs: Format complete.
# mkdir /mnt/my-acfs1
# mount /dev/exc/my-edv1 /mnt/my-acfs1
# df /mnt/my-acfs1
Filesystem       1K-blocks   Used Available Use% Mounted on
/dev/exc/my-edv1   1048576 318180    730396  31% /mnt/my-acfs1
#
# # Mount the file system on other cluster nodes as required.
```

> **Note**
>
> - Each EDV attachment also has a kernel device file at /dev/exc-dev*N*, where *N* is the minor number of the device. The kernel device name is contained as an attribute of the EDV attachment and is visible using the ESCLI lsvolumeattachment command.
>
>   Note that most Linux tools, such as iostat, display the kernel device file at /dev/exc-dev*N*, while Exascale commands use the user-named device file (under /dev/exc/).
>
>   The relationship between the kernel device file and the user-named device file is also recorded in the udev database and is visible using the following Linux command:
>
>   ```
>   # udevadm info device-file
>   ```
>
>   In the udevadm command, for the *device-file* value, you can specify either the kernel device file (/dev/exc-dev*N*) or the user-named device file (under /dev/exc/).
>
> - By default, read and write access to EDV device files is only available to the root operating system user and members of the disk group. Depending on your use case, you may need to modify the permissions on the EDV device files before using them.
>
>   For example, to make the EDV device file at /dev/exc/my-vol readable and writable by the oracle user and dba group, you could configure it using a udev rule similar to the following:
>
>   ```
>   # cat /etc/udev/rules.d/57-edv-user.rules
>   KERNEL=="exc-*", ENV{EXC_ALIAS}=="my-vol", OWNER="oracle", GROUP="dba", MODE="0660"
>   ```
>
> - To facilitate the management of udev rules related to EDV devices, each EDV client node is configured with a template udev rules file at /etc/udev/rules.d/57-edv-user.rules, which you can modify to fulfill your requirements. To maintain existing udev rules, /etc/udev/rules.d/57-edv-user.rules is preserved whenever the EDV client software is updated.
>
> - Each EDV client node can support a maximum of 3000 attachments at the same time. This limit includes the total of all cluster attachments involving the server, as well as local attachments specific to the server.

An ACFS file system on EDV can also be exported to clients outside the Exadata RDMA Network Fabric by using ACFS HANFS. For more information about ACFS, see Oracle Advanced Cluster File System Administrator's Guide.

EDV devices can also be used as raw devices or in conjunction with other Linux file systems, such as XFS, EXT4, and so on.

6 Using ESCLI

This chapter describes how to use the Exascale command line interface (ESCLI):

- Start and Use ESCLI
- ESCLI Command Reference

6.1 Start and Use ESCLI

This topic describes how to start and use the Exascale command line interface (ESCLI).

ESCLI is a command-line administration tool that is located on Exadata compute nodes and storage servers. You can use ESCLI to perform Exascale monitoring and management functions. ESCLI works in conjunction with the Exadata cell command-line interface (CellCLI) and does not replace it.

To use ESCLI you must have access to an Exascale user account and the digital key store (wallet) for the Exascale user account.

To start ESCLI, use the following command line syntax:

```
$ escli [ --wallet wallet-location ] [ --ctrl server[:port] ] [ ESCLI-command ]
```

In the command line:

- --wallet *wallet-location*: Identifies the wallet that you want to use for authentication to Exascale. If you do not specify the wallet location, then ESCLI checks the local server and uses the first Exascale user wallet available in the following search path:
    1. $OSSCONF/eswallet
    2. $ORACLE_BASE/admin/eswallet
    3. /etc/oracle/cell/network-config/eswallet
- --ctrl *server*[:*port*]: Specifies the endpoint address for connection to Exascale control services, also known as Exascale RESTful Services (ERS). If the ERS address is not specified on the command line, then ESCLI connects to the ERS address stored in the user's wallet.
- *ESCLI-command*: Specifies an ESCLI command to run immediately. For example:

```
$ escli --wallet /home/user/user.wallet ls -l @my-data
Total 5
  10.0M   05 Jan 12:59 x
  19.5k   05 Jan 13:08 y
   5.0k   05 Jan 13:09 z1
  10.0M   05 Jan 13:23 z2
  20.0G   05 Jan 13:14 z3
$
```

If an ESCLI command is not specified on the escli command line, ESCLI starts an interactive session and presents a command prompt. During an interactive session, you can run a series of ESCLI commands. The interactive session ends with the exit command. For example:

```
$ escli --wallet /home/user/user.wallet
@> ls
my-data
vault2

@> cd @my-data

@my-data/> ls -l
Total 5
  10.0M   05 Jan 12:59 x
  19.5k   05 Jan 13:08 y
   5.0k   05 Jan 13:09 z1
  10.0M   05 Jan 13:23 z2
  20.0G   05 Jan 13:14 z3

@my-data/> exit
$
```

During an interactive session, ESCLI supports command auto-completion by using the <tab> key. When you press the <tab> key, ESCLI attempts to complete the current command token. If multiple possibilities exist, ESCLI provides a list of alternatives. For example:

```
@> lsc<tab>
lscell              (List all cells in the cluster)
lscelldisk          (Lists cell disks in this cluster)
lscluster               (List cluster information)
lscomputeserver (List all compute servers in the cluster)
```

Command auto-completion also works within a command to suggest or complete command options. For example:

```
@> lscell --<tab>
--attributes                (Lists the specified attributes)
--count                 (Maximum number of results to report)
--detail              (Lists all attributes in a detailed form)
--filter (Used to specify conditions for filtering the list output)
--sort     (Used to sort the output using the specified attributes)
```

6.2 ESCLI Command Reference

This section contains references for the Exascale command line interface (ESCLI) commands. Apart from command help and object description, which are universal, the commands are grouped by functional area:

- ESCLI Command Help
- Describing Resources and Attributes
- Service and Cluster Management
- Security and User Management

- Storage Pool and Pool Disk Management
- Vault Management
- Dataset Management
- File Management
- Template Management
- Resource Profile Management
- Extended Attribute Management
- Block Store Management

6.2.1 ESCLI Command Help

Purpose

The ESCLI help command displays help information for Exascale command line interface (ESCLI) commands.

Syntax

```
help ( command | list | all )
```

Command Options

The options for the help command are:

- *command*: Specifies the ESCLI command for which you want to print help information.
- list: Lists all of the available ESCLI commands.
- all: Prints help information for all of the available ESCLI commands.

Examples

Example 6-1 Print Help Information for a Specific Command

The following example prints help information for the ls command.

```
@> help ls
```

6.2.2 Describing Resources and Attributes

Purpose

The describe command displays information about Exascale resources and their attributes.

Syntax

```
describe { --all | resource-spec } [ --detail ]
```

Command Options

The options for the describe command are:

- --all: Directs the command to display information for all Exascale resources. This option displays a large amount of information, but it is a good starting point if you are unfamiliar with Exascale resources and their attributes.
- *resource-spec*: Directs the command to display information about specific Exascale resources and their attributes as follows:
  - The *resource-spec* value is not case-sensitive.

    Consequently, the following commands are functionally equivalent:

    ```
    @> describe vaults
    ```

    ```
    @> describe VAULTS
    ```

    ```
    @> describe vAUltS
    ```

  - You can specify resource names in singular or plural form.

    Consequently, the following commands are functionally equivalent:

    ```
    @> describe vault
    ```

    ```
    @> describe vaults
    ```

  - If you specify a complete resource name (in singular or plural form), the output contains all attributes of the specified resource.
  - If you specify a resource name with a wildcard (*), the output contains all resources that match the pattern.

    For example, the following command describes all resources starting with the letter 'v' (vaults, volumes, and so on):

    ```
    @> describe v*
    ```

  - If you specify an attribute name (using the notation: <resource>.<attribute>), the output describes the specified attribute.

    For example, the following command describes the attribute that contains the vault name.

    ```
    @> describe vaults.name
    ```

  - If you specify an attribute name with a wildcard (*), the output contains all attributes that match the pattern.

    For example, the following command describes all attributes starting with 'n' in all resources starting with 'v':

    ```
    @> describe v*.n*
    ```

  - If *resource-spec* begins with 'mk', the output contains only the attributes that are initializable during resource creation (initializable != No).

For example, the following command describes all attributes that are initializable during file creation:

```
@> describe mkfiles
```

You can further refine the output using the following options:

- --mandatory: Describes only the mandatory attributes that must be initialized during resource creation (initializable = Mandatory).

  For example:

  ```
  @> describe mkfiles --mandatory
  ```

- --optional: Describes only the optional attributes that may be initialized during resource creation (initializable = Optional).

  For example:

  ```
  @> describe mkfiles --optional
  ```

- If *resource-spec* begins with 'ch', the output contains only the attributes that are modifiable for an existing resource (modifiable = true).

  For example, the following command describes the attributes that may be modified for existing files:

  ```
  @> describe chfiles
  ```

- --detail: Optionally formats the command output so that values are displayed on separate lines, instead of using the default tabular output.

Usage Notes

For each resource, the output includes the resource name and a list of associated attributes. For each attribute, the following information is provided:

- name: The attribute name.
- type: The data type for the attribute value (for example: String, Integer, Date, Boolean, and so on).
- description: A brief description of the attribute, including valid values and default settings when applicable.
- initializable: Specifies whether the attribute value can or must be set during resource creation. Possible values are:
  - No: The attribute value cannot be set during resource creation. The value is set automatically by the system or inherits a default value.
  - Mandatory: The attribute value must be specified during resource creation.
  - Optional: The attribute value may be specified during resource creation. If not specified, the attribute is typically assigned a default value.
- modifiable: Indicates whether the attribute value can be modified after the resource is created (either true or false).

6.2.3 Service and Cluster Management

This section contains references for the Exascale command line interface (ESCLI) commands that are associated with service and cluster management:

- cellcli
  Run a CellCLI command on a specified Exadata storage server.
- chcluster
  Shut down the Exascale cluster or to modify its attributes.
- chfeature
  Manage an Exascale feature.
- chfeatureupdate
  Change attributes for a deferred feature update.
- chservice
  Start, stop, restart, or disable a software service.
- dbmcli
  Execute a DBMCLI command on a compute server.
- lscell
  List storage cells.
- lscelldisk
  List cell disks.
- lscluster
  List cluster information.
- lscomputeserver
  List compute servers.
- lsfeature
  List features associated with an Exascale cluster.
- lsfeatureupdate
  List information about deferred feature updates.
- lsgriddisk
  List grid disks.
- lsservice
  List software services.
- mkfeature
  Create an Exascale feature definition.

6.2.3.1 cellcli

Run a CellCLI command on a specified Exadata storage server.

Syntax

```
cellcli --cell cell-name [ --xml ] [ -e ] CellCLI-command
```

Command Options

The options for the `cellcli` command are:

- *cell-name*: Specifies the name of cell that is the subject of the operation.

  The cell must be identified by its name. Use the lscell command to find all cell names.
- --xml: Generates XML-formatted command output.
- -e *CellCLI-command*: Specifies the CellCLI command. The -e argument is optional.

Usage Notes

To run the cellcli command in ESCLI, the Exascale user must have the cl_admin privilege. On the Exadata storage server, the command runs with the privilege of the celladmin user.

Examples

Example 6-2 Run a CellCLI command

In the following examples, the list cell command is run on CELL1. Both commands perform the same function. However, the second command specifies the optional -e argument.

```
@> cellcli --cell CELL1 list cell
```

```
@> cellcli --cell CELL1 -e list cell
```

Example 6-3 Run a CellCLI command and generate XML-formatted output

In this example, the list cell command is run on CELL1 and the command output is rendered using XML.

```
@> cellcli --cell CELL1 --xml list cell
```

6.2.3.2 chcluster

Shut down the Exascale cluster or to modify its attributes.

Syntax

```
chcluster [ --attributes attribute=value[,attribute=value]... ]
    [ --shutdown ]
```

Command Options

The options for the chcluster command are:

- --attributes: Specifies one or more attribute settings for the cluster.

  Use the describe chcluster command to see details about the cluster attributes you can change. See also Describing Resources and Attributes.
- --shutdown: Shuts down the Exascale cluster.

  To shut down the Exascale cluster, note that you must:

  - Run ESCLI as the Exascale admin user or as another user with the cl_admin cluster level storage privilege.

  - Connect ESCLI directly to an online Exascale control services (ERS) back-end server process. To do this you must start ESCLI and specify the --ctrl option as follows:

    ```
    $ escli --wallet admin-wallet-location --ctrl ERS-server-IP:8080
    ```

    To find the IP address for an online ERS server (*ERS-server-IP*), start a regular ESCLI session and use the following command:

    ```
    @> lsservice --filter serviceType=controlServices,status=ONLINE --attributes url
    ```

    The command displays the private IP addresses associated with all of the online ERS servers. You can use any of the reported IP addresses.

Examples

Example 6-4 Modify the Cluster

This example shows setting the attribute that enables automatic service placement in the Exascale cluster.

```
@> chcluster --attributes servicePlacement=auto
```

Example 6-5 Shut Down the Cluster

This example shows how to shut down the Exascale cluster.

```
$ escli --wallet admin-wallet-location --ctrl ERS-server-IP:8080 chcluster -–shutdown
```

6.2.3.3 chfeature

Manage an Exascale feature.

Purpose

The chfeature command allows you to manage an Exascale software feature.

Syntax

```
chfeature feature-ID [ --attributes attribute=value[,attribute=value]... ]
     [ --disable ] [ --enable ]
```

Command Options

The options for the chfeature command are:

- *feature-ID*: Identifies the feature that is the subject of the operation.
- --attributes: Modifies attributes of the specified feature. This option is used only to downgrade a feature to a specified version and change associated participants if necessary.
- --disable: Disables the specified feature.
- --enable: Enables the specified feature.

6.2.3.4 chfeatureupdate

Change attributes for a deferred feature update.

Purpose

The chfeatureupdate command allows you to change attributes associated with a deferred feature update.

Syntax

```
chfeatureupdate name [ --attributes attribute=value[,attribute=value]... ]
```

Command Options

The options for the chfeatureupdate command are:

- *name*: Identifies the deferred feature update that is the subject of the operation.
- --attributes: Specifies one or more attribute settings for the deferred feature update.

  Use the describe chfeatureupdate command to see details about the deferred feature update attributes you can change. See also Describing Resources and Attributes.

Examples

Example 6-6 Change the Time When a Deferred Feature Update Takes Effect

The following example shows how to change the time when a deferred feature update is scheduled to take effect.

```
@> chfeatureupdate deferred_update_1_1684214264 --attributes expiryTime=2024-01-01T12:31:01
```

6.2.3.5 chservice

Start, stop, restart, or disable a software service.

Purpose

The chservice command allows you to start, stop, restart, or disable an Exascale software service. A chservice command can only perform one operation on one service instance.

Syntax

```
chservice --attributes name=service_cell[,frontend={true|false}]
      { --start | --stop | --restart | --disable [ --force ]}
```

```
chservice --attributes name=service_compute
      { --start | --stop | --restart | --disable [ --force ]}
```

Command Options

The options for the chservice command are:

- --attributes: Primarily identifies the server and service that is the subject of the operation. This option is also used to specify optional attribute settings.
  - name: Identifies the server and service that is the subject of the operation.

    For a service on an Exadata storage server, specify the name attribute using the format: name=*service_cell*. In the attribute value:
    - *service*: Specifies the service that you want to act on, which is one of the following:
      - cellsrv: Specifies the Exadata Cell Server (CELLSRV).
      - egs: Specifies the Exascale cluster service, also known as Exascale global service (EGS).
      - ers: Specifies the Exascale control service, also known as Exascale RESTful service (ERS).
      - syseds: Specifies the system vault manager service.
      - usreds: Specifies the user vault manager service.
      - bsm: Specifies the block store manager (BSM) service.
      - bsw: Specifies the block store worker (BSW) service.
    - *cell*: Specifies the name of the storage server (cell) that hosts the service.

      The cell must be identified by its name. Use the lscell command to find all cell names.

    For a service on an Exadata compute server, specify the name attribute using the format: name=*service_compute*. In the attribute value:
    - *service*: Specifies the service that you want to act on, which is one of the following:
      - egs: Specifies the Exascale cluster service, also known as Exascale global service (EGS).
      - bsw: Specifies the block store worker service.
    - *compute*: Specifies the name of the compute server (dbserver) that hosts the service.

      Depending on the required service location, specify either the name of a specific compute node virtual machine (VM) or the Exadata compute node host domain (KVM host or Dom0). Use the lscomputeserver command to display information about compute servers associated with the Exascale cluster.
  - frontend: Optionally specifies whether to start ERS front-end server processes. This option is only valid when starting or restarting an ERS instance.

    To start the front-end server processes on the ERS instance, specify frontend=true. To not start the front-end server processes on the ERS instance, specify frontend=false.
- You must also specify one of the following operations:
  - --start: Starts a service. The service is also enabled if it was previously disabled.
  - --stop: Stops a service.
  - --restart: Restarts a service. The service is also enabled if it was previously disabled.
  - --disable: Disables a service. A disabled service is stopped and prevented from automatically restarting on the associated server.

    You can also add the --force option to forcibly disable the service, even if the server hosting the service is unavailable. This option only applies when disabling a service.

Examples

Example 6-7 Start the System Vault Manager Service (SYSEDS)

In this example, the system vault manager service is started on CELL1.

```
@> chservice --attributes name=syseds_CELL1 --start
```

Example 6-8 Restart the Exascale Control Service (ERS)

In this example, ERS is restarted on CELL2, along with the front-end ERS server processes.

```
@> chservice --attributes name=ers_CELL2,frontend=true --restart
```

6.2.3.6 dbmcli

Execute a DBMCLI command on a compute server.

Purpose

The dbmcli command enables you to run a DBMCLI command on a specified compute server.

Syntax

```
dbmcli --compute compute-name [ --xml ] [ -e ] dbmcli-command
```

Command Options

The options for the dbmcli command are:

- *compute-name*: Specifies the name of compute server that is the subject of the operation.

  The compute server must be identified by its name. Use the lscomputeserver command to find all compute server names.
- --xml: Generates XML-formatted command output.
- -e *dbmcli-command*: Specifies the DBMCLI command. The -e argument is optional.

Usage Notes

To run the dbmcli command in ESCLI, the Exascale user must have the cl_admin privilege. On the Exadata compute server, the command runs with the privilege of the dbmadmin user.

Examples

Example 6-9 Run a DBMCLI command

In the following examples, the list dbserver command is run on DB01. Both commands perform the same function. However, the second command specifies the optional -e argument.

```
@> dbmcli --compute DB01 list dbserver
```

```
@> dbmcli --compute DB01 -e list dbserver
```

Example 6-10 Run a DBMCLI command and generate XML-formatted output

In this example, the list dbserver command is run on DB01 and the command output is rendered using XML.

```
@> dbmcli --compute DB01 --xml list dbserver
```

6.2.3.7 lscell

List storage cells.

Purpose

The lscell command displays information about Exadata storage servers in the Exascale cluster.

Syntax

```
lscell [ -l ] [ --detail ] [ --attributes attribute[,attribute]... ]
    [ --filter filter[,filter]... ]
    [ --sort [-]attribute[,[-]attribute]... ]
    [ --count value ]
```

Command Options

The options for the lscell command are:

- -l: Displays output in a long, tabular format.
- --detail: Displays detailed output with additional attributes.
- --attributes: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the describe cells command. See also Describing Resources and Attributes.
- --filter: Specifies conditions for including items in the output.
- --sort: Sorts the output using the specified attributes.
- --count: Specifies the maximum number of items to display in the output.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: <*attribute*><*operator*><*value*>.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  - yyyy-MM-dd"T"HH:mm:ss
  - yyyy-MM-dd (Time is assumed to be 00:00 AM)
  - HH:mm:ss (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  Sizes can be specified using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.

For example, the following filter only includes objects created after 9 AM today:

```
--filter createTime>9:00:00
```

- Sorting attributes are specified as: [-]*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with -.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

Examples

Example 6-11 List Storage Servers

The following example shows how to list essential information about all of the storage servers in the Exascale cluster.

```
@> lscell
```

Example 6-12 List Storage Servers

The following example shows how to list detailed information about all of the storage servers in the Exascale cluster.

```
@> lscell --detail
```

6.2.3.8 lscelldisk

List cell disks.

Purpose

The lscelldisk command displays information about Exadata storage server cell disks in the Exascale cluster.

Syntax

```
lscelldisk [ celldisk [ celldisk ]... ] [ -l ] [ --detail ]
        [ --attributes attribute[,attribute]... ]
        [ --filter filter[,filter]... ]
        [ --sort [-]attribute[,[-]attribute]... ]
        [ --count value ]
```

Command Options

The options for the lscelldisk command are:

- *celldisk*: Identifies an Exadata cell disk that you want to list information about. If not specified, the command displays information about all cell disks.
- -l: Displays output in a long, tabular format.
- --detail: Displays detailed output with additional attributes.

- `--attributes`: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the `describe celldisks` command. See also Describing Resources and Attributes.
- `--filter`: Specifies conditions for including items in the output.
- `--sort`: Sorts the output using the specified attributes.
- `--count`: Specifies the maximum number of items to display in the output.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  - `yyyy-MM-dd"T"HH:mm:ss`
  - `yyyy-MM-dd` (Time is assumed to be 00:00 AM)
  - `HH:mm:ss` (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  Sizes can be specified using suffixes `K`, `KB`, `M`, `MB`, `G`, `GB`, `T`, `TB`. The suffix is not case-sensitive.

  For example, the following filter only includes objects created after 9 AM today:

  ```
  --filter createTime>9:00:00
  ```

- Sorting attributes are specified as: `[-]`*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with `-`.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

Examples

Example 6-13   List Cell Disk Information

The following example shows how to list detailed information for all cell disks in the Exascale cluster.

```
@> lscelldisk --detail
```

Example 6-14   List Information for Specific Cell Disks

The following example shows how to list information about specific cell disks named `DISK1`.

```
@> lscelldisk --filter name=DISK1
```

6.2.3.9 lscluster

List cluster information.

Purpose

The lscluster command displays information about the Exascale cluster.

Syntax

```
lscluster [ -l ] [ --detail ] [ --attributes attribute[,attribute]... ]
        [ --filter filter[,filter]... ]
        [ --sort [-]attribute[,[-]attribute]... ]
        [ --count value ] [ --backup ]
```

Command Options

The options for the lscluster command are:

- -l: Displays output in a long, tabular format.
- --detail: Displays detailed output with additional attributes.
- --attributes: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the describe clusters command. See also Describing Resources and Attributes.
- --filter: Specifies conditions for including items in the output.
- --sort: Sorts the output using the specified attributes.
- --count: Specifies the maximum number of items to display in the output.
- --backup: Limits the output to list only volume backup attributes.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  – yyyy-MM-dd"T"HH:mm:ss

  – yyyy-MM-dd (Time is assumed to be 00:00 AM)

  – HH:mm:ss (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  Sizes can be specified using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.

  For example, the following filter only includes objects created after 9 AM today:

  ```
  --filter createTime>9:00:00
  ```

- Sorting attributes are specified as: [-]*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with -.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

Examples

Example 6-15   List Cluster Information

The following example shows how to list essential information about the Exascale cluster.

```
@> lscluster
```

Example 6-16   List Cluster Information

The following example shows how to list detailed information about the Exascale cluster.

```
@> lscluster --detail
```

Example 6-17   List Cluster Information

The following example shows how to list detailed information about volume backup attributes that are associated with the Exascale cluster.

```
@> lscluster --detail --backup
```

6.2.3.10 lscomputeserver

List compute servers.

Purpose

The lscomputeserver command displays information about compute servers associated with the Exascale cluster.

Syntax

```
lscomputeserver [ -l ] [ --detail ] [ --attributes attribute[,attribute]... ]
            [ --filter filter[,filter]... ]
            [ --sort [-]attribute[,[-]attribute]... ]
            [ --count value ]
```

Command Options

The options for the lscomputeserver command are:

- -l: Displays output in a long, tabular format.
- --detail: Displays detailed output with additional attributes.
- --attributes: Identifies specific attributes to display in the output.

To see a list of all available attributes and their descriptions, use the `describe computes` command. See also Describing Resources and Attributes.

- `--filter`: Specifies conditions for including items in the output.
- `--sort`: Sorts the output using the specified attributes.
- `--count`: Specifies the maximum number of items to display in the output.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  – `yyyy-MM-dd"T"HH:mm:ss`

  – `yyyy-MM-dd` (Time is assumed to be 00:00 AM)

  – `HH:mm:ss` (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  Sizes can be specified using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.

  For example, the following filter only includes objects created after 9 AM today:

  ```
  --filter createTime>9:00:00
  ```

- Sorting attributes are specified as: [-]*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with -.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

Examples

Example 6-18 List Compute Servers

The following example shows how to list essential information about all of the compute servers associated with the Exascale cluster.

```
@> lscomputeserver
```

Example 6-19 List Compute Servers

The following example shows how to list detailed information about all of the compute servers associated with the Exascale cluster.

```
@> lscomputeserver --detail
```

6.2.3.11 lsfeature

List features associated with an Exascale cluster.

Purpose

The `lsfeature` command displays information about software features in the Exascale cluster.

Syntax

```
lsfeature [ --detail ] [ --filter filter[,filter]... ] [ -l ]
```

Command Options

The options for the `lsfeature` command are:

- `--detail`: Displays detailed output with additional attributes.
- `--filter`: Specifies conditions for including items in the output.
- `-l`: Displays output in a long, tabular format.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  – `yyyy-MM-dd"T"HH:mm:ss`

  – `yyyy-MM-dd` (Time is assumed to be 00:00 AM)

  – `HH:mm:ss` (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  Sizes can be specified using suffixes `K`, `KB`, `M`, `MB`, `G`, `GB`, `T`, `TB`. The suffix is not case-sensitive.

  For example, the following filter only includes objects created after 9 AM today:

  ```
  --filter createTime>9:00:00
  ```

6.2.3.12 lsfeatureupdate

List information about deferred feature updates.

Purpose

The `lsfeatureupdate` command displays information about all deferred feature updates associated with the Exascale cluster or details for a specific deferred feature update.

Syntax

```
lsfeatureupdate [ name ] [ --attributes attribute[,attribute]... ]
            [ --detail ] [ --filter filter[,filter]... ] [ -l ]
```

Command Options

The options for the lsfeatureupdate command are:

- *name*: Identifies the deferred feature update that you want to list information about. If not specified, the command displays information about all deferred feature updates associated with the Exascale cluster.
- --attributes: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the describe featureupdates command. See also Describing Resources and Attributes.
- --detail: Displays detailed output with additional attributes.
- --filter: Specifies conditions for including items in the output.
- -l: Displays output in a long, tabular format.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  - yyyy-MM-dd"T"HH:mm:ss
  - yyyy-MM-dd (Time is assumed to be 00:00 AM)
  - HH:mm:ss (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  Sizes can be specified using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.

  For example, the following filter only includes objects created after 9 AM today:

  ```
  --filter createTime>9:00:00
  ```

Examples

Example 6-20 List All Deferred Feature Update Information

The following example shows how to list essential information about all of the deferred feature updates associated with the Exascale cluster.

```
@> lsfeatureupdate
```

Example 6-21 List Specific Deferred Feature Update Information

The following example shows how to list detailed information about a specific deferred feature update.

```
@> lsfeatureupdate deferred_update_1_1684214264 --detail
```

6.2.3.13 lsgriddisk

List grid disks.

Purpose

The `lsgriddisk` command displays information about Exadata storage server grid disks in the Exascale cluster.

Syntax

```
lsgriddisk [ grid-disk [ grid-disk ]... ] [ -l ] [ --detail ]
           [ --attributes attribute[,attribute]... ]
           [ --filter filter[,filter]... ]
           [ --sort [-]attribute[,[-]attribute]... ]
           [ --count value ]
```

Command Options

The options for the `lsgriddisk` command are:

- *grid-disk*: Identifies an Exadata grid disk that you want to list information about. If not specified, the command displays information about all grid disks.
- `-l`: Displays output in a long, tabular format.
- `--detail`: Displays detailed output with additional attributes.
- `--attributes`: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the `describe griddisks` command. See also Describing Resources and Attributes.
- `--filter`: Specifies conditions for including items in the output.
- `--sort`: Sorts the output using the specified attributes.
- `--count`: Specifies the maximum number of items to display in the output.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  - `yyyy-MM-dd"T"HH:mm:ss`
  - `yyyy-MM-dd` (Time is assumed to be 00:00 AM)

  - HH:mm:ss (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  Sizes can be specified using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.

  For example, the following filter only includes objects created after 9 AM today:

  ```
  --filter createTime>9:00:00
  ```

- Sorting attributes are specified as: [-]*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with -.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

Examples

Example 6-22 List Grid Disk Information

The following example shows how to list detailed information for all grid disks in the Exascale cluster.

```
@> lsgriddisk --detail
```

Example 6-23 List Information for Specific Grid Disks

The following example shows how to list information about specific grid disks named DISK2.

```
@> lsgriddisk --filter name=DISK2
```

6.2.3.14 lsservice

List software services.

Purpose

The lsservice command displays information about software services in the Exascale cluster.

Syntax

```
lsservice [ -l ] [ --detail ] [ --attributes attribute[,attribute]... ]
    [ --filter filter[,filter]... ]
    [ --sort [-]attribute[,[-]attribute]... ]
    [ --count value ]
```

Command Options

The options for the lsservice command are:

- -l: Displays output in a long, tabular format.
- --detail: Displays detailed output with additional attributes.

- `--attributes`: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the `describe services` command. See also Describing Resources and Attributes.

- `--filter`: Specifies conditions for including items in the output.
- `--sort`: Sorts the output using the specified attributes.
- `--count`: Specifies the maximum number of items to display in the output.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  - `yyyy-MM-dd"T"HH:mm:ss`
  - `yyyy-MM-dd` (Time is assumed to be 00:00 AM)
  - `HH:mm:ss` (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  Sizes can be specified using suffixes `K`, `KB`, `M`, `MB`, `G`, `GB`, `T`, `TB`. The suffix is not case-sensitive.

  For example, the following filter only includes objects created after 9 AM today:

  ```
  --filter createTime>9:00:00
  ```

- Sorting attributes are specified as: `[-]`*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with `-`.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

Examples

Example 6-24 List Service Information

The following example shows how to list essential information about all of the services in the Exascale cluster.

```
@> lsservice
```

Example 6-25 List System Vault Manager Service Information

The following example shows how to list detailed information about system vault manager services in the Exascale cluster.

```
@> lsservice --detail --filter serviceType=systemVaultManagers
```

6.2.3.15 mkfeature

Create an Exascale feature definition.

Purpose

The mkfeature creates the definition for an Exascale software feature.

Syntax

```
mkfeature feature-ID
```

Command Options

The options for the mkfeature command are:

- *feature-ID*: Specifies the feature identifier.

6.2.4 Security and User Management

This section contains references for the Exascale command line interface (ESCLI) commands that are associated with security and user management:

- chacl
  Change an access control list (ACL).
- chuser
  Change attributes of a user, or upload API keys.
- chwallet
  Change authentication-related data in a local wallet.
- lsacl
  List ACLs for files and vaults.
- lskey
  Display key information.
- lsuser
  List user information.
- lswallet
  Display information stored in a wallet.
- mkkey
  Generate and write keys in PEM format.
- mkuser
  Create a new user.
- mkwallet
  Create a local wallet.
- rmuser
  Delete a user.

6.2.4.1 chacl

Change an access control list (ACL).

Purpose

The chacl command allows you to change the ACL for an Exascale vault or file.

Syntax

```
chacl  { file-name | vault-name } acl-string
```

Command Options

The options for the chacl command are:

- { *file-name* | *vault-name* }: Specifies the name of the file or vault that is the subject of the operation. Only one file or vault name is permitted (no wildcards).
- *acl-string*: Specifies an ACL string having the following format.

  ```
  [+]user-ID1:acl-permission[;user-ID2:acl-permission]...
  ```

  In the ACL string:

  - The optional plus (+) operator at the beginning of the ACL string indicates that the specified ACL string merges into the existing ACL. In this case, the changes only impact the users in the specified ACL string, and all other users in the existing ACL retain their permissions. Without the optional plus (+) operator, the ACL is completely overwritten by the specified ACL string.
  - *user-IDn*: Specifies an Exascale user ID.

    Depending on the user creation method, the user ID may be a system-generated value (for example, 96a68014-5762-4579-86ee-29eb743decbd) or a user-specified value (for example, scott).
  - *acl-permission*: Specifies an ACL permission setting, which can be one of the following:
    - I | inspect: Specifies that the user can view attributes of the file or vault, but not its contents.
    - R | read: Specifies that the user can read contents of the file, or list files in the vault. Also confers the inspect permission.
    - U | use: Specifies that the user can write to the file, or use the vault. Also confers all preceding permissions.
    - M | manage: Specifies that the user can manage the file or vault. Also confers all preceding permissions.
    - 0 | none: Specifies that the user is removed from the ACL and loses all permissions. This setting can be used only in conjunction with the plus (+) operator to remove a user from an existing ACL.

Examples

Example 6-26 Replace a File ACL

In this example, the ACL string for the file @my-vault/my-file is replaced with the new ACL string. Under the new ACL, scott is permitted to read and inspect the file. Any other users included in the previous ACL effectively lose all previously held permissions.

```
@> chacl @my-vault/my-file scott:R
```

Example 6-27 Change a File ACL

In this example, the plus sign (+) at the beginning of the ACL string indicates that the specified ACL string is merged into the existing file ACL. In this case, any pre-existing permissions for jason are overwritten, and jason is now permitted to inspect, read, write, and manage the file. Also, scott is removed from the ACL and loses all previously held permissions. No other ACL permissions are changed.

```
@> chacl @my-vault/my-file +jason:manage;scott:none
```

Example 6-28 Replace a File ACL using an ACL String that Specifies Multiple Users

In this example, the ACL string for the file is replaced with the new ACL string that specifies permissions for multiple users. Under the new ACL, scott can inspect the file, and jason can read and inspect the file. Any other users included in the previous ACL effectively lose all previously held permissions.

```
@> chacl @my-vault/my-file scott:inspect;jason:read
```

6.2.4.2 chuser

Change attributes of a user, or upload API keys.

Purpose

The chuser command allows you to change an attribute for an existing Exascale user, or manage API keys for a user.

Syntax

```
chuser user-ID [ --attributes attribute=value[,attribute=value]... ]
    [ --privilege privileges ] [ --public-key-file1 public-key ]
    [ --public-key-file2 public-key ] [ --public-key-file3 public-key ]
    [ --rm-public-key1 ] [ --rm-public-key2 ] [ --rm-public-key3 ]
```

Command Options

The options for the chuser command are:

- *user-ID*: Identifies the unique user ID for the user that is the subject of the operation. You can use the lsuser command to find the ID for each user.
- --attributes: Optionally specifies values for attributes of the user.

Use the describe chuser command to see details about the user attributes you can change. See also Describing Resources and Attributes.

- --privilege: Changes the user's privileges. The *privileges* value is a list of one or more privileges of the form [+|-]*privilege-1*|*privilege-2*|....

  A privilege is one of the following: no_privilege, cellsrv, egs, ers, syseds, usreds, bsm, bsw, ms, vlt_manage, vlt_use, vlt_read, vlt_inspect, cl_admin, cl_operator, cl_monitor, on_behalf_of, user_create, system_restore. For descriptions, see User Privileges.

  no_privilege cannot be combined with any other type of privileges, otherwise an error is returned.

  Vault top-level privileges (vlt_manage, vlt_use, vlt_read, and vlt_inspect) are mutually exclusive. If any two or more are combined, an error is returned.

  Cluster privileges (cl_admin, cl_operator, and cl_monitor) are mutually exclusive. If any two or more are combined, an error is returned.

  An optional plus sign (+) at the beginning of the privilege string indicates that the specified privileges are added to the user's existing privileges. An optional minus sign (-) at the beginning of the privilege string indicates that the specified privileges are removed from the user's existing privileges. Otherwise, the user's existing privileges are overwritten by the specified privileges.

- --public-key-file1: Uploads a public key in slot 1.
- --public-key-file2: Uploads a public key in slot 2.
- --public-key-file3: Uploads a public key in slot 3.
- --rm-public-key1: Deletes the public key in slot 1.
- --rm-public-key2: Deletes the public key in slot 2.
- --rm-public-key3: Deletes the public key in slot 3.

Examples

Example 6-29 Change the Name of a User

In this example, the name of the specified user is changed to SCOTTY.

```
@> chuser 59f6dce4-5687-4728-8751-acfe99089be2 --attributes name=SCOTTY
```

Example 6-30 Assign the Vault Top-Level Read Privilege to a User

In this example, the vault top-level read privilege is assigned to the specified user.

```
@> chuser 59f6dce4-5687-4728-8751-acfe99089be2 --privilege vlt_read
```

Example 6-31 Upload a Public Key for a User

In this example, the public key contained in the file new-pub.pem is uploaded into public key slot 1 for the specified user.

```
@> chuser 59f6dce4-5687-4728-8751-acfe99089be2 --public-key-file1 new-pub.pem
```

Example 6-32 Delete a Public Key Associated With a User

In this example, the key in public key slot 1 is deleted for the specified user.

```
@> chuser 59f6dce4-5687-4728-8751-acfe99089be2 --rm-public-key1
```

6.2.4.3 chwallet

Change authentication-related data in a local wallet.

Purpose

The chwallet command changes ESCLI authentication-related data like the user identifier and private key in a local wallet.

Syntax

```
chwallet --wallet wallet-location
    [ --attributes attribute=value[,attribute=value]... ]
    [ --private-key-file pem-file [ --force ]]
    [ --fetch-trust-store ]
    [ --trusted-cert-file trust-file-name ]
    [ --clear-old-trusted-certs ]
    [ --private-key-remove ]
```

Command Options

The options for the chwallet command are:

- --wallet: Identifies the wallet that is the subject of the operation. *wallet-location* must be specified as a directory location or SSO file location. The wallet must already exist. A wallet can be created using the ESCLI mkwallet command.
- --attributes: Optionally specifies values for attributes of the wallet.

  To see various attributes of a wallet, use the lswallet command with the --detail option.
- --private-key-file: Specifies a regular file location that contains a private key in PEM format.
- --force: Optionally replaces the private key in the wallet without confirmation. This option works in conjunction with the --private-key-file option.
- --fetch-trust-store: Gets the trusted certificates from the Exascale storage cluster and stores them in the wallet.
- --trusted-cert-file: Reads trusted certificates from the specified file and stores them in the wallet. You can use this option to read a PEM format file containing one trusted certificate. Or, you can use a file containing multiple trusted certificates, which is a concatenation of multiple PEM format files, each containing one trusted certificate.
- --clear-old-trusted-certs: Removes all previous trusted certificates from the wallet.

  This option is useful for removing old and expired trusted certificates.

> **Note**
>
> This option also removes manually added trusted certificates and trusted certificates not issued by the trust store.

- --private-key-remove: Removes the private key from the wallet.

Usage Notes

Before running the chwallet command to modify the Exascale user associated with a wallet, shut down all clients using the wallet (including Exascale services, Oracle Database processes, Oracle Grid Infrastructure processes, and so on). If you modify the user associated with an active wallet, the associated client connections end abnormally.

This requirement applies only to modifying the Exascale user associated with a wallet. Other wallet operations, including key rotation and certificate updates, may occur while a wallet is in use.

Examples

Example 6-33 Set a Private Key and User Identifier in a Wallet

The following example shows setting the private key and user identifier inside the wallet at /home/user/user.wallet.

```
@> chwallet --wallet /home/user/user.wallet --private-key-file /home/user/privatekey.pem --attributes 
user=0b9b8510-f88e-4f7b-ac57-10943c73dbe8
```

Example 6-34 Set the URL for the Exascale cluster services (EGS) in a Wallet

The following example shows setting the URL for the Exascale cluster services (EGS) inside the wallet at /home/user/user.wallet.

```
@> chwallet --wallet /home/user/user.wallet --attributes exaRootUrl="egs=egsexc4:192.0.2.217:5045 
egs=egsexc4:192.0.2.218:5045 egs=egsexc4:192.0.2.219:5045"
```

Example 6-35 Fetch Trust Store Certificates to a Wallet

The following example gets the trusted certificates from the Exascale storage cluster and stores them in the specified wallet. The command also removes the certificates that previously resided in the wallet.

```
@> chwallet --wallet /home/user/user.wallet --fetch-trust-store --clear-old-trusted-certs
```

Example 6-36 Load Trust Store Certificates to a Wallet

The following examples read the trusted certificates from the specified PEM files and stores them in the specified wallet. Each command also removes the certificates that previously resided in the wallet.

Both examples are functionally equivalent assuming that /tmp/concatenated-trust.pem is a concatenation of /tmp/trust1.pem, /tmp/trust2.pem, and /tmp/trust3.pem.

```
@> chwallet --wallet /home/user/user.wallet --trusted-cert-file /tmp/trust1.pem --trusted-cert-file /tmp/
trust2.pem --trusted-cert-file /tmp/trust3.pem --clear-old-trusted-certs
```

```
@> chwallet --wallet /root/eswallet/ --trusted-cert-file /tmp/concatenated-trust.pem --clear-old-trusted-certs
```

Related Topics

- mkwallet
  Create a local wallet.

6.2.4.4 lsacl

List ACLs for files and vaults.

Purpose

The lsacl command allows you to list access control lists (ACLs) for Exascale files and vaults.

Syntax

```
lsacl  [ file-name | vault-name ]...
```

Usage Notes

Note the following information when using this command:

- *file-name* and *vault-name* specify the names of files or vaults that are the subject of the operation. You can specify multiple files or vaults for which you want to display ACLs. The asterisk (*) can be used for wildcard searches.
- If you do not specify any *file-name* or *vault-name*, then the output depends on the current working directory in the ESCLI session. If the current working directory is the root directory, then the command returns the ACLs for all vaults. If the current working directory is not the root directory, then the command returns the ACLs for all files under the current working directory.

Examples

Example 6-37   List ACLs for all vaults

This example lists the ACLs for all vaults.

```
@> lsacl
```

Example 6-38   List ACLs for all files in a vault

This example lists the ACLs for all files in the @my-data vault.

```
@> cd @my-data
@my-data/> lsacl
```

Example 6-39   List ACLs for a Vault and Files

This example lists the ACLs for the @my-data vault and the files that it contains.

```
@> lsacl @my-data @my-data/*
```

6.2.4.5 lskey

Display key information.

Purpose

The lskey command displays information about the specificed key file.

Syntax

```
lskey pem-file [ --attributes attribute[,attribute]... ] [ --detail ] [ -l ]
```

Command Options

The options for the lskey command are:

- *pem-file*: Identifies the key file that you want to display information about. The *pem-file* must specify a regular file location for a key file in PEM format.
- --attributes: Identifies specific attributes to display in the output.

  You can use this option to focus on a subset of attributes as an alternative to the --detail option.
- --detail: Displays detailed output with additional attributes.
- -l: Displays output in a long, tabular format.

Examples

Example 6-40   List Detailed Key Information

The following example displays detailed information from the key file named pub.pem.

```
@> lskey pub.pem --detail
```

Example 6-41   List Specific Key Information

The following example displays specific attributes (id, privateKeyFile, and publicKeyFingerprint) from the key file named priv.pem.

```
@> lskey priv.pem --attributes id,privateKeyFile,publicKeyFingerprint
```

6.2.4.6 lsuser

List user information.

Purpose

The lsuser command displays information about Exascale users.

Syntax

```
lsuser [ user-ID ] [ --attributes attribute[,attribute]... ]
    [ --detail ] [ -l ]
```

Command Options

The options for the lsuser command are:

- *user-ID*: Identifies the user that you want to list information about. If no user is identified, the command displays information about all users.
- --attributes: Identifies specific attributes to display in the output.

  You can use this option to focus on a subset of attributes as an alternative to the --detail option.
- --detail: Displays detailed output with additional attributes.
- -l: Displays output in a long, tabular format.

Examples

Example 6-42 List Information for a Specific User

The following example shows how to list detailed information about the user ID: SCOTT0123.

```
@> lsuser SCOTT0123 --detail
```

6.2.4.7 lswallet

Display information stored in a wallet.

Purpose

The lswallet command displays ESCLI authentication data that is stored in a wallet.

Syntax

```
lswallet [ --wallet wallet-location ] [ --detail ]
     [ --private-key-file pem-file ]
     [ --public-key-file pem-file ]
     [ --public-key-file8 pem-file ]
     [ --attributes attribute[,attribute]... ]
     [ --trusted-cert-file cert-file ]
```

Command Options

The options for the lswallet command are:

- --wallet: Identifies the wallet that you want to display information about. The *wallet-location* must be specified as a directory location or SSO file location. If not specified, the command displays information about the wallet used to authenticate the current ESCLI session.
- --detail: Displays detailed output with additional attributes, including the user ID and fingerprint of the public key from the wallet.

- --private-key-file: Extracts the private key from the wallet and stores it in the specified PEM file.
- --public-key-file: Extracts the public key from the wallet and stores it in the specified PEM file.
- --public-key-file8: Extracts the public key from the wallet in PKCS8 format and stores it in the specified PEM file.
- --attributes: Identifies specific attributes to display in the output.

  You can use this option to focus on a subset of attributes as an alternative to the --detail option.
- --trusted-cert-file: Extracts the trusted certificates from the wallet and saves them in PEM format files.

  Files are created with numeric suffixes 1, 2, 3, and so on. For example, if you specify --trusted-cert-file cert_file, then the certificates are stored in the files cert_file1, cert_file2, cert_file3, and so on.

Examples

Example 6-43 List Wallet Information

The following example displays essential information from the wallet used to authenticate the current ESCLI session.

```
@> lswallet
```

Example 6-44 List Detailed Wallet Information

The following example displays detailed information from the wallet used to authenticate the current ESCLI session.

```
@> lswallet --detail
```

Example 6-45 Extract a Private Key

The following example extracts the private key from the wallet used to authenticate the current ESCLI session. The private key is stored in /home/user/privatekey.pem

```
@> lswallet --private-key-file /home/user/privatekey.pem
```

Example 6-46 Extract a Public Key

The following example extracts the public key from the wallet used to authenticate the current ESCLI session. The public key is stored in /home/user/publickey.pem

```
@> lswallet --public-key-file /home/user/publickey.pem
```

Example 6-47 Extract Certificates to Files

The following example extracts the trusted certificates from the wallet used to authenticate the current ESCLI session. The certificates are stored in regular files /home/user/cert1, /home/user/cert2, /home/user/cert3, and so on.

```
@> lswallet --trusted-cert-file /home/user/cert
```

6.2.4.8 mkkey

Generate and write keys in PEM format.

Purpose

The mkkey command generates and writes a private key to a regular file in PEM format. You can also generate and write the corresponding public key to a regular file in PEM format.

Syntax

```
mkkey [ --private-key-file pem-file ] [ --public-key-file pem-file ]
    [ --attributes attribute=value[,attribute=value]... ]
```

Command Options

The options for the mkkey command are:

- --private-key-file: Specifies a regular file location for the private key.
- --public-key-file: Optionally specifies a regular file location for the corresponding public key.
- --attributes: Optionally specifies attributes to change. This option provides an alternative method for specifying the key file names.

Examples

Example 6-48 Generate a Key Pair

The following examples generate a key pair and stores the private and public keys in /home/user/privatekey.pem and /home/user/publickey.pem, respectively.

```
@> mkkey --private-key-file /home/user/privatekey.pem --public-key-file /home/user/publickey.pem
```

```
@> mkkey --attributes privateKeyFile=/home/user/privatekey.pem,publicKeyFile=/home/user/publickey.pem
```

6.2.4.9 mkuser

Create a new user.

Purpose

The mkuser command allows you to create a new Exascale user with the specified user name.

Syntax

```
mkuser user-name [ --attributes attribute=value[,attribute=value]... ]
```

Command Options

The options for the mkuser command are:

- *user-name*: Specifies the name of the user. The user name must start with an alphanumeric character, and can only contain alphanumeric characters, hyphens (-) and periods (.).

- --attributes: Optionally specifies attributes to set.

  Use the describe mkuser command to see details about the user attributes you can set. See also Describing Resources and Attributes.

Usage Notes

Note the following information when using this command:

- You can set the user identifier by specifying the id attribute.

  If specified, the user identifier must start with an alphanumeric character, and can only contain alphanumeric characters, hyphens (-) and periods (.).

  If none is specified, then a unique user ID is generated automatically and returned by the command. The user ID value is required when managing users using chuser, rmuser, and lsuser.

  The user ID is fixed at user creation time. There is no way to modify the user ID afterward.

- You can set the user's privileges by specifying the privilege attribute.

  The privilege value is a list of one or more privileges of the form *privilege-1|privilege-2|...*.

  A privilege is one of the following: no_privilege, cellsrv, egs, ers, syseds, usreds, bsm, bsw, ms, vlt_manage, vlt_use, vlt_read, vlt_inspect, cl_admin, cl_operator, cl_monitor, on_behalf_of, user_create, system_restore. For descriptions, see User Privileges.

  no_privilege cannot be combined with any other type of privileges, otherwise an error is returned.

  Vault top-level privileges (vlt_manage, vlt_use, vlt_read, and vlt_inspect) are mutually exclusive. If any two or more are combined, an error is returned.

  Cluster privileges (cl_admin, cl_operator, and cl_monitor) are mutually exclusive. If any two or more are combined, an error is returned.

  If the privilege attribute is not specified, then the user is created and assigned only the vlt_inspect privilege by default.

Examples

Example 6-49 Create a User

This example shows creating a user named CHERIE.

```
@> mkuser CHERIE
```

Example 6-50 Create a User with a Specific User Identifier

This example shows creating a user named SCOTT using SCOTT0123 as the user identifier.

```
@> mkuser SCOTT --attributes id=SCOTT0123
```

Example 6-51 Create a User with a Specific Privileges

This example shows creating a user named PETER with a specific set of privileges.

```
@> mkuser PETER --attributes privilege=vlt_read|ers
```

6.2.4.10 mkwallet

Create a local wallet.

Purpose

The mkwallet creates a new Oracle wallet at the specified location.

Syntax

```
mkwallet --wallet wallet-location
```

Command Options

The options for the mkwallet command are:

- --wallet: Specifies the location where the new wallet is created. *wallet-location* must be specified as a directory location or SSO file location.

Examples

Example 6-52 Create a Wallet

The following example shows creating a wallet at /home/user/user.wallet.

```
@> mkwallet --wallet /home/user/user.wallet
```

6.2.4.11 rmuser

Delete a user.

Purpose

The rmuser command deletes an existing Exascale user.

Syntax

```
rmuser user-ID
```

Command Options

The options for the rmuser command are:

- *user-ID*: Specifies the unique user ID for the user that is being deleted. You can use the lsuser command to find the ID for each user.

Examples

Example 6-53 Delete a User

This example shows deleting the user with the ID 0b9b8510-f88e-4f7b-ac57-10943c73dbe8.

```
@> rmuser 0b9b8510-f88e-4f7b-ac57-10943c73dbe8
```

6.2.5 Storage Pool and Pool Disk Management

This section contains references for the Exascale command line interface (ESCLI) commands that are associated with storage pool and pool disk management:

- chpooldisk
  Modify a pool disk.
- chstoragepool
  Modify a storage pool.
- lspooldisk
  List pool disks.
- lsstoragepool
  List storage pools.
- lsstoragepooloperation
  List ongoing storage pool operations.
- mkstoragepool
  Create a storage pool.
- rmstoragepool
  Delete a storage pool.

6.2.5.1 chpooldisk

Modify a pool disk.

Purpose

The chpooldisk command enables you to modify an Exascale pool disk.

Syntax

```
chpooldisk pooldisk-ID { --offline | --online }
```

Command Options

The options for the chpooldisk command are:

- *pooldisk-ID*: Identifies the Exadata pool disk that you want to modify. You can use the lspooldisk command to find the ID for each pool disk.
- --offline: Takes the pool disk offline.
- --online: Bring the pool disk online.

Examples

Example 6-54 Bring a Pool Disk Online

The following example shows how to bring a pool disk online.

```
@> chpooldisk 10 --online
```

6.2.5.2 chstoragepool

Modify a storage pool.

Purpose

The chstoragepool command enables you to modify an Exascale storage pool.

Syntax

```
chstoragepool storagepool-name
          [ --attributes attribute=value[,attribute=value]... ]
          [ --reconfig [ --force ]]
```

Command Options

The options for the chstoragepool command are:

- *storagepool-name*: Identifies an Exascale storage pool that you want to modify.
- --attributes: Optionally specifies one or more attribute settings for the storage pool.

  Use the describe chstoragepool command to see details about the storage pool attributes you can change. See also Describing Resources and Attributes.
- --reconfig: Reconfigures the storage pool and makes permanent any changes to the number or size of the underlying pool disks. Reconfiguration may trigger a rebalance operation to ensure that data is spread evenly across the pool disks in the storage pool. The time required for reconfiguration depends on the amount of data movement required to rebalance the storage pool.
- --force: Optionally forces a reconfiguration operation even if the system detects no apparent change to the underlying pool disks.

Examples

Example 6-55 Set the Disk Offline Timer for a Storage Pool

The following example shows how to set the disk offline timer attribute associated with the Exascale storage pool named my-pool1.

```
@> chstoragepool my-pool1 --attributes diskOfflineTimerInMins=60
```

Example 6-56 Rediscover Pool Disks and Reconfigure the Storage Pool

The following example shows how to forcibly re-discover the pool disks and reconfigure the specified storage pool.

```
@> chstoragepool my-pool1 --reconfig --force
```

6.2.5.3 lspooldisk

List pool disks.

Purpose

The lspooldisk command displays information about Exadata storage server pool disks in the Exascale cluster.

Syntax

```
lspooldisk [ pooldisk-ID [ pooldisk-ID ]... ] [ -l ] [ --detail ]
        [ --attributes attribute[,attribute]... ]
        [ --filter filter[,filter]... ]
        [ --sort [-]attribute[,[-]attribute]... ]
        [ --count value ]
```

Command Options

The options for the lspooldisk command are:

- *pooldisk-ID*: Identifies an Exadata pool disk that you want to list information about. If not specified, the command displays information about all pool disks.
- -l: Displays output in a long, tabular format.
- --detail: Displays detailed output with additional attributes.
- --attributes: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the describe pooldisks command. See also Describing Resources and Attributes.
- --filter: Specifies conditions for including items in the output.
- --sort: Sorts the output using the specified attributes.
- --count: Specifies the maximum number of items to display in the output.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  - yyyy-MM-dd"T"HH:mm:ss
  - yyyy-MM-dd (Time is assumed to be 00:00 AM)
  - HH:mm:ss (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  Sizes can be specified using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.

For example, the following filter only includes objects created after 9 AM today:

```
--filter createTime>9:00:00
```

- Sorting attributes are specified as: [-]*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with -.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

Examples

Example 6-57 List Pool Disk Information

The following example shows how to list detailed information for all pool disks in the Exascale cluster.

```
@> lspooldisk --detail
```

Example 6-58 List Information for Specific Pool Disks

The following example shows how to list information about specific pool disks named DISK3.

```
@> lspooldisk --filter name=DISK3
```

6.2.5.4 lsstoragepool

List storage pools.

Purpose

The lsstoragepool command displays information about Exascale storage pools.

Syntax

```
lsstoragepool [ storagepool-name [ storagepool-name ]... ]
        [ -l ] [ --detail ]
        [ --attributes attribute[,attribute]... ]
        [ --filter filter[,filter]... ]
        [ --sort [-]attribute[,[-]attribute]... ]
        [ --count value ]
```

Command Options

The options for the lsstoragepool command are:

- *storagepool-name*: Identifies an Exascale storage pool that you want to list information about. If not specified, the command displays information about all storage pools.
- -l: Displays output in a long, tabular format.
- --detail: Displays detailed output with additional attributes.

- --attributes: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the describe storagepools command. See also Describing Resources and Attributes.
- --filter: Specifies conditions for including items in the output.
- --sort: Sorts the output using the specified attributes.
- --count: Specifies the maximum number of items to display in the output.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  - yyyy-MM-dd"T"HH:mm:ss
  - yyyy-MM-dd (Time is assumed to be 00:00 AM)
  - HH:mm:ss (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  Sizes can be specified using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.

  For example, the following filter only includes objects created after 9 AM today:

  ```
  --filter createTime>9:00:00
  ```

- Sorting attributes are specified as: [-]*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with -.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

Examples

Example 6-59 List Storage Pool Information

The following example shows how to list information for all storage pools in the Exascale cluster.

```
@> lsstoragepool
```

Example 6-60 List Information for a Specific Storage Pool

The following example shows how to list detailed information about the storage pool named my-pool1.

```
@> lsstoragepool my-pool1 --detail
```

6.2.5.5 lsstoragepooloperation

List ongoing storage pool operations.

Purpose

The lsstoragepooloperation command displays information about storage pool operations that are currently active in the Exascale cluster.

Syntax

```
lsstoragepooloperation [ operation-ID [ operation-ID ]... ]
                 [ -l ] [ --detail ]
                 [ --attributes attribute[,attribute]... ]
                 [ --filter filter[,filter]... ]
                 [ --sort [-]attribute[,[-]attribute]... ]
                 [ --count value ]
```

Command Options

The options for the lsstoragepooloperation command are:

- *operation-ID*: Identifies an Exascale storage pool operation that you want to list information about. If not specified, the command displays information about all active operations.
- -l: Displays output in a long, tabular format.
- --detail: Displays detailed output with additional attributes.
- --attributes: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the describe storagepooloperations command. See also Describing Resources and Attributes.
- --filter: Specifies conditions for including items in the output.
- --sort: Sorts the output using the specified attributes.
- --count: Specifies the maximum number of items to display in the output.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: <*attribute*><*operator*><*value*>.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  – yyyy-MM-dd"T"HH:mm:ss

  – yyyy-MM-dd (Time is assumed to be 00:00 AM)

  – HH:mm:ss (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  Sizes can be specified using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.

For example, the following filter only includes objects created after 9 AM today:

```
--filter createTime>9:00:00
```

- Sorting attributes are specified as: [-]*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with -.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

Examples

Example 6-61 List Storage Pool Operations

The following example shows how to list information for all ongoing storage pool operations in the Exascale cluster.

```
@> lsstoragepooloperation
```

Example 6-62 List Information for a Specific Operation

The following example shows how to list detailed information about a specific operation.

```
@> lsstoragepooloperation 3192092 --detail
```

6.2.5.6 mkstoragepool

Create a storage pool.

Purpose

The mkstoragepool command creates a pool of Exascale storage.

Syntax

```
mkstoragepool storagepool-name
        [ --attributes attribute=value[,attribute=value]... ]
        [ --nowait ]
```

Command Options

The options for the mkstoragepool command are:

- --attributes: Optionally specifies one or more attribute settings for the storage pool.

  Use the describe mkstoragepool command to see details about the storage pool attributes you can set. See also Describing Resources and Attributes.

- --nowait: Specifies that the command returns immediately while the operation completes in the background. Without this option, the command does not return until the storage pool is created.

Usage Notes

Before you create a storage pool, you need some pool disks that are already associated with the storage pool name.

Examples

Example 6-63 Create a Storage Pool

The following example shows creating a storage pool named my-pool1.

```
@> mkstoragepool my-pool1
```

6.2.5.7 rmstoragepool

Delete a storage pool.

Purpose

The rmstoragepool command enables you to delete an Exascale storage pool.

Syntax

```
rmstoragepool storagepool-name
```

Command Options

The options for the rmstoragepool command are:

- *storagepool-name*: Identifies the Exascale storage pool that you want to delete.

Usage Notes

You can use the rmstoragepool command in cases where you want to:

- Remove storage hardware from the Exascale cluster.
- Migrate data to another storage pool.

Examples

Example 6-64 Delete a Storage Pool

The following example shows how to delete a storage pool named my-pool1.

```
@> rmstoragepool my-pool1
```

6.2.6 Vault Management

This section contains references for the Exascale command line interface (ESCLI) commands that are associated with vault management:

- chvault
  Change the attributes of a vault.
- mkvault
  Create a vault.

- rmvault
  Delete a vault.

6.2.6.1 chvault

Change the attributes of a vault.

Purpose

The chvault command modifies the attributes of a vault.

Syntax

```
chvault vault --attributes attribute=value[,attribute=value]...
```

Command Options

The options for the chvault command are:

- *vault*: Identifies the name of the vault that you want to change.
- --attributes: Specifies attributes to change.

  Use the describe chvault command to see details about the vault attributes you can change. See also Describing Resources and Attributes.

Usage Notes

Note the following information when using this command:

- The command only changes the specified attributes. Other settings are unaffected.
- Following is the list of vault-specific resource provisioning attributes that may be changed:
  - spaceProvEF, spaceProvHC: Provisions the vault with the specified amount of extreme flash (EF) or high capacity (HC) storage space.
  - iopsProvEF, iopsProvHC: Provisions the vault with the specified number of IOPS from extreme flash (EF) or high capacity (HC) storage.
  - flashCacheProv: Provisions the vault with the specified amount of flash cache space.
  - flashLogProv: Boolean value (true or false) indicating whether the vault is provisioned with access to Exadata Smart Flash Log. The default value is true.
  - xrmemCacheProv: Provisions the vault with the specified amount of Exadata RDMA Memory Cache (XRMEM cache) space.

    Use this attribute only on systems with HC or EF storage, but not both. For systems with HC and EF storage, use the following media-specific XRMEM cache provisioning attributes.
  - xrmemCacheProvEF: Provisions the vault with the specified amount of XRMEM cache space associated with EF media.
  - xrmemCacheProvHC: Provisions the vault with the specified amount of XRMEM cache space associated with HC media.
- To specify an amount of space, you can:
  - Use an integer value representing the number of bytes. The maximum value is 4503599626321920.

  - Specify a space value using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.
  - Use unlimited to specify an unlimited amount of space.
- To specify a number of IOPS, you can:
  - Use an integer value. The maximum value is 4294967295.
  - Use unlimited to specify an unlimited number of IOPS.
- Specifying 0 (zero) as the value for a cache provisioning attribute (flashCacheProv, xrmemCacheProv, xrmemCacheProvEF, or xrmemCacheProvHC) effectively disables use of the corresponding cache for the vault.

Examples

Example 6-65 Set the Amount of Space Provisioned for a Vault

The following example shows setting the amount of provisioned HC storage space to 10 terabytes for the vault named my-data.

```
@> chvault my-data --attributes spaceProvHC=10T
```

6.2.6.2 mkvault

Create a vault.

Purpose

The mkvault command creates an Exascale vault.

Syntax

```
mkvault [@]vault [ --attributes attribute=value[,attribute=value]... ]
```

Command Options

The options for the mkvault command are:

- *vault*: Specifies the name of the vault.
- --attributes: Optionally specifies attributes to set.

  Use the describe mkvault command to see details about the vault attributes you can set. See also Describing Resources and Attributes.

Usage Notes

Note the following information when using this command:

- The default value for all resource provisioning attributes is effectively unlimited. Consequently, a vault that is created without any resource provisioning attribute settings has access to all of the resources in the Exascale cluster.

  Following is the list of vault-specific resource provisioning attributes that may be set:

  - spaceProvEF, spaceProvHC: Provisions the vault with the specified amount of extreme flash (EF) or high capacity (HC) storage space.

  - iopsProvEF, iopsProvHC: Provisions the vault with the specified number of IOPS from extreme flash (EF) or high capacity (HC) storage.
  - flashCacheProv: Provisions the vault with the specified amount of flash cache space.
  - flashLogProv: Boolean value (true or false) indicating whether the vault is provisioned with access to Exadata Smart Flash Log. The default value is true.
  - xrmemCacheProv: Provisions the vault with the specified amount of Exadata RDMA Memory Cache (XRMEM cache) space.

    Use this attribute only on systems with HC or EF storage, but not both. For systems with HC and EF storage, use the following media-specific XRMEM cache provisioning attributes.
  - xrmemCacheProvEF: Provisions the vault with the specified amount of XRMEM cache space associated with EF media.
  - xrmemCacheProvHC: Provisions the vault with the specified amount of XRMEM cache space associated with HC media.
- Prior to consumption, all provisioned resources are logically provisioned. For example, the spaceProvHC attribute specifies the maximum amount of HC storage space that the vault can consume. However, physical space is only consumed when data is written to files in the vault.
- To specify an amount of space, you can:
  - Use an integer value representing the number of bytes. The maximum value is 4503599626321920.
  - Specify a space value using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.
  - Use unlimited to specify an unlimited amount of space.
- To specify a number of IOPS, you can:
  - Use an integer value. The maximum value is 4294967295.
  - Use unlimited to specify an unlimited number of IOPS.
- If you specify a value for any provisioning attributes, then the vault is limited to use the provisioned resources from the corresponding storage pools and the other storage pools are effectively disabled.

  If you specify a space provisioning attribute (spaceProvEF, or spaceProvHC) without a corresponding IOPS provisioning attribute (iopsProvEF, or iopsProvHC), then the vault can consume the specified space and use unlimited IOPS in the associated storage pool.

  If you specify an IOPS provisioning attribute without a corresponding space provisioning attribute, then the vault can consume unlimited space in the associated storage pool but the I/O bandwidth is limited by the IOPS provisioning attribute.
- Specifying 0 (zero) as the value for a cache provisioning attribute (flashCacheProv, xrmemCacheProv, xrmemCacheProvEF, or xrmemCacheProvHC) effectively disables use of the corresponding cache for the vault.

Examples

Example 6-66 Create a Vault with Unlimited Storage

The following example shows creating a vault named my-data that has unlimited access to all of the resources in the Exascale cluster.

```
@> mkvault @my-data
```

Example 6-67 Create a Vault with Provisioned Storage

The following example shows creating a vault named my-data2 that is provisioned with 5 terabytes of HC storage space. The resulting vault is implicitly provisioned with unlimited IOPS.

```
@> mkvault @my-data2 --attributes spaceProvHC=5T
```

Example 6-68 Create a Vault with Provisioned Storage

The following example shows creating a vault named my-data3 that is provisioned to use EF and HC storage media. From the EF storage pool, the vault can consume 10 terabytes of storage space and unlimited IOPS. From the HC storage pool, the vault can consume unlimited storage space and 100000 IOPS.

```
@> mkvault @my-data3 --attributes spaceProvEF=5T,iopsProvHC=100000
```

6.2.6.3 rmvault

Delete a vault.

Purpose

The rmvault command deletes a vault.

Syntax

```
rmvault [@]vault [ --force ] [ --nowait ]
```

Command Options

The options for the rmvault command are:

- *vault*: Identifies the vault that is being deleted.
- --force: Also deletes all files in the vault.
- --nowait: Instead of waiting for the command to finish, the command returns immediately while the operation completes in the background.

Examples

Example 6-69 Delete a Vault

The following example shows deleting the vault named my-data.

```
@> rmvault @my-data
```

6.2.7 Dataset Management

This section contains references for the Exascale command line interface (ESCLI) commands that are associated with dataset management:

- lsdataset
  List datasets.

6.2.7.1 lsdataset

List datasets.

Purpose

The lsdataset command displays information about Exascale datasets. It can also list the files contained in a specific dataset.

Syntax

```
lsdataset [ dataset-ID ] [ -l ] [ --detail ]
      [ --attributes attribute[,attribute]... ]
      [ --filter filter[,filter]... ]
      [ --sort [-]attribute[,[-]attribute]... ]
      [ --count value ] [ --files [ --recursive ]]
```

Command Options

The options for the lsdataset command are:

- *dataset-ID*: The identifier for a dataset that you want to display information about.

  Each system-defined dataset has a unique composite identifier, which contains unique identifiers for the associated entities. The dataset identifier has one of the following formats:

  - @*Vault-name*: Identifies the vault-level dataset for a specific named vault.
  - @*Vault-name*:*GI-cluster-ID*: Identifies the dataset for a specific Oracle Grid Infrastructure (GI) cluster, which consumes storage space in the specified vault.
  - @*Vault-name*:*GI-cluster-ID*:*CDB-ID*: Identifies the dataset for a specific Oracle multitenant container database (CDB) that belongs to the specified GI cluster.
  - @*Vault-name*:*GI-cluster-ID*:*CDB-ID*:*PDB-ID*: Identifies the dataset for an Oracle pluggable database (PDB) that is associated with the specified CDB, GI cluster, and vault.

  The asterisk (*) can be used for wildcard searches. For example, @*Vault-name*:*GI-cluster-ID*:* displays information about all CDB datasets associated with the specified GI cluster and vault.

  If *dataset-ID* is not specified, the command displays information about all datasets in all vaults.

- -l: Displays output in a long, tabular format.
- --detail: Displays detailed output with additional attributes.
- --attributes: Identifies specific attributes to display in the output.

To see a list of all available attributes and their descriptions, use the `describe datasets` command. See also Describing Resources and Attributes.

- `--filter`: Specifies conditions for including items in the output.
- `--sort`: Sorts the output using the specified attributes.
- `--count`: Specifies the maximum number of items to display in the output.
- `--files`: Lists files that are in the specified dataset.
- `--recursive`: Also lists files that are in descendant datasets.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  – `yyyy-MM-dd"T"HH:mm:ss`

  – `yyyy-MM-dd` (Time is assumed to be 00:00 AM)

  – `HH:mm:ss` (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  Sizes can be specified using suffixes `K`, `KB`, `M`, `MB`, `G`, `GB`, `T`, `TB`. The suffix is not case-sensitive.

  For example, the following filter only includes objects created after 9 AM today:

  ```
  --filter createTime>9:00:00
  ```

- Sorting attributes are specified as: `[-]`*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with `-`.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

- You must specify a *dataset-ID* in conjunction with the `--files` option.

Examples

Example 6-70 Display All Datasets

You can use the `lsdataset` command with no other options to display a list of all datasets across all vaults.

```
@> lsdataset
id                                                                name
@my-data                                                          @my-data
@my-data:a5b4997a027d6f91ffd9729702ff6ec5                         @my-data/cluster1
@my-data:a5b4997a027d6f91ffd9729702ff6ec5:1427076301              @my-data/cluster1:mydb
@my-data:a5b4997a027d6f91ffd9729702ff6ec5:1427076301:1402749181   @my-data/cluster1:mydb.PDB$SEED
```

```
@my-data:a5b4997a027d6f91ffd9729702ff6ec5:1427076301:2132037342     @my-data/cluster1:mydb.CDB$ROOT
@my-data:a5b4997a027d6f91ffd9729702ff6ec5:1427076301:2164757665     @my-data/
cluster1:mydb.MYDB_PDB1
@vault2                                          @vault2
...
```

Example 6-71 Display Additional Dataset Attributes

```
@> lsdataset --attributes id,name,fileUsageHC,datasetUsageHC
id                                               name                      fileUsageHC  datasetUsageHC
@my-data                                         @my-data                  0.0000       3.4961G
@my-data:a5b4997a027d6f91ffd9729702ff6ec5                  @my-data/cluster1              294.5339M
3.4961G
@my-data:a5b4997a027d6f91ffd9729702ff6ec5:1427076301             @my-data/cluster1:mydb
25.2656M      3.2085G
@my-data:a5b4997a027d6f91ffd9729702ff6ec5:1427076301:1402749181     @my-data/
cluster1:mydb.PDB$SEED      650.6719M      650.6719M
@my-data:a5b4997a027d6f91ffd9729702ff6ec5:1427076301:2132037342     @my-data/
cluster1:mydb.CDB$ROOT       1.9056G       1.9056G
@my-data:a5b4997a027d6f91ffd9729702ff6ec5:1427076301:2164757665     @my-data/
cluster1:mydb.MYDB_PDB1     658.1719M      658.1719M
@vault2                                          @vault2                   0.0000      110.8750M
...
```

Example 6-72 Display Files in a Dataset

```
@> lsdataset --files @my-data:a5b4997a027d6f91ffd9729702ff6ec5:1427076301:2164757665
DATAFILE/mydb_pdb1_db.f
DATAFILE/mydb_pdb1_tmp.f
DATAFILE/mydb_pdb1_xdb.f
DATAFILE/mydb_pdb1_ax.f
```

Example 6-73 Using the Recursive Display Option

The following examples illustrate using the `--recursive` option.

Without the `--recursive` option, the following example shows the files belonging directly to the specified GI cluster dataset.

```
@> lsdataset --files @my-data:a5b4997a027d6f91ffd9729702ff6ec5
cluster1.ocr
cluster1/vfile1
```

With the `--recursive` option, the following example shows the files belonging to the specified GI cluster and all of the associated CDB and PDB datasets.

```
@> lsdataset --files --recursive @my-data:a5b4997a027d6f91ffd9729702ff6ec5
DATAFILE/mydb_pdb0_ax.f
dbs/xspfile.ora
DATAFILE/mydb_pdb0.xml
DATAFILE/mydb_pdb1_db.f
DATAFILE/t_ax1.f
DATAFILE/t_db1.f
CLUSTER1-A5B4997A027D6F91FFD9729702FF6EC5/MYDB/TEMPFILE/TEMP.OMF.2012709C
```

```
DATAFILE/t_xdb1.f
DATAFILE/mydb_pdb1_tmp.f
dbs/t_cf1.f
DATAFILE/mydb_pdb0_xdb.f
DATAFILE/mydb_pdb0_tmp.f
DATAFILE/t_undo1.f
cluster1.ocr
DATAFILE/mydb_pdb1_xdb.f
CLUSTER1-A5B4997A027D6F91FFD9729702FF6EC5/MYDB/PASSWORD/pwdMYDB.4ABC8604
cluster1/vfile1
DATAFILE/mydb_pdb0_db.f
DATAFILE/mydb_pdb1_ax.f
...
```

6.2.8 File Management

This section contains references for the Exascale command line interface (ESCLI) commands that are associated with file management:

- cd
  Change the current directory.
- chfile
  Change file attributes.
- clonefile
  Clone one or more files.
- extentmap
  Display storage extent information.
- getfile
  Copy a file from Exascale to the local file system.
- ls
  List files and vaults.
- lssnapshots
  List snapshots associated with a file.
- mkfile
  Create a file.
- putfile
  Copy a file from the local file system to Exascale.
- rmfile
  Delete a file.
- snapshotfile
  Create snapshots.

6.2.8.1 cd

Change the current directory.

Purpose

The cd command allows you to change the current directory for file operations. File operations are performed relative to the specified directory if absolute paths are not used.

Syntax

```
cd directory
```

Command Options

directory: Specifies the directory to change to. You can specify an absolute directory path or a directory path that is relative to the current directory.

Examples

Example 6-74   Change Directories using an Absolute Directory Path

You can change to a specified absolute directory path, such as @my-vault/dir.

```
@> cd @my-vault/dir
@my-vault/dir/>
```

Example 6-75   Change Directories using a Relative Directory Path

You can change directories using a specified relative directory path, such as ../dir2.

```
@my-vault/dir/> cd ../dir2
@my-vault/dir2/>
```

6.2.8.2 chfile

Change file attributes.

Purpose

The chfile command changes attributes of a file.

Syntax

```
chfile file-name --attributes attribute=value[,attribute=value]...
```

Command Options

The options for the chfile command are:

- *file-name*: Specifies the name of the file you are changing.
- --attributes: Specifies attributes to change.

  Use the describe chfile command to see details about the file attributes you can change. See also Describing Resources and Attributes.

Examples

Example 6-76   Change a File

The following example shows changing the size of the file named my-file to 100 KB.

```
@my-data/> chfile my-file --attributes size=100k
File altered.
```

```
@my-data/> ls -l
Total 1
100.0k  29 Jun 21:58   my-file
```

6.2.8.3 clonefile

Clone one or more files.

Purpose

The clonefile command allows you to create a writable clone of a file or group of files.

Syntax

```
clonefile source target [ source target ]... [ --exclude exclude-spec ]...
```

Command Options

The options for the clonefile command are:

- *source*: Specifies the name of the file or files that are the clone source.
- *target*: Specifies the name of the file or files that are the clone destination.
- --exclude: Specifies the name of the file or files that are excluded from the clone operation.

Usage Notes

Note the following information when using this command:

- You can use a wildcard (*) in the *source* to specify multiple source files, in which case the corresponding *target* must also contain a matching wildcard.
- All files in a clone operation must be in the same vault, otherwise and error is returned.
- Multiple *source* and *target* pairs are permitted. In this case, the source file specifications are considered in order, and only the first match is used.
- Files specified using the --exclude option are excluded from the clone operation. Such exclusions apply across all of the *source* and *target* pairs.
- If you specify a null (empty) string as a *target* value, no clones are created for the corresponding *source*. Specifying a null *target* is an alternative method of excluding files from the clone operation.

  The *source* and null *target* is considered in order along with all other *source* and *target* pairs, and only the first match is used.
- All clones created in the same operation are point-in-time consistent.

Examples

Example 6-77 Clone a File

The following example shows cloning a file. The source file @my-data/file1 is and the clone is @my-data/clone-file1.

```
@> clonefile @my-data/file1 @my-data/clone-file1
```

Example 6-78 Clone a Group of Files

The following example shows cloning a group of files using a wildcard. Assuming the existence of files named @my-data/file1, @my-data/file2, and so on, the example creates a clone of @my-data/file1 named @my-data/clone-dir/file1, @my-data/file2 named @my-data/clone-dir/file2, and so on.

```
@> clonefile @my-data/file* @my-data/clone-dir/file*
```

Example 6-79 Clone Multiple File Groups

The following example shows cloning multiple groups of files. The example clones @my-data/a* and @my-data/b*, with the resulting clones located under @my-data/clone/.

```
@> clonefile @my-data/a* @my-data/clone/a* @my-data/b* @my-data/clone/b*
```

Example 6-80 Ordering Significance

These examples show the significance of the order in which the *source* and *target* pairs are specified.

In the first command, the clones for files matching @my-data/a* are created in @my-data/clone/a, and the clones for the other files are created in @my-data/clone/other.

```
@> clonefile @my-data/a* @my-data/clone/a/a* @my-data/* @my-data/clone/other/*
```

In the following command, all of the clones are created in @my-data/clone/other because all of the files match @my-data/*. In this case, the second *source* and *target* pair is never used.

```
@> clonefile @my-data/* @my-data/clone/other/* @my-data/a* @my-data/clone/a/a*
```

Example 6-81 Clone a Group of Files with Exclusions

The following examples show cloning files in @my-data, except for those matching with @my-data/a* or @my-data/b*.

```
@> clonefile @my-data/* @my-data/clone/* --exclude @my-data/a* --exclude @my-data/b*
```

```
@> clonefile @my-data/a* "" @my-data/b* "" @my-data/* @my-data/clone/*
```

Example 6-82 Invalid Commands

The following examples show invalid commands. The first and second commands are invalid because multiple vaults are referenced. The final command is invalid because the clone destination does not contain a wildcard to match the source specification.

```
@> clonefile @my-data/a* @my-data-backup/a*
```

```
@> clonefile @my-data/a* @my-data/clone/a* @vault2/a* @vault2/clone/a*
```

```
@> clonefile @my-data/withwc* @my-data/clone/withoutwc
```

6.2.8.4 extentmap

Display storage extent information.

Purpose

The extentmap command enables Exascale cluster administrators to display storage extent information for Exascale file and vaults.

Syntax

```
extentmap [ --name name --file-offset offset [ --mirror mirror ]]
          [ --extId extent --target target ]
```

Command Options

The options for the extentmap command are:

- --name: A file or vault name for which you want to display extent information.

  Vault names are preceded with the @ symbol.

- --file-offset: Specifies the offset location for which to display information.

  The value is expressed in bytes and can be a single number or a range. Use this option when you specify a file name for translation.

- --mirror: Optionally specifies the mirror number for which to display information.

  For high (3-way) redundancy, the mirror number can be 0, 1, or 2.

  You can use this option when you specify a file name for translation.

- --extId: An extent identifier for which you want to display information.

  You can use this option to translate an extent identifier to a vault or file name.

- --target: Specifies the translation target; either vaults or files.

  You can use this option when you specify an extent identifier for translation.

Usage Notes

This command requires the cluster administrator privilege.

Examples

Example 6-83 Display Extent Information for a Vault

The following example shows how to display extent information for the specified vault.

```
@> extentmap --name @my-data
Vault Extent ID: c000_0000_003a:0000_0000_0000:00000000:00000000
```

Example 6-84 Display Extent Information for a Specific File Offset

The following example shows how to display extent information for a specified byte offset in a specified file.

```
@> extentmap --name @my-data/file1 --file-offset 123
extId: c000_0000_003a:8000_0000_000a:00000000:00000000
rpmId: 3222802720
extentEndOffset: 8388608
mirrorHealth: OK
translationPath:
o/192.0.2.1;192.0.2.1;192.0.2.1;192.0.2.1:41681/cell1_CD_exdisk6_EDSCELL1
mirrorNumber: 0

extId: c000_0000_003a:8000_0000_000a:00000000:00000000
rpmId: 3222802720
extentEndOffset: 8388608
mirrorHealth: OK
translationPath:
o/192.0.2.3;192.0.2.3;192.0.2.3;192.0.2.3:41681/cell3_CD_exdisk1_EDSCELL3
mirrorNumber: 1

extId: c000_0000_003a:8000_0000_000a:00000000:00000000
rpmId: 3222802720
extentEndOffset: 8388608
mirrorHealth: OK
translationPath:
o/192.0.2.6;192.0.2.6;192.0.2.6;192.0.2.6:41681/cell6_CD_exdisk9_EDSCELL6
mirrorNumber: 2
```

Example 6-85 Display Extent Information for a Specific Mirror

The following example shows how to display extent information for a specific extent mirror, using a specified byte offset in a specified file.

```
@> extentmap --name @my-data/file1 --file-offset 123 --mirror 1
extId: c000_0000_003a:8000_0000_000a:00000000:00000000
rpmId: 3222802720
extentEndOffset: 8388608
mirrorHealth: OK
translationPath:
o/192.0.2.3;192.0.2.3;192.0.2.3;192.0.2.3:41681/cell3_CD_exdisk1_EDSCELL3
mirrorNumber: 1
```

Example 6-86 Display Vault Name for a Specific Extent

The following example shows how to display the vault name for a specific extent.

```
@> extentmap --extId c000_0000_003a:8000_0000_000a:00000000:00000000 --target vaults
vaultName: my-data
SysEdsURL: 192.0.2.3;192.0.2.3;192.0.2.3;192.0.2.3:41681
```

Example 6-87   Display File Name for a Specific Extent

The following example shows how to display the file name for a specific extent.

```
@> extentmap --extId c000_0000_003a:8000_0000_000a:00000000:00000000 --target files
vaultName: my-data
fileName: file1
UserEdsURL: 192.0.2.3;192.0.2.3;192.0.2.3;192.0.2.3:41681
```

6.2.8.5 getfile

Copy a file from Exascale to the local file system.

Purpose

The getfile command copies a file from an Exascale vault to the local file system.

Syntax

```
getfile exa-file local-file
```

Command Options

The options for the getfile command are:

- *exa-file*: Specifies a file in an Exascale vault.
- *local-file*: Specifies a regular file location in the local file system.

Examples

Example 6-88   Get a File

The following example copies the Exascale file at @my-data/file1 to /tmp/file1.copy on the local file system.

```
@> getfile @my-data/file1 /tmp/file1.copy
```

6.2.8.6 ls

List files and vaults.

Purpose

The ls command displays information about Exascale files and vaults.

Syntax

```
ls [ name [ name ]... ] [ -l ] [ --detail ]
  [ --attributes attribute[,attribute]... ]
  [ --filter filter[,filter]... ]
  [ --sort [-]attribute[,[-]attribute]... ]
  [ --count value ] [ -t ] [ --include-being-deleted ]
```

Command Options

The options for the ls command are:

- *name*: A file or vault name that you want to display information for. Vault names are preceded with the @ symbol. The asterisk (*) can be used for wildcard searches. If not specified, then information is displayed about all files or vaults in the current level of the ESCLI file hierarchy.
- -l: Displays output in a long, tabular format.
- --detail: Displays detailed output with additional attributes.
- --attributes: Identifies specific attributes to display in the output.

  You can use this option to focus on a subset of attributes as an alternative to the --detail option.
- --filter: Specifies conditions for including items in the output.
- --sort: Sorts the output using the specified attributes.
- --count: Specifies the maximum number of items to display in the output.
- -t: Sorts files or vaults by ascending creation time.
- --include-being-deleted: Lists all files, including deleted files in the process of releasing their allocated storage.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: <*attribute*><*operator*><*value*>.

  Attributes can be file attributes, or vault attributes with the vault. prefix.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  – yyyy-MM-dd"T"HH:mm:ss
  – yyyy-MM-dd (Time is assumed to be 00:00 AM)
  – HH:mm:ss (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  File sizes can be specified using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.

  For example, the following filter only includes files created after 1 PM today and with a size greater than 10 megabytes:

  ```
  --filter createTime>13:00:00,size>10M
  ```

- Sorting attributes are specified as: [-]*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with -.

For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

```
--sort -name,createTime
```

Examples

Example 6-89 Display All Vaults

If you use the `ls` command at the root level, you get a listing of all vaults.

```
@> ls
my-data
vault2
```

Example 6-90 Display Files in a Vault

You can specify the name of a vault followed by the slash character (/) to list all the files within the vault.

```
@> ls @my-data/
x
y
z1
z2
z3
```

Example 6-91 Display File Information in a Long Format

If you use the `-l` option, the output includes additional information about each file or vault.

```
@my-data/> ls -l
Total 5
  10.0M   05 Jan 12:59 x
  19.5k   05 Jan 13:08 y
   5.0k   05 Jan 13:09 z1
  10.0M   05 Jan 13:23 z2
  20.0G   05 Jan 13:14 z3
```

Example 6-92 Sorting the Output of the ls Command

The following example shows the files in the vault `my-data`, with the most recent appearing first.

```
@my-data/> ls --sort -createTime
z2
z3
z1
y
x
```

Example 6-93 Listing Files Using a Wildcard Search

You can use the * symbol as a wildcard when specifying the files to list.

```
@my-data> ls z*
z1
z2
z3
```

Example 6-94 Display Detailed File Information

If you use the --detail option, the output includes detailed information about the files and vaults. This example shows the detailed output for the file x in the vault my-data.

```
@> ls @my-data/x --detail
name                x
size              0
createTime          2019-01-05 12:59:57 GMT
fileType            8
vault.name          my-data
vault.createTime      2018-11-08 22:41:07 GMT
```

Example 6-95 Display Specific Attributes for Files or Vaults

The following example shows the size, name, and vault name attributes for the files in the my-data vault.

```
@my-data/> ls -l --attributes size,name,vault.name
Total 5
     0 x my-data
  19.5k y my-data
   5.0k z1 my-data
  10.0M z2 my-data
  20.0G z3 my-data
```

6.2.8.7 lssnapshots

List snapshots associated with a file.

Purpose

The lssnapshots command displays information about snapshots that are associated with the specified file.

Syntax

```
lssnapshots file-name [ --all | --tree ] [ -l ] [ --detail ]
        [ --attributes attribute[,attribute]... ]
        [ --filter filter[,filter]... ] [ --count value ]
```

Command Options

The options for the lssnapshots command are:

- *file-name*: Identifies the file that is the subject of the command. The specified file can be a snapshot, clone, or regular file.
- --all: Lists information about all files, clones, and snapshots belonging to the snapshot tree that contains the specified file. The output is presented in a series of lists. The first list contains the files and clones in the snapshot tree. The remaining output lists snapshots that are grouped by their source file.
- --tree: Lists information about all files, clones, and snapshots belonging to the snapshot tree that contains the specified file. The output is arranged in a graphical tree.
- -l: Displays output in a long, tabular format.
- --detail: Displays detailed output with additional attributes.
- --attributes: Identifies specific attributes to display in the output.

  You can use this option to focus on a subset of attributes as an alternative to the --detail option.
- --filter: Specifies conditions for including items in the output.
- --sort: Sorts the output using the specified attributes.
- --count: Specifies the maximum number of items to display in the output.

Usage Notes

Note the following information when using this command:

- If you do not specify the command options --all or --tree, then the command displays information about immediate snapshots of the specified source file.
- You cannot use any other command options in conjunction with --all or --tree.
- Filters are specified as: *<attribute><operator><value>*.

  Attributes can be file attributes, or vault attributes with the vault. prefix. Multiple filter conditions are delimited by commas.

  The allowed operators are =, >=, <=, >, and <.

  Dates can be specified using the following formats:

  – yyyy-MM-dd"T"HH:mm:ss

  – yyyy-MM-dd (Time is assumed to be 00:00 AM)

  – HH:mm:ss (Date is assumed to be today)

  Any of these formats can be followed by a timezone specification.

  File sizes can be specified using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.

  For example, to filter the results to show only snapshots created after noon today, with a size greater than 10 megabytes, you could use the following:

  ```
  --filter creationTime>12:00:00,size>10M
  ```
- Output is ordered according to the snapshot creation time, starting with the most recent.

Examples

Example 6-96   List Basic Snapshot Information

The following example shows how to list information about all of the snapshots that are based on the specified file.

```
@> lssnapshots @my-data/my-file
```

Example 6-97   List Detailed Snapshot Information

The following example shows how to list detailed information about all of the snapshots that are based on the specified file.

```
@> lssnapshots @my-data/my-file --detail
```

Example 6-98   List Snapshot Information with Specific Attributes

The following example shows how to list specific attributes about all of the snapshots that are based on the specified file.

```
@> lssnapshots @my-data/my-file --detail --attributes name,vault.name,createTime,vault.createTime
```

Example 6-99   Filter the List of Snapshots

The following example shows how to specify a filter that constrains the command output. In the example, the output is limited to the snapshots that were created after the specified date.

```
@> lssnapshots @my-data/my-file --filter createTime>2020-01-01
```

Example 6-100   List the Latest Snapshots

The following example shows how to specify a count that constrains the command output. In the example, the output is limited to the 5 most recently created snapshots.

```
@> lssnapshots @my-data/my-file --count 5
```

Example 6-101   Display a Snapshot Tree

The command output in this example assumes a group of related files created using the following command sequence:

```
@my-data/> putfile somelocalfile file1
Putting file somelocalfile to @my-data/file1
Success.

@my-data/> snapshotfile file1 snap1_of_file1
Success.

@my-data/> clonefile file1 clone1
Success.

@my-data/> snapshotfile file1 snap2_of_file1
Success.
```

```
@my-data/> snapshotfile clone1 snap1_of_clone1
Success.
```

The following example shows how to display all files in a snapshot tree using the --tree command option.

```
@> lssnapshots @my-data/clone1 --tree

+---- [1] @my-data/snap1_of_file1
+---- [2] @my-data/?8000_0000_000a:00000001
|
|`---+---- [4] @my-data/snap1_of_clone1
|    +---- [6] @my-data/clone1
|
+---- [3] @my-data/snap2_of_file1
+---- [5] @my-data/file1
```

Note that the command output includes an internal snapshot (@my-data/?8000_0000_000a:00000001), which was implicitly created as part of the cloning operation to create @my-data/clone1.

Example 6-102   List all Files in a Snapshot Tree

The following example shows how to display all files in a snapshot tree using the --all command option. This example assumes the same scenario as the previous example.

```
@> lssnapshots @my-data/clone1 --all

---FILE/CLONES:---

2022-05-02 06:14:41 UTC @my-data/file1
2022-05-02 06:15:15 UTC @my-data/clone1

---SNAPSHOTS:---

@my-data/file1
  2022-05-02 06:15:06 UTC @my-data/snap1_of_file1
  2022-05-02 06:15:15 UTC @my-data/?8000_0000_000a:00000001
  2022-05-02 06:15:24 UTC @my-data/snap2_of_file1

@my-data/clone1
  2022-05-02 06:15:45 UTC @my-data/snap1_of_clone1
```

6.2.8.8 mkfile

Create a file.

Purpose

The mkfile command creates a file within a vault.

Syntax

```
mkfile file-name [ --template template ]
   [ --attributes attribute=value[,attribute=value]... ]
```

Command Options

The options for the mkfile command are:

- *file-name*: Specifies the name of the file you are creating. The file name cannot contain wildcard characters. Only one file name can be specified.
- --template: Creates the file using the specified template.
- --attributes: Optionally specifies attributes to change.

  Use the describe mkfile command to see details about the file attributes you can set. See also Describing Resources and Attributes.

Usage Notes

Note the following information when using this command:

- You can set the file size by specifying the size attribute.

  File sizes can be specified using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.

  Regardless of the file size setting, storage space is physically materialized only when the file contents is written. So, after the mkfile command, the file initially consumes no physical storage space. The file only consumes physical storage space when data is written to it.
- If you set any of the file storage attributes (contentType, mediaType, and redundancy), the specified value overrides the setting in the default template.

  For a description of the file storage attributes, see File Storage Attributes.
- You can set the Exascale file type by specifying the fileType attribute.

  Every Exascale file type is associated with an Exascale template. So, you can view a complete list of the Exascale file types by using the ESCLI lstemplate command. For example:

  ```
  @> lstemplate --cluster
  ```

  ```
  @> lstemplate --vault vault2
  ```

Examples

Example 6-103 Create a File in a Vault

The following example shows creating the file named my-file within the vault my-data.

```
@my-data/> mkfile my-file --attributes size=20k
File created.

@my-data/> ls -l
  20.0k   08 Jan 15:53 my-file
```

Related Topics

- About Exascale File Storage Attributes
- lstemplate
  List file templates.

6.2.8.9 putfile

Copy a file from the local file system to Exascale.

Purpose

The putfile command copies a file from the local file system to an Exascale vault.

Syntax

```
putfile local-file exa-file
```

Command Options

The options for the putfile command are:

- *local-file*: Specifies a regular file in the local file system.
- *exa-file*: Specifies a file location in an Exascale vault.

Examples

Example 6-104 Put a File

The following example copies the local file at /tmp/file1 to @my-data/file1.copy on Exascale storage.

```
@> putfile /tmp/file1 @my-data/file1.copy
```

6.2.8.10 rmfile

Delete a file.

Purpose

The rmfile command deletes a file in an Exascale vault.

Syntax

```
rmfile file-name [ --force ] [ --nowait ] [ --with-snapshots ]
```

Command Options

The options for the rmfile command are:

- *file-name*: Identifies the file to be deleted.
- --force: Deletes the specified file even if it is in use.
- --nowait: Instead of waiting for the command to finish, the command returns immediately while the operation completes in the background.
- --with-snapshots: Also deletes snapshots associated with the specified file.

Examples

Example 6-105 Delete a File

The following example deletes the file at @my-data/file1.

```
@> rmfile @my-data/file1
```

6.2.8.11 snapshotfile

Create snapshots.

Purpose

The snapshotfile command creates a snapshot of a file or group of files.

Syntax

```
snapshotfile source1 target1 [ sourceN targetN ]...
         [ --exclude exclude-spec ]...
```

Command Options

The options for the snapshotfile command are:

- *source1-N*: Specifies the name of the file or files that are the snapshot source.
- *target1-N*: Specifies the name of the file or files that are the snapshot destination.
- --exclude: Specifies the name of the file or files that are excluded from the snapshot operation.

Usage Notes

Note the following information when using this command:

- You can use a wildcard (*) in the *source* to specify multiple source files, in which case the corresponding *target* must also contain a matching wildcard.
- All files in a snapshot operation must be in the same vault, otherwise and error is returned.
- Multiple *source* and *target* pairs are permitted. In this case, the source file specifications are considered in order, and only the first match is used.
- Files specified using the --exclude option are excluded from the snapshot operation. Such exclusions apply across all of the *source* and *target* pairs.
- If you specify a null (empty) string as a *target* value, no snapshots are created for the corresponding *source*. Specifying a null *target* is an alternative method of excluding files from the snapshot operation.

  The *source* and null *target* is considered in order along with all other *source* and *target* pairs, and only the first match is used.
- All snapshots created in the same operation are point-in-time consistent.

Examples

Example 6-106 Snapshot a File

This example shows creating a snapshot. The source file @my-data/file1 is and the snapshot is @my-data/snap-file1.

```
@> snapshotfile @my-data/file1 @my-data/snap-file1
```

Example 6-107 Snapshot a Group of Files

This example shows creating snapshots for a group of files using a wildcard. Assuming the existence of files named @my-data/file1, @my-data/file2, and so on, the example creates a snapshot of @my-data/file1 named @my-data/snap-dir/file1, @my-data/file2 named @my-data/snap-dir/file2, and so on.

```
@> snapshotfile @my-data/file* @my-data/snap-dir/file*
```

Example 6-108 Snapshot Multiple File Groups

This example shows creating snapshots for multiple groups of files using wildcards. The source file groups are @my-data/a* and @my-data/b*, with the resulting snapshots located under @my-data/snap/.

```
@> snapshotfile @my-data/a* @my-data/snap/a* @my-data/b* @my-data/snap/b*
```

Example 6-109 Ordering Significance

These examples show the significance of the order in which the *source* and *target* pairs are specified.

In the first command, the snapshots for files matching @my-data/a* are created in @my-data/snap/a, and the snapshots for the other files are created in @my-data/snap/other.

```
@> snapshotfile @my-data/a* @my-data/snap/a/a* @my-data/* @my-data/snap/other/*
```

In the following command, all of the snapshots are created in @my-data/snap/other because all of the files match @my-data/*. In this case, the second *source* and *target* pair is never used.

```
@> snapshotfile @my-data/* @my-data/snap/other/* @my-data/a* @my-data/snap/a/a*
```

Example 6-110 Snapshot a Group of Files with Exclusions

The following examples show how to snapshot files in @my-data, except for those matching with @my-data/a* or @my-data/b*.

```
@> snapshotfile @my-data/* @my-data/snap/* --exclude @my-data/a* --exclude @my-data/b*
```

```
@> snapshotfile @my-data/a* "" @my-data/b* "" @my-data/* @my-data/snap/*
```

Example 6-111 Invalid Commands

The following examples show invalid commands. The first and second commands are invalid because multiple vaults are referenced. The final command is invalid because the snapshot destination does not contain a wildcard to match the source specification.

```
@> snapshotfile @my-data/a* @my-data-backup/a*
```

```
@> snapshotfile @my-data/a* @my-data/snap/a* @vault2/a* @vault2/snap/a*
```

```
@> snapshotfile @my-data/withwc* @my-data/snap/withoutwc
```

6.2.9 Template Management

This section contains references for the Exascale command line interface (ESCLI) commands that are associated with template management:

- chtemplate
  Change an attribute for an existing template.
- lstemplate
  List file templates.
- mktemplate
  Create a file template.
- rmtemplate
  Delete a file template.

6.2.9.1 chtemplate

Change an attribute for an existing template.

Purpose

The chtemplate command changes an attribute for an existing Exascale file template.

Syntax

```
chtemplate { --file-type file-type | --name template-name }
        [ --vault vault | --cluster ]
        [ --content-type content-type ]
        [ --media-type media-type ]
        [ --redundancy redundancy ]
```

Command Options

Specify one or more of the following chtemplate command options to identify the templates that are the subject of the operation:

- --file-type: Alters a template associated with the specified file type.
- --name: Alters a user-defined template having the specified name.
- --vault: Alters a template associated with the specified vault.

- --cluster: Alters a template associated with the cluster.

Specify one or more of the following File Storage Attributes, which are associated with the template:

- --media-type: Specifies the physical media type used to store the file. Exascale uses this attribute to place the file in a storage pool that utilizes the specified media type. Possible values are:
  - HC: Identifies high capacity storage, using hard disk drives (HDDs) on high-performance Exadata storage servers.
  - EF: Identifies extreme flash storage, using low-latency, high-throughput flash devices.
- --redundancy: Specifies the number of data copies that are maintained. Currently, the only permitted value is:
  - high: Indicates three mirrored copies of the file data.
- --content-type: Specifies the type of content in the file. Exascale internally uses this attribute to place file extents on physically separate devices in a manner that maximizes availability in the event of a failure. Possible values are:
  - DATA: Principally associated with user data.
  - RECO: Primarily for data used in backup and recovery operations.

Examples

Example 6-112 Change a User-Defined Cluster-Wide Template

In this example, the content type is set to DATA for the user-defined template named T1. The template is associated with the cluster, as neither --vault or --cluster are specified, and the current working directory in the ESCLI session is the root directory.

```
@> chtemplate --name T1 --content-type DATA
```

Example 6-113 Change a User-Defined Vault-Specific Template

In this example, the media type is set to HC and the redundancy is set to high for the user-defined template named T2. The template is associated with the vault named vault2, as neither --vault or --cluster are specified, and the current working directory in the ESCLI session is inside vault2.

```
@vault2> chtemplate --name T2 --media-type HC --redundancy high
```

Example 6-114 Change a User-Defined Vault-Specific Template

In this example, the media type, content type, and redundancy are all set for the user-defined template named T3 that is associated with the vault named vault2.

```
@> chtemplate --name T3 --vault vault2 --content-type DATA --media-type HC --redundancy high
```

Example 6-115 Change a User-Defined Cluster-Wide Template

In this example, the media type, content type, and redundancy are all set for the cluster-wide user-defined template named T4. The --cluster option overrides the fact that the current working directory in the ESCLI session is inside vault2.

```
@vault2> chtemplate --name T4 --cluster --content-type DATA --media-type HC --redundancy high
```

Example 6-116 Change a Template for a Specific Vault and File Type

This example changes the template associated with the DATAFILE file type and the vault named vault2.

```
@> chtemplate --file-type DATAFILE --vault vault2 --content-type DATA --media-type HC --redundancy high
```

Example 6-117 Change a Cluster-Wide Template for a Specific File Type

This example changes the cluster-wide template associated with the DATAFILE file type.

```
@> chtemplate --file-type DATAFILE --cluster --content-type DATA --media-type HC --redundancy high
```

Related Topics

- About Exascale File Storage Attributes

6.2.9.2 lstemplate

List file templates.

Purpose

The lstemplate command displays information about Exascale file templates that are associated with the vault or the Exascale cluster.

Syntax

```
lstemplate [ template-name ]
        [ --vault vault [ --vault-level-only ] | --cluster ]
        [ -l ] [ --detail ] [ --default ]
        [ --attributes attribute[,attribute]... ]
        [ --filter filter[,filter]... ]
        [ --sort [-]attribute[,[-]attribute]... ]
```

Command Options

The options for the lstemplate command are:

- *template-name*: Optionally limits the output to templates that match the specified template name.
- --vault: Lists templates associated with the specified vault.
- --vault-level-only: List only the vault-specific templates associated with the specified vault.
- --cluster: Lists templates associated with the cluster.
- -l: Displays output in a long, tabular format.

- `--detail`: Displays detailed output with additional attributes.
- `--default`: List default configuration details for system-defined templates (`templateType=systemType` or `templateType=oracleFileType`).
- `--attributes`: Identifies specific attributes to display in the output.

  You can use this option to focus on a subset of attributes as an alternative to the `--detail` option.
- `--filter`: Specifies conditions for including items in the output.
- `--sort`: Sorts the output using the specified attributes.

Usage Notes

Note the following information when using this command:

- If `--vault` is specified without `--vault-level-only`, then the output contains all templates affecting the specified vault, including the vault-specific templates and any cluster-level templates not overridden by vault-specific templates.
- If neither `--vault` nor `--cluster` is specified, and the user entered a vault in the ESCLI session, then the output contains all templates affecting the specified vault, including the vault-specific templates and any cluster-level templates not overridden by vault-specific templates.
- If neither `--vault` nor `--cluster` is specified, and the user hasn't entered a vault in the ESCLI session (ESCLI prompt shows `@>`), then templates associated with the cluster are listed.
- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  For example, the following filter only includes objects created after 9 AM today:

  ```
  --filter createTime>9:00:00
  ```
- Sorting attributes are specified as: `[-]`*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with `-`.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

Examples

Example 6-118 List Cluster-Wide Templates

These examples list all templates associated with the cluster.

```
@> lstemplate
```

```
@vault2> lstemplate --cluster
```

Example 6-119 List Effective Vault-Specific Templates

These examples list the effective templates associated with the vault vault2. The output includes templates associated with the vault, and templates associated with the cluster that are not overridden by vault-specific templates.

```
@vault2> lstemplate
```

```
@> lstemplate --vault vault2
```

Example 6-120 List Only Templates Affecting a Vault

This examples lists only the vault-specific templates affecting files in the vault vault2.

```
@> lstemplate --vault vault2 --vault-level-only
```

Example 6-121 List a Specific Cluster-Wide Template

This example displays detailed information about the cluster-wide template named DATAFILE.

```
@> lstemplate --detail DATAFILE
```

Example 6-122 List Default Configuration Information for a Specific Cluster-Wide Template

This example displays default configuration information for the cluster-wide template named DATAFILE.

```
@> lstemplate --default DATAFILE
```

Example 6-123 List Specific Cluster-Wide Templates

This example displays detailed information about the cluster-wide templates with names ending with FILE, whose content type is DATA, and sort the list by names (ascending) and media types (descending).

```
@> lstemplate --detail *FILE --sort name,-mediaType --filter contentType=DATA
```

6.2.9.3 mktemplate

Create a file template.

Purpose

The mktemplate command creates an Exascale file template.

Syntax

```
mktemplate { --file-type file-type | --name template-name }
        [ --vault vault | --cluster ]
        --content-type content-type
```

```
      --media-type media-type
      --redundancy redundancy
```

Command Options

Specify the following mktemplate command options to identify the template and its scope:

- --file-type: Create a template associated with the specified file type. If you specify this option, the template name is the same as the file type.
- --name: Create a user-defined template having the specified name.
- --vault: Create a template associated with the specified vault.
- --cluster: Create a template associated with the cluster.

Specify the following File Storage Attributes, which are associated with the template:

- --media-type: Specifies the physical media type used to store the file. Exascale uses this attribute to place the file in a storage pool that utilizes the specified media type. Possible values are:
  - HC: Identifies high capacity storage, using hard disk drives (HDDs) on high-performance Exadata storage servers.
  - EF: Identifies extreme flash storage, using low-latency, high-throughput flash devices.
- --redundancy: Specifies the number of data copies that are maintained. Currently, the only permitted value is:
  - high: Indicates three mirrored copies of the file data.
- --content-type: Specifies the type of content in the file. Exascale internally uses this attribute to place file extents on physically separate devices in a manner that maximizes availability in the event of a failure. Possible values are:
  - DATA: Principally associated with user data.
  - RECO: Primarily for data used in backup and recovery operations.

Usage Notes

Note the following information when using this command:

- You can define the scope of the template by specifying the --vault option or the --cluster option. You cannot specify both options simultaneously.

  If neither option is specified, then the scope of the template is inferred from the current working directory in the ESCLI session. If the current working directory in the ESCLI session is the root directory, then the template is associated with the cluster. Otherwise, the template is associated with the vault that is referenced in the current working directory.
- You must define the template as a user template by specifying the --name option, or you must associate the template with a file type by using the --file-type option. You cannot specify both options simultaneously.
- You can view a list of Exascale file types by using the ESCLI lstemplate command. For example:

  ```
  @> lstemplate --cluster
  ```

  ```
  @> lstemplate --vault vault2 --all
  ```

Examples

Example 6-124 Create a User-Defined Cluster-Wide Template

This example shows creating a user-defined template named T1. The template is associated with the cluster, as neither --vault or --cluster are specified, and the current working directory in the ESCLI session is the root directory.

```
@> mktemplate --name T1 --content-type DATA --media-type HC --redundancy high
```

Example 6-125 Create a User-Defined Vault-Specific Template

This example shows creating a user-defined template named T2. The template is associated with the vault named vault2, as neither --vault or --cluster are specified, and the current working directory in the ESCLI session is inside vault2.

```
@vault2> mktemplate --name T2 --content-type DATA --media-type HC --redundancy high
```

Example 6-126 Create a User-Defined Vault-Specific Template

This example shows creating a user-defined template named T3 that is associated with the vault named vault2.

```
@> mktemplate --name T3 --vault vault2 --content-type DATA --media-type HC --redundancy high
```

Example 6-127 Create a User-Defined Cluster-Wide Template

This example shows creating a user-defined cluster-wide template named T4. The --cluster option overrides the fact that the current working directory in the ESCLI session is inside vault2.

```
@vault2> mktemplate --name T4 --cluster --content-type DATA --media-type HC --redundancy high
```

Example 6-128 Create a Template for a Specific Vault and File Type

This example shows creating a template associated with the DATAFILE file type and the vault named vault2.

```
@> mktemplate --file-type DATAFILE --vault vault2 --content-type DATA --media-type HC --redundancy high
```

Example 6-129 Create a Cluster-Wide Template for a Specific File Type

This example shows creating a cluster-wide template associated with the DATAFILE file type.

```
@> mktemplate --file-type DATAFILE --cluster --content-type DATA --media-type HC --redundancy high
```

Related Topics

- About Exascale File Storage Attributes
- lstemplate
  List file templates.

6.2.9.4 rmtemplate

Delete a file template.

Purpose

The rmtemplate command deletes an Exascale file template that is associated with the vault or the cluster.

Syntax

```
rmtemplate [ --vault vault | --cluster ]
        { --file-type file-type | --name template-name }
```

Command Options

Specify one or more of the following rmtemplate command options to identify the template to delete:

- --vault: Specifies a template associated with the specified vault.
- --cluster: Specifies a template associated with the cluster.
- --file-type: Specifies a template associated with the specified file type.
- --name: Specifies a user-defined template having the specified name.

Usage Notes

Note the following information when using this command:

- File type templates associated with the cluster cannot be deleted.

Examples

Example 6-130 Delete a User-Defined Cluster-Wide Template

This example shows deleting a user-defined template named T1. The template is associated with the cluster, as neither --vault or --cluster are specified, and the current working directory in the ESCLI session is the root directory.

```
@> rmtemplate --name T1
```

Example 6-131 Delete a User-Defined Vault-Specific Template

This example shows deleting a user-defined template named T2. The template is associated with the vault named vault2, as neither --vault or --cluster are specified, and the current working directory in the ESCLI session is inside vault2.

```
@vault2> rmtemplate --name T2
```

Example 6-132 Delete a User-Defined Vault-Specific Template

This example shows deleting a user-defined template named T3 that is associated with the vault named vault2.

```
@> rmtemplate --name T3 --vault vault2
```

Example 6-133 Delete a User-Defined Cluster-Wide Template

This example shows deleting a user-defined cluster-wide template named T4. The --cluster option overrides the fact that the current working directory in the ESCLI session is inside vault2.

```
@vault2> mktemplate --name T4 --cluster
```

Example 6-134 Delete a Template for a Specific Vault and File Type

This example shows deleting a template associated with the DATAFILE file type and the vault named vault2.

```
@> rmtemplate --file-type DATAFILE --vault vault2
```

6.2.10 Resource Profile Management

This section contains references for the Exascale command line interface (ESCLI) commands that are associated with resource profile management:

- chresourceprofile
  Change a resource profile associated with a vault.
- lsresourceprofile
  List resource profiles.
- mkresourceprofile
  Create a resource profile.
- rmresourceprofile
  Delete a resource profile.

6.2.10.1 chresourceprofile

Change a resource profile associated with a vault.

Purpose

The chresourceprofile command changes the attribute settings for an existing resource profile that is associated with an Exascale vault.

Syntax

```
chresourceprofile [vault-name/]resource-profile-name
         --attributes attribute=value[,attribute=value]...
```

Command Options

The options for the chresourceprofile command are:

- *resource-profile-name*: Identifies the resource profile being changed.
- *vault-name*: Identifies the vault that the resource profile is associated with.

  If not specified, then the current working directory is used to identify the vault.
- --attributes: Specifies attributes to change. The following list outlines all of the available attributes:
  - iopsShareEF, iopsShareHC: Specifies the relative share of I/O bandwidth (IOPS) from extreme flash (EF) or high capacity (HC) storage that is available to each client associated with the resource profile. Each client's share is relative to all other client shares associated with the corresponding media type. A higher share value implies higher priority. The range of valid values is 1-100, and the default value is 1.
  - iopsLimitEF, iopsLimitHC: Specifies the upper limit of the I/O bandwidth (IOPS) from extreme flash (EF) or high capacity (HC) storage that is available to each client associated with the resource profile. Each value represents a fraction out of 10000. The range of valid values is 1-10000, and the default value is 10000 (effectively unlimited).
  - enableFlashCache, enableXrmemCache, enableFlashLog: Enables or disables use of the flash cache, XRMEM cache, or flash log for clients associated with the resource profile. Each value is Boolean, and the default is true (enabled).
  - flashCacheMin, xrmemCacheMin: Specifies the guaranteed minimum fraction of flash cache or XRMEM cache space available to each client associated with the resource profile. The range of valid values is 0-10000, and the default value is 0 (no set minimum). For each cache type, if the sum of all values across all resource profiles exceeds 10000, then all the values are proportionally scaled down. Each value applies only when the corresponding cache is enabled in the resource profile (for example, enableFlashCache=true).
  - flashCacheMax, xrmemCacheMax: Specifies the maximum fraction of flash cache or XRMEM cache space available to each client associated with the resource profile. The range of valid values is 0-10000, and the default value is 10000. Each value applies only when the corresponding cache is enabled in the resource profile (for example, enableFlashCache=true).

Usage Notes

In addition to regular user-defined resource profiles, you can also modify the system-reserved resource profile named $UNASSIGNED. All Exascale clients not explicitly associated with a resource profile are automatically governed by the $UNASSIGNED profile. The $UNASSIGNED resource profile contains only two modifiable attributes:

- flashCacheMax: Specifies the maximum fraction of flash cache shared by clients associated with the $UNASSIGNED profile. The range of valid values is 0-10000. The default value is 10000.
- xrmemCacheMax: Specifies the maximum fraction of XRMEM cache shared by clients associated with the $UNASSIGNED profile. The range of valid values is 0-10000. The default value is 10000.

All Exascale clients governed by the $UNASSIGNED profile share the corresponding cache resources. The behavior differs from regular resource profiles, where each application of the resource profile defines the resource allocation for one associated client.

You must create the $UNASSIGNED resource profile before you can modify it. If the $UNASSIGNED resource profile does not exist, all unassigned Exascale clients share any unassigned flash cache space and XRMEM cache space. If there is no unassigned space to share, the system automatically reserves 5% of the cache space for unassigned Exascale clients.

Examples

Example 6-135 Change a Resource Profile

In this example, the iopsShareHC attribute is set to 1100 for the silver resource profile in conjunction with the vault2 vault.

```
@> chresourceprofile vault2/silver --attributes iopsShareHC=1100
```

Example 6-136 Change a Resource Profile

In this example, the enableFlashLog attribute is set to false for the silver resource profile in conjunction with the vault2 vault. In this example, the vault association is not specified in the command but is derived from the current working directory in the ESCLI session.

```
@vault2> chresourceprofile silver --attributes enableFlashLog=false
```

6.2.10.2 lsresourceprofile

List resource profiles.

Purpose

The lsresourceprofile command displays information about Exascale resource profiles that are associated with a vault.

Syntax

```
lsresourceprofile [ [vault-name/]resource-profile-name
                 [,[vault-name/]resource-profile-name ]... ]
              [ -l ] [ --detail ]
              [ --attributes attribute[,attribute]... ]
              [ --filter filter[,filter]... ]
              [ --sort [-]attribute[,[-]attribute]... ]
```

Command Options

The options for the lsresourceprofile command are:

- *resource-profile-name*: Identifies a specific resource profile name.
- *vault-name*: Identifies a specific vault.

  If not specified, then the current working directory is used to specify the vault.
- -l: Displays output in a long, tabular format.
- --detail: Displays detailed output with additional attributes.
- --attributes: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the describe resourceprofiles command. See also Describing Resources and Attributes.
- --filter: Specifies conditions for including items in the output.

- --sort: Sorts the output using the specified attributes.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  For example, the following filter only includes objects created after 9 AM today:

  ```
  --filter createTime>9:00:00
  ```

- Sorting attributes are specified as: [-]*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with -.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

Examples

Example 6-137 List Resource Profiles

This example lists the resource profiles associated with the vault named vault2.

```
@> lsresourceprofile --filter vault=vault2
```

Example 6-138 List Resource Profiles

This example lists the resource profiles associated with the vault named vault2. In this example, the vault association is not specified in the command but is derived from the current working directory in the ESCLI session.

```
@vault2> lsresourceprofile
```

Example 6-139 List a Specific Resource Profile

This example shows two ways to display detailed information about the resource profile named silver that is associated with the vault named vault2.

```
@> lsresourceprofile vault2/silver --detail
```

```
@> lsresourceprofile silver --filter vault=vault2 --detail
```

6.2.10.3 mkresourceprofile

Create a resource profile.

Purpose

The mkresourceprofile command creates a resource profile.

Syntax

```
mkresourceprofile [vault-name/]resource-profile-name
             [ --attributes attribute=value[,attribute=value]... ]
```

Command Options

The options for the mkresourceprofile command are:

- *resource-profile-name*: Specifies the name for the resource profile being created.
- *vault-name*: Specifies the vault that the resource profile is associated with.

  If not specified, then the current working directory is used to specify the vault.
- --attributes: Optionally specifies values for attributes of the resource profile. The following list outlines all of the available attributes:
  - iopsShareEF, iopsShareHC: Specifies the relative share of I/O bandwidth (IOPS) from extreme flash (EF) or high capacity (HC) storage that is available to each client associated with the resource profile. Each client's share is relative to all other client shares associated with the corresponding media type. A higher share value implies higher priority. The range of valid values is 1-100, and the default value is 1.
  - iopsLimitEF, iopsLimitHC: Specifies the upper limit of the I/O bandwidth (IOPS) from extreme flash (EF) or high capacity (HC) storage that is available to each client associated with the resource profile. Each value represents a fraction out of 10000. The range of valid values is 1-10000, and the default value is 10000 (effectively unlimited).
  - enableFlashCache, enableXrmemCache, enableFlashLog: Enables or disables use of the flash cache, XRMEM cache, or flash log for clients associated with the resource profile. Each value is Boolean, and the default is true (enabled).
  - flashCacheMin, xrmemCacheMin: Specifies the guaranteed minimum fraction of flash cache or XRMEM cache space available to each client associated with the resource profile. The range of valid values is 0-10000, and the default value is 0 (no set minimum). For each cache type, if the sum of all values across all resource profiles exceeds 10000, then all the values are proportionally scaled down. Each value applies only when the corresponding cache is enabled in the resource profile (for example, enableFlashCache=true).
  - flashCacheMax, xrmemCacheMax: Specifies the maximum fraction of flash cache or XRMEM cache space available to each client associated with the resource profile. The range of valid values is 0-10000, and the default value is 10000. Each value applies only when the corresponding cache is enabled in the resource profile (for example, enableFlashCache=true).

Usage Notes

In addition to regular user-defined resource profiles, you can also create a system-reserved resource profile named $UNASSIGNED. All Exascale clients not explicitly associated with a resource profile are automatically governed by the $UNASSIGNED profile. The $UNASSIGNED resource profile contains only two modifiable attributes:

- flashCacheMax: Specifies the maximum fraction of flash cache shared by clients associated with the $UNASSIGNED profile. The range of valid values is 0-10000. The default value is 10000.
- xrmemCacheMax: Specifies the maximum fraction of XRMEM cache shared by clients associated with the $UNASSIGNED profile. The range of valid values is 0-10000. The default value is 10000.

All Exascale clients governed by the $UNASSIGNED profile share the corresponding cache resources. The behavior differs from regular resource profiles, where each application of the resource profile defines the resource allocation for one associated client.

If the $UNASSIGNED resource profile does not exist, all unassigned Exascale clients share any unassigned flash cache space and XRMEM cache space. If there is no unassigned space to share, the system automatically reserves 5% of the cache space for unassigned Exascale clients.

Examples

Example 6-140 Create a Resource Profile with Default Attribute Values

This example shows two ways to create a resource profile with default attribute values, which is named silver and is associated with the vault named vault2.

```
@> mkresourceprofile vault2/silver
```

```
@> mkresourceprofile silver --attributes vault=vault2
```

Example 6-141 Create a Resource Profile with Default Attribute Values

This example shows how to create a resource profile with default attribute values, which is named bronze and is associated with the vault named vault2. In this example, the vault association is not specified in the command but is derived from the current working directory in the ESCLI session.

```
@vault2> mkresourceprofile bronze
```

Example 6-142 Create a Resource Profile with Specific Attribute Values

This example shows how to create a resource profile that contains specific attribute settings.

```
@> mkresourceprofile vault2/gold --attributes iopsShareHC=11,enableFlashLog=false
```

6.2.10.4 rmresourceprofile

Delete a resource profile.

Purpose

The rmresourceprofile command deletes a resource profile that is associated with an Exascale vault.

Syntax

```
rmresourceprofile [vault-name/]resource-profile-name
```

Command Options

The options for the rmresourceprofile command are:

- *resource-profile-name*: Identifies the name for the resource profile to delete.
- *vault-name*: Identifies the vault that the resource profile is associated with.

  If not specified, then the current working directory is used to identify the vault.

Examples

Example 6-143 Delete a Resource Profile

This example shows how to delete a resource profile named silver that is associated with the vault named vault2.

```
@> rmresourceprofile vault2/silver
```

Example 6-144 Delete a Resource Profile

This example shows how to delete a resource profile named bronze that is associated with the vault named vault2. In this example, the vault association is not specified in the command but is derived from the current working directory in the ESCLI session.

```
@vault2> rmresourceprofile bronze
```

6.2.11 Extended Attribute Management

This section contains references for the Exascale command line interface (ESCLI) commands that are associated with extended attribute management:

- chxattr
  Set an extended attribute for a file or vault.
- lsxattr
  List an extended attribute for a file or vault.
- rmxattr
  Delete an extended attribute for a file or vault.

6.2.11.1 chxattr

Set an extended attribute for a file or vault.

Purpose

The chxattr command allows you to set an extended attribute for an Exascale file or vault.

Syntax

```
chxattr { file-name | vault-name } --name attribute-name
      { --value attribute-value | --bin-value-file file-name }
```

Command Options

The options for the chxattr command are:

- { *file-name* | *vault-name* }: Specifies the name of the file or vault that is the subject of the operation.
- --name: Specifies the name of the extended attribute.
- --value: Specifies the value of the extended attribute.
- --bin-value-file: Specifies the regular file from which to read the value of the extended attribute. The value is assumed to be binary.

Examples

Example 6-145 Set an Extended Attribute for a Vault

You can add an extended attribute to a vault, such as @my-data. The name of the attribute is custom-attr, and its value is my-value.

```
@> chxattr @my-data --name custom-attr --value my-value
```

Example 6-146 Set an Extended Attribute for a File

You can add an extended attribute to a file, such as @my-data/file1. The name of the attribute is custom-attr, and its value is my-value.

```
@> chxattr @my-data/file1 --name custom-attr --value myv-alue
```

Example 6-147 Set a Binary Extended Attribute for a File

You can add a binary extended attribute to a file, such as @my-data/file1. The name of the attribute is binary-attr, and its value is the contents of the file at /home/user/bin-val-in.dat.

```
@> chxattr @my-data/file1 --name binary-attr --bin-value-file /home/user/bin-val-in.dat
```

6.2.11.2 lsxattr

List an extended attribute for a file or vault.

Purpose

The lsxattr command displays information about non-standard attributes created for Exascale files and vaults.

Syntax

```
lsxattr { file-name | vault-name } --name attribute-name
    [ --bin-value-file file-name ]
```

Command Options

The options for the lsxattr command are:

- { *file-name* | *vault-name* }: Specifies the name of the file or vault for which you want to list the specified extended attribute value.
- --name: Specifies the name of the extended attribute.
- --bin-value-file: Specifies the file where you want to write the value of the extended attribute.

Usage Notes

If you do not specify the --bin-value-file option, binary values are output in Base64 format.

Examples

Example 6-148 List an Extended Attribute

You can use the following commands to list the custom-attr extended attribute.

```
@> lsxattr @my-data --name custom-attr
```

```
@> lsxattr @my-data/file1 --name custom-attr
```

Example 6-149 List a Binary Extended Attribute

You can use the following commands to list the binary-attr extended attribute and write the binary value to a regular file at /home/user/bin-val-out.dat.

```
@> lsxattr @my-data/file1 --name binary-attr --bin-value-file /home/user/bin-val-out.dat
```

6.2.11.3 rmxattr

Delete an extended attribute for a file or vault.

Purpose

The rmxattr command deletes non-standard attributes created for Exascale files and vaults.

Syntax

rmxattr { *file-name* | *vault-name* } --name *attribute-name*

Command Options

The options for the rmxattr command are:

- { *file-name* | *vault-name* }: Specifies the name of the file or vault for which you want to delete the specified extended attribute value.
- --name: Specifies the name of the extended attribute.

Examples

Example 6-150 Delete an Extended Attribute

You can use the following commands to delete the custom-attr extended attribute.

```
@> rmxattr @my-data --name custom-attr
```

```
@> rmxattr @my-data/file1 --name custom-attr
```

6.2.12 Block Store Management

This section contains references for the Exascale command line interface (ESCLI) commands that are associated with block store management:

- acfsctl
  Control registration for Exascale-managed ACFS file systems.
- chacfsfilesystem
  Change attributes of an Exascale-managed ACFS file system.
- chvolume
  Change attributes of a volume.
- chvolumeattachment
  Change attributes of a volume attachment.
- chvolumebackup
  Change attributes of a volume backup.
- chvolumegroup
  Change attributes of a volume group.
- chvolumesnapshot
  Change attributes of a volume snapshot.
- lsacfsfilesystem
  List Exascale-managed ACFS file systems.
- lsinitiator
  List EDV initiator information.
- lsvolume
  List volumes.

- lsvolumeattachment
  List volume attachments.
- lsvolumebackup
  List volume backups.
- lsvolumegroup
  List volume groups.
- lsvolumesnapshot
  List volume snapshots.
- mkacfsfilesystem
  Create an Exascale-managed ACFS file system.
- mkvolume
  Create a volume or clone a volume snapshot.
- mkvolumeattachment
  Create an attachment for an Exascale volume.
- mkvolumebackup
  Create a backup of a volume snapshot.
- mkvolumegroup
  Create a volume group.
- mkvolumegroupsnapshot
  Create snapshots of all volumes in the specified volume group.
- mkvolumesnapshot
  Create a snapshot of a volume.
- rmacfsfilesystem
  Delete an Exascale-managed ACFS file system.
- rmvolume
  Delete volumes.
- rmvolumeattachment
  Delete a volume attachment.
- rmvolumebackup
  Delete a volume backup.
- rmvolumegroup
  Delete volume groups.
- rmvolumesnapshot
  Delete volume snapshots.

6.2.12.1 acfsctl

Control registration for Exascale-managed ACFS file systems.

Purpose

The `acfsctl` command controls registration for Exascale-managed implementations of Oracle Advanced Cluster File System (ACFS).

Syntax

```
acfsctl register volume-ID mount-path
    [ --attributes attribute=value[,attribute=value]... ]


acfsctl deregister acfs-ID [ --force ]
```

Command Options

The options for the acfsctl register command are:

- *volume-ID*: Identifies the volume that hosts the file system. You can use the lsacfsfilesystem command to find the volume identifier associated with each Exascale-managed ACFS file system.
- *mount-path*: Specifies the file system mount path.
- --attributes: Optionally specifies attributes governing how the file system is mounted. The permitted attributes are:
  - mountLeafOwner, mountLeafGroup, mountLeafMode: Specify the ownership and permission attributes applied to the leaf directory in the file system mount path.
  - mountNoRootSUID: Controls whether non-root users are permitted to run root-owned binaries with the set user ID (setuid) permission.
    - true: Only the root user can run root-owned setuid binaries.
    - false: Non-root users can also run root-owned setuid binaries.

    If not specified, the default value is false.
  - mountReadOnly: Specifies whether the file system is mounted with read-only access.
    - true: Mounts the file system with read-only access.
    - false: Mounts the file system with read-write access.

    If not specified, the default value is false.

  You cannot set any other file system attributes during registration.

  Use the describe acfsfilesystems to see further details about all Exascale-managed ACFS file system attributes. See also Describing Resources and Attributes.

The options for the acfsctl deregister command are:

- *acfs-ID*: Identifies the ACFS file system being deregistered. You can use the lsacfsfilesystem command to find the identifier for each Exascale-managed ACFS file system.
- --force: Optionally ignores any errors and forces deregistration of the file system, even if it is use.

Usage Notes

Note the following information when using the acfsctl register command:

- The specified volume (*volume-ID*) must be associated with an existing cluster-wide EDV attachment and contain an existing ACFS file system, otherwise the command fails. Registration of a volume with a node-specific EDV attachment is not permitted.

- The ACFS file system is mounted on every node in the Oracle Grid Infrastructure (GI) cluster and the ACFS details are registered with the GI cluster. After registration, the file system is automatically mounted (or remounted) by Oracle Clusterware as required.

Note the following when using the acfsctl deregister command:

- The ACFS file system is dismounted and the mount point directory is removed on every node in the Oracle Grid Infrastructure (GI) cluster. The command also removes the ACFS file system registration in the GI cluster.

6.2.12.2 chacfsfilesystem

Change attributes of an Exascale-managed ACFS file system.

Purpose

The chacfsfilesystem command allows you to modify the attributes of an Exascale-managed implementation of Oracle Advanced Cluster File System (ACFS).

Syntax

```
chacfsfilesystem acfs-ID [ --attributes attribute=value[,attribute=value]... ]
```

Command Options

The options for the chacfsfilesystem command are:

- *acfs-ID*: Identifies the ACFS file system being changed. You can use the lsacfsfilesystem command to find the identifier for each ACFS file system.
- --attributes: Optionally specifies attributes to change.

  Use the describe chacfsfilesystem command to see details about the ACFS file system attributes you can change. See also Describing Resources and Attributes.

Examples

Example 6-151 Change the Quarantine State of an ACFS File System

This example shows changing the quarantine state of the specified ACFS file system to false.

```
@> chacfsfilesystem acfs0003_50e52177583f4be4bad68ac20b65001e --attributes quarantined=false
```

6.2.12.3 chvolume

Change attributes of a volume.

Purpose

The chvolume command allows you to modify the attributes of an Exascale volume.

Syntax

```
chvolume volume-ID [ --attributes attribute=value[,attribute=value]... ]
```

Command Options

The options for the chvolume command are:

- *volume-ID*: Identifies the volume being changed. You can use the lsvolume command to find the identifier for each volume.
- --attributes: Optionally specifies attributes to change.

  Use the describe chvolume command to see details about the volume attributes you can change. See also Describing Resources and Attributes.

Usage Notes

- The chvolume command only proceeds if the specified volume is not attached.
- A volume can have up to two owners. To modify volume ownership, you can add or remove one volume owner at a time.

  To add a volume owner, specify owners=+*user-ID* as an attribute setting. To remove a volume owner, use owners=-*user-ID*.

  For example, see Example 6-153.

Examples

Example 6-152 Change the Volume Size

This example shows changing the size of the volume with the volume ID vol0002_50e52177583f4be4bad68ac20b65001e.

```
@> chvolume vol0002_50e52177583f4be4bad68ac20b65001e --attributes size=200m
```

Example 6-153 Change the Volume Owners

The following examples show different ways to change the volume owners.

- The following command adds peter as an owner of the specified volume:

  ```
  @> chvolume vol0002_50e52177583f4be4bad68ac20b65001e --attributes owners=+peter
  ```

- The following command removes dave as an owner of the specified volume:

  ```
  @> chvolume vol0002_50e52177583f4be4bad68ac20b65001e --attributes owners=-dave
  ```

6.2.12.4 chvolumeattachment

Change attributes of a volume attachment.

Purpose

The chvolumeattachment command allows you to modify the attributes of an Exascale volume attachment.

Syntax

```
chvolumeattachment attachment-ID
            [ --attributes attribute=value[,attribute=value]... ]
```

Command Options

The options for the chvolumeattachment command are:

- *attachment-ID*: Identifies the volume attachment being changed. You can use the lsvolumeattachment command to find the identifier for each volume attachment.
- --attributes: Optionally specifies attributes to change.

  Use the describe ch*volumeattachment command to see details about the volume attachment attributes you can change. See also Describing Resources and Attributes.

Examples

Example 6-154 Change the Quarantine State of a Volume Attachment

This example shows changing the quarantine state of the specified volume attachment to false.

```
@> chvolumeattachment att0002_50e52177583f4be4bad68ac20b65001e --attributes quarantined=false
```

6.2.12.5 chvolumebackup

Change attributes of a volume backup.

Purpose

The chvolumebackup command allows you to modify the attributes of an Exascale volume backup.

Syntax

```
chvolumebackup volume-backup-ID
          [ --attributes attribute=value[,attribute=value]... ]
```

Command Options

The options for the chvolumebackup command are:

- *volume-backup-ID*: Identifies the volume backup being changed. You can use the lsvolumebackup command to find the identifier for each volume backup.
- --attributes: Optionally specifies attributes to change.

  Use the describe chvolumebackup command to see details about the volume backup attributes you can change. See also Describing Resources and Attributes.

Examples

Example 6-155 Change the Name of a Volume Backup

This example shows changing the name of the specified volume backup to my-volume-backup.

```
@> chvolumebackup vol0002_bkp0001_50e52177583f4be4bad68ac20b65001e --attributes name=my-volume-backup
```

6.2.12.6 chvolumegroup

Change attributes of a volume group.

Purpose

The chvolumegroup command allows you to modify the attributes of an Exascale volume group.

Syntax

```
chvolumegroup volume-group-ID
      [ --attributes attribute=value[,attribute=value]... ]
```

Command Options

The options for the chvolumegroup command are:

- *volume-group-ID*: Identifies the volume group being changed. You can use the lsvolumegroup command to find the identifier for each volume group.
- --attributes: Optionally specifies attributes to change.

  The following attributes can be changed:

  – name: Specifies the name of the volume group.

  – resourceSharing: Optionally specifies how provisioned resources are shared by volumes in the group. Permitted values are:

    * aggregate: Configures the volume group to share the I/Os per second (IOPS) bandwidth provisioned to each volume in the group. In this case, a volume can exceed its provisioned limit by sharing unused IOPS provisioned to other volumes in the group.

      For example, consider a group with two volumes that are provisioned with 1000 IOPS each. In this example, aggregate resource sharing enables one volume to use 2000 IOPS when the other volume is idle.

    * specified: Configures the volume group to share the I/Os per second (IOPS) bandwidth limit specified by the iopsProvisioned attribute.

    * none: Specifies that the volume group is not a resource-sharing volume group. In this case, each volume is only subject to resource limits imposed at other levels, such as the individual volume level or the Exascale vault level.

  – iopsProvisioned: Optionally specifies the I/O bandwidth limit provisioned to the volume group, expressed in I/Os per second (IOPS).

  – owners: Modifies the volume group owners.

A volume group can have up to two owners. To modify volume group ownership, you can add or remove one owner at a time.

To add an owner, specify owners=+*user-ID* as an attribute setting. To remove an owner, use owners=-*user-ID*.

For example, see Example 6-156.

Usage Notes

- When resource sharing is disabled on a volume group (resourceSharing=none), the I/O bandwidth provisioned to the volume group must be zero (iopsProvisioned=0).
- When a volume group uses aggregate resource sharing (resourceSharing=aggregate), the I/O bandwidth provisioned to the volume group must be zero (iopsProvisioned=0).
- When a volume group employs specified resource sharing (resourceSharing=specified), the iopsProvisioned attribute value must be greater than zero.
- A volume must always have at least one owner in common with each volume group it belongs to. Any modification violating this requirement is not permitted.

Examples

Example 6-156 Change the Volume Group Owners

The following examples show different ways to change the volume group owners.

- The following command adds peter as an owner of the specified volume group:

  ```
  @> chvolumegroup volgrp0001_08a99eb7e8ff41499dee41de75af3e62 --attributes owners=+peter
  ```

- The following command removes dave as an owner of the specified volume group:

  ```
  @> chvolumegroup volgrp0001_08a99eb7e8ff41499dee41de75af3e62 --attributes owners=-dave
  ```

6.2.12.7 chvolumesnapshot

Change attributes of a volume snapshot.

Purpose

The chvolumesnapshot command allows you to modify the attributes of an Exascale volume snapshot.

Syntax

```
chvolumesnapshot volume-snapshot-ID
        [ --attributes attribute=value[,attribute=value]... ]
```

Command Options

The options for the chvolumesnapshot command are:

- *volume-snapshot-ID*: Identifies the volume snapshot being changed. You can use the lsvolumesnapshot command to find the identifier for each volume snapshot.
- --attributes: Optionally specifies attributes to change.

Use the describe chvolumesnapshot command to see details about the volume snapshot attributes you can change. See also Describing Resources and Attributes.

Examples

Example 6-157 Change the Name of a Volume Snapshot

This example shows changing the name of the specified volume snapshot to vol-snap1.

```
@> chvolumesnapshot vol0001_snap0001_50e52177583f4be4bad68ac20b65001e --attributes name=vol-snap1
```

6.2.12.8 lsacfsfilesystem

List Exascale-managed ACFS file systems.

Purpose

The lsacfsfilesystem command displays information about Exascale-managed implementations of Oracle Advanced Cluster File System (ACFS).

Syntax

```
lsacfsfilesystem [ acfs-ID [ acfs-ID ]... ] [ -l ] [ --detail ]
            [ --attributes attribute[,attribute]... ]
            [ --filter filter[,filter]... ]
            [ --sort [-]attribute[,[-]attribute]... ]
            [ --count value ]
```

Command Options

The options for the lsacfsfilesystem command are:

- *acfs-ID*: Identifies the ACFS file system that you want to list information about. If not specified, the command displays information about all file systems.
- -l: Displays output in a long, tabular format.
- --detail: Displays detailed output with additional attributes.
- --attributes: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the describe acfsfilesystems command. See also Describing Resources and Attributes.
- --filter: Specifies conditions for including items in the output.
- --sort: Sorts the output using the specified attributes.
- --count: Specifies the maximum number of items to display in the output.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

- yyyy-MM-dd"T"HH:mm:ss
- yyyy-MM-dd (Time is assumed to be 00:00 AM)
- HH:mm:ss (Date is assumed to be today)

A date can also be followed by a timezone specification.

Sizes can be specified using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.

For example, the following filter only includes objects created after 9 AM today:

```
--filter createTime>9:00:00
```

- Sorting attributes are specified as: [-]*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with -.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

```
--sort -name,createTime
```

Examples

Example 6-158 List an Exascale-Managed ACFS File System

The following example shows how to list detailed information for an Exascale-managed ACFS file system by using an ACFS ID. In the example, the ACFS ID is acfs0001_2fb06f13cddd4a8d8d636d1f794046cb.

```
@> lsacfsfilesystem acfs0001_2fb06f13cddd4a8d8d636d1f794046cb --detail
```

Example 6-159 List Specific Exascale-Managed ACFS File System Attributes

The following example shows how to list specific attributes of all Exascale-managed ACFS file systems. In the example, the attributes being listed are name and mountPath.

```
@> lsacfsfilesystem --attributes name,mountPath
```

6.2.12.9 lsinitiator

List EDV initiator information.

Purpose

The lsinitiator command displays information about Exascale Direct Volume (EDV) initiators.

Syntax

```
lsinitiator [ edv-initiator-ID ]
      [ --attributes attribute[,attribute]... ]
      [ --detail ] [ -l ]
```

Command Options

The options for the lsinitiator command are:

- *edv-initiator-ID*: Identifies the EDV initiator that you want to list information about. If not specified, the command displays information about all EDV initiators.
- --attributes: Identifies specific attributes to display in the output.

  You can use this option to focus on a subset of attributes as an alternative to the --detail option.
- --detail: Displays detailed output with additional attributes.
- -l: Displays output in a long, tabular format.

Examples

Example 6-160 List EDV Initiator Information

The following example shows how to list all EDV initiators in the Exascale cluster.

```
@> lsinitiator
```

Example 6-161 List Information for a Specific EDV Initiator

The following example shows how to list detailed information about the EDV initiator with the ID: 3d8c1b1c-2fa8-fc6e-3d8c-1b1c2fa8fc6e.

```
@> lsinitiator 3d8c1b1c-2fa8-fc6e-3d8c-1b1c2fa8fc6e
```

6.2.12.10 lsvolume

List volumes.

Purpose

The lsvolume command displays information about Exascale volumes.

Syntax

```
lsvolume [ volume-ID [ volume-ID ]... ] [ -l ] [ --detail ]
    [ --attributes attribute[,attribute]... ]
    [ --filter filter[,filter]... ]
    [ --sort [-]attribute[,[-]attribute]... ]
    [ --count value ]
```

Command Options

The options for the lsvolume command are:

- *volume-ID*: Identifies an Exascale volume that you want to list information about. If not specified, the command displays information about all volumes.
- -l: Displays output in a long, tabular format.
- --detail: Displays detailed output with additional attributes.

- --attributes: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the describe volumes command. See also Describing Resources and Attributes.
- --filter: Specifies conditions for including items in the output.
- --sort: Sorts the output using the specified attributes.
- --count: Specifies the maximum number of items to display in the output.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  – yyyy-MM-dd"T"HH:mm:ss

  – yyyy-MM-dd (Time is assumed to be 00:00 AM)

  – HH:mm:ss (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  Sizes can be specified using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.

  For example, the following filter only includes objects created after 9 AM today:

  ```
  --filter createTime>9:00:00
  ```

- Sorting attributes are specified as: [-]*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with -.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

Examples

Example 6-162 List Information for a Specific Volume

The following example shows how to list detailed information for volume ID: vol0002_50e52177583f4be4bad68ac20b65001e.

```
@> lsvolume vol0002_50e52177583f4be4bad68ac20b65001e --detail
```

Example 6-163 List Specific Volume Attributes

The following example shows how to list specific volume attributes for volume ID: vol0002_50e52177583f4be4bad68ac20b65001e.

```
@> lsvolume vol0002_50e52177583f4be4bad68ac20b65001e --attributes id,size,iopsProvisioned
```

Example 6-164 List Volumes Matching a Filter

The following example shows how to list detailed volume information for volumes that are sized greater than 100 MB. The output is also sorted by volume size.

```
@> lsvolume --detail --filter size>100M --sort size
```

6.2.12.11 lsvolumeattachment

List volume attachments.

Purpose

The lsvolumeattachment command displays information about Exascale volume attachments.

Syntax

```
lsvolumeattachment [ attachment-ID [ attachment-ID ]... ] [ -l ] [ --detail ]
          [ --attributes attribute[,attribute]... ]
          [ --filter filter[,filter]... ]
          [ --sort [-]attribute[,[-]attribute]... ]
          [ --count value ]
```

Command Options

The options for the lsvolumeattachment command are:

- *attachment-ID*: Identifies the volume attachment that you want to list information about. If not specified, the command displays information about all volume attachments.
- -l: Displays output in a long, tabular format.
- --detail: Displays detailed output with additional attributes.
- --attributes: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the describe volumeattachments command. See also Describing Resources and Attributes.
- --filter: Specifies conditions for including items in the output.
- --sort: Sorts the output using the specified attributes.
- --count: Specifies the maximum number of items to display in the output.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: <*attribute*><*operator*><*value*>.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  - yyyy-MM-dd"T"HH:mm:ss
  - yyyy-MM-dd (Time is assumed to be 00:00 AM)
  - HH:mm:ss (Date is assumed to be today)

A date can also be followed by a timezone specification.

Sizes can be specified using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.

For example, the following filter only includes objects created after 9 AM today:

```
--filter createTime>9:00:00
```

- Sorting attributes are specified as: [-]*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with -.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

Examples

Example 6-165 List an EDV Attachment Using an Attachment ID

The following example shows how to list attachment information for an Exascale Direct Volume by using an attachment ID. In the example, the attachment ID is att0003_50e52177583f4be4bad68ac20b65001e.

```
@> lsvolumeattachment att0003_50e52177583f4be4bad68ac20b65001e
```

Example 6-166 List Specific EDV Attachment Attributes

The following example shows how to list specific attributes of all EDV attachments. In the example, the attributes being listed are id and devicePath.

```
@> lsvolumeattachment --attributes id,devicePath
```

6.2.12.12 lsvolumebackup

List volume backups.

Purpose

The lsvolumebackup command displays information about Exascale volume backups.

Syntax

```
lsvolumebackup [ volume-backup-ID [ volume-backup-ID ]... ]
           [ -l ] [ --detail ]
           [ --attributes attribute[,attribute]... ]
           [ --filter filter[,filter]... ]
           [ --sort [-]attribute[,[-]attribute]... ]
           [ --count value ]
```

Command Options

The options for the lsvolumebackup command are:

- *volume-backup-ID*: Identifies an Exascale volume backup that you want to list information about. If not specified, the command displays information about all volume backups.
- `-l`: Displays output in a long, tabular format.
- `--detail`: Displays detailed output with additional attributes.
- `--attributes`: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the `describe volumebackups` command. See also Describing Resources and Attributes.
- `--filter`: Specifies conditions for including items in the output.
- `--sort`: Sorts the output using the specified attributes.
- `--count`: Specifies the maximum number of items to display in the output.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  - `yyyy-MM-dd"T"HH:mm:ss`
  - `yyyy-MM-dd` (Time is assumed to be 00:00 AM)
  - `HH:mm:ss` (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  Sizes can be specified using suffixes `K`, `KB`, `M`, `MB`, `G`, `GB`, `T`, `TB`. The suffix is not case-sensitive.

  For example, the following filter only includes objects created after 9 AM today:

  ```
  --filter createTime>9:00:00
  ```

- Sorting attributes are specified as: `[-]`*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with `-`.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

Examples

Example 6-167 List Information for Volume Backups

The following example shows how to list detailed information about all volume backups.

```
@> lsvolumebackup --detail
```

Example 6-168 List Information for Specific Volume Backups

The following example shows how to list detailed information about two volume backups with specific IDs.

```
@> lsvolumebackup vol0002_bkp0001_fedd311081ee490481b7e19bdb691999
vol0002_bkp0002_fedd311081ee490481b8f2acab692334 --detail
```

Example 6-169 List Information for Specific Volume Backups

The following example shows how to list information about volume backups with names starting with `my-volume-backup`.

```
@> lsvolumebackup --filter name=my-volume-backup*
```

6.2.12.13 lsvolumegroup

List volume groups.

Purpose

The `lsvolumegroup` command displays information about Exascale volume groups.

Syntax

```
lsvolumegroup [ volume-group-ID [ volume-group-ID ]... ] [ -l ] [ --detail ]
        [ --attributes attribute[,attribute]... ]
        [ --filter filter[,filter]... ]
        [ --sort [-]attribute[,[-]attribute]... ]
        [ --count value ]
```

Command Options

The options for the `lsvolumegroup` command are:

- *volume-group-ID*: Identifies an Exascale volume group that you want to list information about. If not specified, the command displays information about all volume groups.
- `-l`: Displays output in a long, tabular format.
- `--detail`: Displays detailed output with additional attributes.
- `--attributes`: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the `describe volumegroups` command. See also Describing Resources and Attributes.
- `--filter`: Specifies conditions for including items in the output.
- `--sort`: Sorts the output using the specified attributes.
- `--count`: Specifies the maximum number of items to display in the output.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

The allowed operators are =, !=, >=, <=, >, and <.

Multiple comma-separated filter conditions are combined using AND logic.

Dates can be specified using the following formats:

- yyyy-MM-dd"T"HH:mm:ss
- yyyy-MM-dd (Time is assumed to be 00:00 AM)
- HH:mm:ss (Date is assumed to be today)

A date can also be followed by a timezone specification.

For example, the following filter only includes objects created after 9 AM today:

```
--filter createTime>9:00:00
```

- Sorting attributes are specified as: [-]*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with -.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

```
--sort -name,createTime
```

Examples

Example 6-170 List Information for Volume Groups

The following example shows how to list detailed information about all volume groups.

```
@> lsvolumegroup --detail
```

Example 6-171 List Information for Specific Volume Groups

The following example shows how to list detailed information about two volume groups with specific IDs.

```
@> lsvolumegroup volgrp0001_08a99eb7e8ff41499dee41de75af3e62
volgrp0002_67e399099f3845eeb2310485a13d1aee --detail
```

Example 6-172 List Information for Specific Volume Groups

The following examples shows how to list information about volume groups matching the specified filter conditions.

```
@> lsvolumegroup --filter name=my-group*
```

```
@> lsvolumegroup --filter iopsProvisioned>1000
```

6.2.12.14 lsvolumesnapshot

List volume snapshots.

Purpose

The `lsvolumesnapshot` command displays information about Exascale volume snapshots.

Syntax

```
lsvolumesnapshot [ volume-snap-ID [ volume-snap-ID ]... ] [ -l ] [ --detail ]
          [ --attributes attribute[,attribute]... ]
          [ --filter filter[,filter]... ]
          [ --sort [-]attribute[,[-]attribute]... ]
          [ --count value ]
```

Command Options

The options for the `lsvolumesnapshot` command are:

- *volume-snap-ID*: Identifies an Exascale volume snapshot that you want to list information about. If not specified, the command displays information about all volume snapshots.
- `-l`: Displays output in a long, tabular format.
- `--detail`: Displays detailed output with additional attributes.
- `--attributes`: Identifies specific attributes to display in the output.

  To see a list of all available attributes and their descriptions, use the `describe volumesnapshots` command. See also Describing Resources and Attributes.
- `--filter`: Specifies conditions for including items in the output.
- `--sort`: Sorts the output using the specified attributes.
- `--count`: Specifies the maximum number of items to display in the output.

Usage Notes

Note the following information when using this command:

- Filter conditions are specified as: *<attribute><operator><value>*.

  The allowed operators are =, !=, >=, <=, >, and <.

  Multiple comma-separated filter conditions are combined using AND logic.

  Dates can be specified using the following formats:

  - `yyyy-MM-dd"T"HH:mm:ss`
  - `yyyy-MM-dd` (Time is assumed to be 00:00 AM)
  - `HH:mm:ss` (Date is assumed to be today)

  A date can also be followed by a timezone specification.

  Sizes can be specified using suffixes `K`, `KB`, `M`, `MB`, `G`, `GB`, `T`, `TB`. The suffix is not case-sensitive.

For example, the following filter only includes objects created after 9 AM today:

```
--filter createTime>9:00:00
```

- Sorting attributes are specified as: [-]*attribute*. Multiple sort attributes are comma-delimited. The default sort order is ascending. For descending sort order, prefix the attribute name with -.

  For example, use the following to primarily sort by name in descending order, and use creation time in ascending order to further sort entries with the same name:

  ```
  --sort -name,createTime
  ```

Examples

Example 6-173 List Information for Volume Snapshots

The following example shows how to list detailed information about all of the volume snapshots.

```
@> lsvolumesnapshot --detail
```

Example 6-174 List Information for a Specific Volume Snapshot

The following example shows how to list detailed information for volume snapshot ID: vol0002_snap0001_50e52177583f4be4bad68ac20b65001e.

```
@> lsvolumesnapshot vol0002_snap0001_50e52177583f4be4bad68ac20b65001e
```

Example 6-175 List Specific Attributes for Volume Snapshots

The following example shows how to list specific attributes of all the volume snapshots.

```
@> lsvolumesnapshot --attributes id,volume,size
```

6.2.12.15 mkacfsfilesystem

Create an Exascale-managed ACFS file system.

Purpose

The mkacfsfilesystem command creates an Exascale-managed implementation of Oracle Advanced Cluster File System (ACFS) on an Exascale volume using Exascale Direct Volumes (EDV).

Syntax

```
mkacfsfilesystem volume-ID mount-path [ --force ]
           [ --attributes attribute=value[,attribute=value]... ]
```

Command Options

The options for the mkacfsfilesystem command are:

- *volume-ID*: Identifies the volume that hosts the file system. You can use the lsvolume command to find the identifier for each volume.
- *mount-path*: Specifies the file system mount path.
- `--force`: Optionally instructs the command to ignore and overwrite any existing file system on the volume.
- `--attributes`: Optionally specifies attributes for the file system.

  Use the `describe mkacfsfilesystem` command to see details about the ACFS file system attributes you can set. See also Describing Resources and Attributes.

Usage Notes

Note the following information when using this command:

- The specified volume (*volume-ID*) must be associated with a cluster-wide EDV attachment, otherwise the command fails. Using a volume with a node-specific EDV attachment is not permitted.
- The ACFS file system is mounted on every node in the Oracle Grid Infrastructure (GI) cluster and the ACFS details are registered with the GI cluster. After registration, the file system is automatically mounted (or remounted) by Oracle Clusterware as required.
- Before using the `mkacfsfilesystem` command, ensure that ACFS is configured appropriately on the target system.

  For instance, you can create a file system using ACFS encryption by specifying the `encryptionEnabled`, `encryptionAlgorithm`, and `encryptionKeyLength` attributes in the `mkacfsfilesystem` command. For example:

  ```
  @> mkacfsfilesystem vol0001_bbd6fb4c75e2411b9bf366fe702eabaf /mnt/acfs1
     --attributes encryptionEnabled=true,encryptionAlgorithm=AES,encryptionKeyLength=192
  ```

  However, the command fails if ACFS encryption is not initialized on the target system. To initialize ACFS encryption, the system administrator must run the following command before the file system is created:

  ```
  # acfsutil encr init
  ```

6.2.12.16 mkvolume

Create a volume or clone a volume snapshot.

Purpose

The `mkvolume` command allows you to create a new Exascale volume, clone a volume snapshot, or directly clone an existing volume.

Syntax

To create a new volume:

```
mkvolume size --vault vault
      [ --attributes attribute=value[,attribute=value]... ]
```

To clone a volume snapshot:

```
mkvolume --attributes volumeSnapshot=parent-snapshot-ID[,name=clone-name][,iopsProvisioned=integer-value]
[,iopsInherited={true|false}]
```

To directly clone an existing volume:

```
mkvolume --attributes volumeSource=parent-volume-ID[,name=clone-name][,iopsProvisioned=integer-value]
[,iopsInherited={true|false}]
```

> **Note**
>
> To directly clone an existing volume, you must use Oracle Exadata System Software release 25.1.0 or later.

Command Options

The options for the mkvolume command are:

- *size*: Specifies the size of the volume. The *size* can be specified using suffixes K, KB, M, MB, G, GB, T, TB. The suffix is not case-sensitive.

  > **Note**
  >
  > If you intend to use the volume to support Oracle ACFS, note that ACFS requires a minimum volume size of 512 MB.

- --vault: Specifies the vault that the volume is created in.
- --attributes: Optionally specifies attributes for the volume.

  Use the describe mkvolume command to see details about the volume attributes you can set. See also Describing Resources and Attributes.

Usage Notes

The following usage notes apply only when the mkvolume command is used to clone a volume.

- When cloning a volume snapshot, volumeSnapshot=*parent-snapshot-ID* identifies the volume snapshot that you want to clone.
- When directly cloning an existing volume, volumeSource=*parent-volume-ID* identifies the volume that you want to clone.
- When cloning a volume, either directly or using a volume snapshot, you can optionally specify the following attribute settings. No additional command options or attribute settings are required or permitted.
  - name=*clone-name*: Optionally specifies the name of the volume clone, which makes it easier for you to identify later. If not specified, a system-generated name is assigned.
  - iopsProvisioned=*integer-value*: Optionally specifies the I/O bandwidth provisioned for the volume clone. The I/O bandwidth is expressed as the number of I/Os per second (IOPS).

  - iopsInherited={true|false}: Optionally specifies whether the volume clone inherits I/O bandwidth from the nearest ancestor in the volume hierarchy with provisioned (not inherited) I/O bandwidth.

- When cloning a volume, either directly or using a volume snapshot, the following describes the relationship between the iopsProvisioned and iopsInherited attribute settings:
  - If iopsProvisioned is unspecified and iopsInherited is not set to false (iopsInherited is unspecified or iopsInherited=true), then:
    - The volume clone inherits I/O bandwidth from its nearest ancestor.
    - The volume clone has the setting: iopsInherited=true.
  - If iopsProvisioned is unspecified and iopsInherited=false, then:
    - The volume clone is provisioned with unlimited iopsProvisioned.
    - The volume clone has the setting: iopsInherited=false.
  - If you specify a value for iopsProvisioned and iopsInherited is not set to true (iopsInherited is unspecified or iopsInherited=false), then:
    - The volume clone is governed by the specified iopsProvisioned value.
    - The volume clone has the setting: iopsInherited=false.
  - An error occurs if you specify a value for iopsProvisioned and iopsInherited=true. This is an invalid combination.

Examples

Example 6-176 Create a Volume

This example shows creating a volume that is 100 MB in size and located in the vault named my-volumes.

```
@> mkvolume 100m --vault my-volumes
```

Example 6-177 Create a Volume with Attribute Settings

This example shows creating a volume that is 200 MB in size and located in the vault named my-volumes. The example also includes the following specific attribute settings:

- The volume name is set to vol2.
- The volume is provisioned with 1000 IOPS (IOs per second).

```
@> mkvolume 200m --vault my-volumes --attributes name=vol2,iopsProvisioned=1000
```

6.2.12.17 mkvolumeattachment

Create an attachment for an Exascale volume.

Purpose

The mkvolumeattachment command allows you to create an attachment for an Exascale volume.

Syntax

```
mkvolumeattachment volume-ID device-name
            --attributes { giClusterId=cluster-ID |
                      initiator=edv-initiator-ID }
```

Command Options

The options for the mkvolumeattachment command are:

- *volume-ID*: Specifies the identifier for the Exascale volume for which you want to create an attachment. You can use the lsvolume command to find the identifier for each volume.
- *device-name*: For an Exascale Direct Volume (EDV) attachment, specifies the device name to use in conjunction with the attachment. This is a user-supplied name, which is applied to the device file that is associated with the attachment. After attachment, the corresponding device file is located under /dev/exc/.
- --attributes: Specifies different attributes depending on the attachment type.
  - For a cluster-wide EDV attachment only, the giClusterId attribute specifies the Oracle Grid Infrastructure (GI) cluster ID associated with the volume attachment. When you specify this attribute, the corresponding device file is created on all nodes in the Oracle GI cluster.
  - For a node-specific EDV attachment only, the initiator attribute specifies the EDV initiator ID associated with the volume attachment. When you specify this attribute, the corresponding device file is only created on the cluster node associated with the specified EDV initiator ID.

Usage Notes

The following notes apply to EDV attachments:

- During initial system deployment with Oracle Exadata Deployment Assistant (OEDA), the Exascale Direct Volume (EDV) service is configured on each Exadata compute node (bare-metal or VM) and runs with the permissions of the Exascale user that manages the Oracle Grid Infrastructure (GI) cluster. To create an EDV attachment, you must use the Exascale user linked with the EDV service.

  If the GI cluster uses a non-role-separated user configuration with one Oracle OS user account, then the associated Exascale user is related to the EDV service. If the GI cluster uses a role-separated configuration with a Grid OS user account and an Oracle OS user account, then the EDV service is linked to the Exascale user associated with the Grid OS account.

  To find the Exascale user linked with the EDV service, use the ESCLI lsinitiator command with the --detail option and examine the user attribute.
- Each EDV attachment also has a kernel device file at /dev/exc-dev*N*, where *N* is the minor number of the device. The kernel device name is contained as an attribute of the EDV attachment and is visible using the ESCLI lsvolumeattachment command.

  Note that most Linux tools, such as iostat, display the kernel device file at /dev/exc-dev*N*, while Exascale commands use the user-named device file (under /dev/exc/).

The relationship between the kernel device file and the user-named device file is also recorded in the udev database and is visible using the following Linux command:

```
# udevadm info device-file
```

In the udevadm command, for the *device-file* value, you can specify either the kernel device file (/dev/exc-dev*N*) or the user-named device file (under /dev/exc/).

- By default, read and write access to EDV device files is only available to the root operating system user and members of the disk group. Depending on your use case, you may need to modify the permissions on the EDV device files before using them.

  For example, to make the EDV device file at /dev/exc/my-vol readable and writable by the oracle user and dba group, you could configure it using a udev rule similar to the following:

  ```
  # cat /etc/udev/rules.d/57-edv-user.rules
  KERNEL=="exc-*", ENV{EXC_ALIAS}=="my-vol", OWNER="oracle", GROUP="dba", MODE="0660"
  ```

- To facilitate the management of udev rules related to EDV devices, each EDV client node is configured with a template udev rules file at /etc/udev/rules.d/57-edv-user.rules, which you can modify to fulfill your requirements. To maintain existing udev rules, /etc/udev/rules.d/57-edv-user.rules is preserved whenever the EDV client software is updated.
- Each EDV client node can support a maximum of 3000 attachments at the same time. This limit includes the total of all cluster attachments involving the server, as well as local attachments specific to the server.

Examples

Example 6-178 Create a Cluster-Wide EDV Volume Attachment

You can create a cluster-wide EDV volume attachment by specifying a *volume-ID*, *device-name*, and *cluster-ID*.

In the following example, the *volume-ID* is vol0007_50e52177583f4be4bad68ac20b65001e, the *device-name* is my-vol, and the *cluster-ID* is 72071863FA3E7FCAFF9F42A96957F4C5.

```
@> mkvolumeattachment vol0007_50e52177583f4be4bad68ac20b65001e my-vol --attributes
giClusterId=72071863-fa3e-7fca-ff9f-42a96957f4c5
```

Example 6-179 Create a Node-Specific EDV Volume Attachment

You can create a node-specific EDV volume attachment by specifying a *volume-ID*, *device-name*, and *edv-initiator-ID*.

In the following example, the *volume-ID* is vol0007_50e52177583f4be4bad68ac20b65001e, the *device-name* is my-vol, and the *edv-initiator-ID* is b0b057aa-1f2c-0f48-b0b0-57aa1f2c0f48.

```
@> mkvolumeattachment vol0007_50e52177583f4be4bad68ac20b65001e my-vol --attributes
initiator=b0b057aa-1f2c-0f48-b0b0-57aa1f2c0f48
```

6.2.12.18 mkvolumebackup

Create a backup of a volume snapshot.

Purpose

The mkvolumebackup command creates a backup of an Exascale volume snapshot.

Syntax

```
mkvolumebackup --attributes volumeSnapshot=volume-snapshot-ID,destination=backup-
destination[,attribute=value]...
```

Command Options

The command options for the mkvolumebackup command are:

- --attributes: Specifies attributes settings for the volume backup:
  - volumeSnapshot: Identifies the volume snapshot to back up. This attribute must be specified. You can use the lsvolumesnapshot command to find the identifier for each volume snapshot.
  - destination: Specifies the backup destination as a location in an Exascale vault.

    If the destination path ends with a forward slash (/), a file name is automatically created using the following pattern: vol_<*volume-ID*>_bkp_<*backup-ID*>.
  - You can also specify optional attributes, including a user-defined name that you can use to easily identify the backup and Exascale file storage attributes that you want to associate with the backup file.

    Use the describe mkvolumebackup command to view details about all the volume attributes you can set with mkvolumebackup.

Examples

Example 6-180 Create a Volume Snapshot Backup

This example shows creating a backup for volume snapshot ID vol0002_snap0001_50e52177583f4be4bad68ac20b65001e with the backup being stored at @my-vault1/bkup/.

```
@> mkvolumebackup --attributes
volumeSnapshot=vol0002_snap0001_3c0e9866d66345afbb316472f0bce825,destination=@my-vault1/bkup/
```

Example 6-181 Create a Volume Snapshot Backup with Optional Attributes

This example shows creating another backup for volume snapshot ID vol0002_snap0001_50e52177583f4be4bad68ac20b65001e. In this case, the backup destination is @my-vault1/bkup2/. Additionally, the backup file content type is explicitly set to RECO and the backup name is set to my-backup2.

```
@> mkvolumebackup --attributes
volumeSnapshot=vol0002_snap0001_3c0e9866d66345afbb316472f0bce825,destination=@my-vault1/
bkup2/,contentType=RECO,name=my-backup2
```

6.2.12.19 mkvolumegroup

Create a volume group.

Purpose

The mkvolumegroup command creates a volume group.

Syntax

```
mkvolumegroup [ volume-group-name ]
          [ --attributes attribute=value[,attribute=value]... ]
```

Command Options

The command options for the mkvolumegroup command are:

- *volume-group-name*: Specifies the name of the volume group.

  If the name is not specified here, it must be specified as an attribute setting.

- --attributes: Specifies attributes settings for the volume group:
  - name: Specifies the name of the volume group.

    If the name is not specified as an attribute setting, it must be specified as the first argument of the mkvolumegroup command.

  - resourceSharing: Optionally specifies how provisioned resources are shared by volumes in the group. The default value is none. Permitted values are:
    * aggregate: Configures the volume group to share the I/Os per second (IOPS) bandwidth provisioned to each volume in the group. In this case, a volume can exceed its provisioned limit by sharing unused IOPS provisioned to other volumes in the group.

      For example, consider a group with two volumes that are provisioned with 1000 IOPS each. In this example, aggregate resource sharing enables one volume to use 2000 IOPS when the other volume is idle.

    * specified: Configures the volume group to share the I/Os per second (IOPS) bandwidth limit specified by the iopsProvisioned attribute.
    * none: Specifies that the volume group is not a resource-sharing volume group. In this case, each volume is only subject to resource limits imposed at other levels, such as the individual volume level or the Exascale vault level.
  - iopsProvisioned: Optionally specifies the I/O bandwidth limit provisioned to the volume group, expressed in I/Os per second (IOPS). The default value is 0.

Usage Notes

- When resource sharing is disabled on a volume group (resourceSharing=none), the I/O bandwidth provisioned to the volume group must be zero (iopsProvisioned=0).
- When a volume group uses aggregate resource sharing (resourceSharing=aggregate), the I/O bandwidth provisioned to the volume group must be zero (iopsProvisioned=0).
- When a volume group employs specified resource sharing (resourceSharing=specified), the iopsProvisioned attribute value must be greater than zero.

Examples

Example 6-182 Create a Volume Group

These examples show how to create volume groups with the specified names and other attribute settings.

```
@> mkvolumegroup vol-group1 --attributes resourceShareing=specified,iopsProvisioned=1000
```

```
@> mkvolumegroup --attributes name=vol-group2,resourceShare=aggregate
```

6.2.12.20 mkvolumegroupsnapshot

Create snapshots of all volumes in the specified volume group.

Purpose

The mkvolumegroupsnapshot creates a set of volume snapshots with point-in-time consistency for all volumes in the specified volume group.

Syntax

```
mkvolumegroupsnapshot volume-group-ID
```

Command Options

The options for the mkvolumegroupsnapshot command are:

- *volume-group-ID*: Identifies the volume group that you want to act on. You can use the lsvolumegroup command to find the identifier for each volume group.

Usage Notes

The volume snapshot operation fails if the specified volume supports Exascale Direct Volume (EDV) attachments and Oracle Advanced Cluster File System (ACFS) is currently mounted. To create a volume snapshot in this situation, you must either:

- Unmount the affected file system on all servers before you run the command to create the volume snapshot. Then, mount the file system again afterward.
- Alternatively, you can use the Oracle ACFS command-line utility (acfsutil) to synchronize the file system and create the volume snapshot. On an EDV client node, run the acfsutil volsnap create command as the root user and specify the ACFS mountpoint or EDV device.

  For example:

  ```
  # acfsutil volsnap create /mnt/my-acfs
  ```

Examples

Example 6-183 Snapshot Volumes in a Volume Group

This example shows creating a snapshot of volumes in the volume group with the ID volgrp0001_8b599d1725244432a1f47407b4e9493e.

```
@> mkvolumegroupsnapshot volgrp0001_8b599d1725244432a1f47407b4e9493e
```

6.2.12.21 mkvolumesnapshot

Create a snapshot of a volume.

Purpose

The mkvolumesnapshot creates a snapshot of the specified Exascale volume.

Syntax

```
mkvolumesnapshot volume-ID
        [ --attributes attribute=value[,attribute=value]... ]
```

Command Options

The options for the mkvolumesnapshot command are:

- *volume-ID*: Identifies the volume that you want to snapshot. You can use the lsvolume command to find the identifier for each volume.
- --attributes: Optionally specifies attributes for the volume snapshot.

  Use the describe mkvolumesnapshot command to see details about the volume snapshot attributes you can set. See also Describing Resources and Attributes.

Usage Notes

The volume snapshot operation fails if the specified volume supports Exascale Direct Volume (EDV) attachments and Oracle Advanced Cluster File System (ACFS) is currently mounted. To create a volume snapshot in this situation, you must either:

- Unmount the affected file system on all servers before you run the command to create the volume snapshot. Then, mount the file system again afterward.
- Alternatively, you can use the Oracle ACFS command-line utility (acfsutil) to synchronize the file system and create the volume snapshot. On an EDV client node, run the acfsutil volsnap create command as the root user and specify the ACFS mountpoint or EDV device.

  For example:

  ```
  # acfsutil volsnap create /mnt/my-acfs
  ```

Examples

Example 6-184 Create a Volume Snapshot

This example shows creating a snapshot of volume vol0001_50e52177583f4be4bad68ac20b65001e.

```
@> mkvolumesnapshot vol0001_50e52177583f4be4bad68ac20b65001e
```

Example 6-185 Create a Volume Snapshot with Attribute Settings

This example shows creating a snapshot of volume vol0001_50e52177583f4be4bad68ac20b65001e with specific attribute settings.

```
@> mkvolumesnapshot vol0001_50e52177583f4be4bad68ac20b65001e --attributes name=snap1
```

6.2.12.22 rmacfsfilesystem

Delete an Exascale-managed ACFS file system.

Purpose

The rmacfsfilesystem command deletes an Exascale-managed implementation of Oracle Advanced Cluster File System (ACFS).

Syntax

```
rmacfsfilesystem acfs-ID
```

Command Options

The options for the rmacfsfilesystem command are:

- *acfs-ID*: Identifies the ACFS file system being deleted. You can use the lsacfsfilesystem command to find the identifier for each Exascale-managed ACFS file system.

Usage Notes

Note the following information when using this command:

- The ACFS file system is dismounted and the mount point directory is removed on every node in the Oracle Grid Infrastructure (GI) cluster. The command also removes the ACFS file system registration in the GI cluster. Furthermore, the file system is removed from the underlying volume, effectively deleting the files in the file system. The command does not alter the underlying volume or volume attachment in any other way.

6.2.12.23 rmvolume

Delete volumes.

Purpose

The rmvolume command deletes Exascale volumes.

Syntax

```
rmvolume volume-ID [ volume-ID ]... [ --keep-data ]
```

Command Options

The options for the `rmvolume` command are:

- *volume-ID*: Identifies the volume being deleted. You can use the lsvolume command to find the identifier for each volume.
- `--keep-data`: Optionally instructs the command to delete the metadata associated with the specified volume(s) while preserving the underlying volume data file(s) for potential future use.

Usage Notes

Note the following information when using this command:

- You must detach the volume before it can be deleted.

Examples

Example 6-186 Delete a Volume

This example shows deleting the volume with the volume ID `vol0001_50e52177583f4be4bad68ac20b65001e`.

```
@> rmvolume vol0001_50e52177583f4be4bad68ac20b65001e
```

6.2.12.24 rmvolumeattachment

Delete a volume attachment.

Purpose

The `rmvolumeattachment` command allows you to delete an Exascale volume attachment.

Syntax

```
rmvolumeattachment attachment-ID [ --force ]
```

Command Options

The options for the `rmvolumeattachment` command are:

- *attachment-ID*: Identifies the volume attachment that is the subject of the operation. You can use the lsvolumeattachment command to find the identifier for each volume attachment.
- `--force`: Optionally removes a volume attachment even if it is being used.

Examples

Example 6-187 Delete an EDV Attachment

You can delete an Exascale Direct Volume (EDV) attachment by specifying just the *attachment-ID*. In the following example, the *attachment-ID* is att0003_50e52177583f4be4bad68ac20b65001e.

```
@> rmvolumeattachment att0003_50e52177583f4be4bad68ac20b65001e
```

6.2.12.25 rmvolumebackup

Delete a volume backup.

Purpose

The rmvolumebackup command deletes a backup of an Exascale volume.

Syntax

```
rmvolumebackup volume-backup-ID [ --keep-data ]
```

Command Options

The options for the rmvolumebackup command are:

- *volume-backup-ID*: Identifies the volume backup that is being deleted. You can use the lsvolumebackup command to find the identifier for each volume backup.
- --keep-data: Optionally instructs the command to delete the metadata associated with the specified volume backups(s) while preserving the underlying volume data file(s) for potential future use.

Examples

Example 6-188 Delete a Volume Backup

This example shows deleting the volume backup having the ID vol0002_bkp0001_50e52177583f4be4bad68ac20b65001e.

```
@> rmvolumebackup vol0002_bkp0001_50e52177583f4be4bad68ac20b65001e
```

6.2.12.26 rmvolumegroup

Delete volume groups.

Purpose

The rmvolumegroup command deletes the specified Exascale volume groups.

Syntax

```
rmvolumegroup volume-group-ID [ volume-group-ID ]...
```

Command Options

The options for the rmvolumegroup command are:

- *volume-group-ID*: Identifies the volume group being deleted. You can use the lsvolumegroup command to find the identifier for each volume group.

Usage Notes

To delete a volume group, you must first remove all volumes from it. See Adding and Removing Volumes in a Volume Group.

Examples

Example 6-189 Delete a Volume Group

This example shows deleting the volume group having the ID volgrp0001_c2a3cec3a7e7413181fa1730093b3110.

```
@> rmvolumegroup volgrp0001_c2a3cec3a7e7413181fa1730093b3110
```

6.2.12.27 rmvolumesnapshot

Delete volume snapshots.

Purpose

The rmvolumesnapshot command allows you to delete one or more Exascale volume snapshots.

Syntax

```
rmvolumesnapshot volume-snap-ID [ volume-snap-ID ]... [ --keep-data ]
```

Command Options

The options for the rmvolumesnapshot command are:

- *volume-snap-ID*: Identifies the volume snapshot being deleted. You can use the lsvolumesnapshot command to find the identifier for each volume snapshot.
- --keep-data: Optionally instructs the command to delete the metadata associated with the specified volume snapshot(s) while preserving the underlying volume data file(s) for potential future use.

Usage Notes

You must detach the volume snapshot before it can be deleted.

Examples

Example 6-190 Delete a Volume Snapshot

This example shows deleting the volume snapshot having the ID vol0002_snap0001_50e52177583f4be4bad68ac20b65001e.

```
@> rmvolumesnapshot vol0002_snap0001_50e52177583f4be4bad68ac20b65001e
```