Abstract

Document generated on: 2018-11-19 (revision: 7098)
# Table of Contents

Preface .......................................................................................................................... ix
1 What’s New in the Oracle VM Manager Command Line Interface (CLI)? ....................... 1
   1.1 CLI Changes in Release 3.4 ....................................................................................... 1
      1.1.1 CLI Modifications ......................................................................................... 1
      1.1.2 CLI New Features ....................................................................................... 2
      1.1.3 CLI Object Attribute Changes .................................................................... 4
2 Introduction to the Oracle VM Manager Command Line Interface (CLI) ......................... 5
   2.1 Starting and Stopping the CLI ................................................................................ 5
   2.2 Connecting to the CLI .......................................................................................... 6
   2.3 SSH Host Keys ..................................................................................................... 7
   2.4 Key-Based Authentication .................................................................................... 7
   2.5 Using SSH to Run Background Processes ........................................................... 9
   2.6 Using the Oracle VM CLI and Getting Help ........................................................... 9
      2.6.1 Return Status Values .................................................................................... 12
      2.6.2 Masking Sensitive Data ................................................................................ 13
   2.7 Passing in a CLI Command at the Command Line .................................................. 13
   2.8 Sample Scripts ...................................................................................................... 13
   2.9 Integrating the CLI into Your Applications ........................................................... 16
   2.10 Configuring the Oracle VM CLI .......................................................................... 16
      2.10.1 SSH Port ..................................................................................................... 17
      2.10.2 Timeout ....................................................................................................... 17
      2.10.3 Case Sensitivity ........................................................................................... 17
      2.10.4 Public Key Authentication Expiry ............................................................... 17
      2.10.5 Lock Exceptions ........................................................................................ 17
   2.11 CLI Logs ............................................................................................................. 18
I Using the CLI ............................................................................................................. 19
   3 Discovering Oracle VM Servers .............................................................................. 21
      3.1 Discovering an Oracle VM Server ................................................................... 21
   4 Discovering Storage .................................................................................................. 23
      4.1 Discovering a file server ..................................................................................... 23
      4.2 Discovering a storage array .............................................................................. 23
   5 Creating a Network ................................................................................................... 27
      5.1 Creating a virtual machine network ................................................................. 27
   6 Creating a Server Pool .............................................................................................. 29
      6.1 Creating a server pool ........................................................................................ 29
      6.2 Adding Oracle VM Servers to a server pool ...................................................... 30
   7 Creating a Storage Repository .................................................................................. 31
      7.1 Creating a storage repository on a file server .................................................... 31
      7.2 To create a storage repository on a storage array .......................................... 32
      7.3 Importing resources to a storage repository ..................................................... 33
         7.3.1 Importing a virtual appliance .................................................................. 33
         7.3.2 Importing a virtual machine template ..................................................... 33
         7.3.3 Importing an ISO file ............................................................................... 33
   8 Creating a Virtual Machine ....................................................................................... 35
      8.1 Creating a virtual machine from a virtual appliance ......................................... 35
      8.2 Creating a virtual machine from a template .................................................... 35
      8.3 Creating a virtual machine from an ISO .......................................................... 35
      8.4 Starting a virtual machine ................................................................................ 37
II CLI Command Reference ............................................................................................. 39
   A CLI Command Reference ....................................................................................... 41
      A.1 abort Job ........................................................................................................ 41
A.2 ackEvent ................................................................. 41
A.3 addAccessHost ..................................................... 42
A.4 addAdminServer ................................................... 43
A.5 addPolicyServer .................................................. 44
A.6 addRefreshServer ................................................ 45
A.7 add BondPort ..................................................... 46
A.8 add FileSystem .................................................... 47
A.9 add PhysicalDisk ................................................ 48
A.10 add Port .......................................................... 48
A.11 add Server ........................................................ 49
A.12 add ServerPool .................................................. 51
A.13 add StorageInitiator ........................................... 52
A.14 add Tag ............................................................ 53
A.15 add VlanInterface .............................................. 54
A.16 add Vm ............................................................. 55
A.17 add Vnic ........................................................... 56
A.18 changeServerAgentPassword ................................ 57
A.19 checkUpToDate .................................................. 58
A.20 clearVmAllRcvdMessages .................................... 58
A.21 clearVmAllSentMessages .................................... 59
A.22 clearVmRcvdMessage ......................................... 60
A.23 clearVmSentMessage .......................................... 61
A.24 clone Vm .......................................................... 62
A.25 cloneCdToRepo .................................................. 63
A.26 clonePdToPd ...................................................... 64
A.27 clonePdToRepo ................................................... 65
A.28 clonePdToStorageArray ..................................... 67
A.29 cloneVdToPd ...................................................... 68
A.30 cloneVdToRepo .................................................. 69
A.31 createVmFromAssembly (Deprecated) ..................... 70
A.32 createVmFromVirtualAppliance ................................ 71
A.33 createVmFromVirtualApplianceVm .......................... 72
A.34 create AccessGroup ............................................ 73
A.35 create AntiAffinityGroup ..................................... 75
A.36 create BondPort ................................................ 76
A.37 create CpuCompatibilityGroup ............................... 77
A.38 create FileServer ............................................... 78
A.39 create FileSystem ............................................. 80
A.40 create Network .................................................. 82
A.41 create PhysicalDisk ............................................ 83
A.42 create Repository ............................................... 84
A.43 create RepositoryExport ..................................... 86
A.44 create ServerController ...................................... 87
A.45 create ServerPool ............................................. 88
A.46 create ServerPoolNetworkPolicy ............................ 90
A.47 create ServerUpdateGroup ................................... 91
A.48 create ServerUpdateRepository ............................. 92
A.49 create StorageArray .......................................... 94
A.50 create Tag ....................................................... 97
A.51 create VirtualDisk ............................................. 97
A.52 create VlanInterface .......................................... 99
A.53 create Vm ......................................................... 100
A.54 create VmCloneCustomizer .................................. 105
A.55 create VmCloneNetworkMapping ............................ 106
A.164 sendVmMessage ................................................................. 230
A.165 set .............................................................................. 231
A.166 setArchiveConfig ............................................................. 232
A.167 setDbBackupConfig ............................................................ 233
A.168 setMaintenanceMode ......................................................... 234
A.169 setStatsConfig ................................................................. 235
A.170 setVnicMacAddrRange ......................................................... 236
A.171 show ................................................................. 237
A.172 showallcustomcmds .......................................................... 239
A.173 showclisession ............................................................... 239
A.174 showcustomcmds ............................................................. 240
A.175 showobjtypes ................................................................. 241
A.176 showversion ................................................................. 242
A.177 start ................................................................. 242
A.178 stop ................................................................. 243
A.179 suspend ................................................................. 244
A.180 takeOwnership ............................................................... 244
A.181 upgrade ................................................................. 245
A.182 validate ................................................................. 246
Glossary ............................................................................. 249
Index .............................................................................. 253
Preface

The Oracle VM Manager Command Line Interface User's Guide is your reference to the Oracle VM Manager Command Line Interface (CLI). The CLI is intended to offer the same features as the Oracle VM Manager graphical user interface, so you can manage your Oracle VM environment using the CLI and not need to use the Oracle VM Manager graphical user interface, if you so choose. The CLI connects to an Oracle VM Manager instance from either the Oracle VM Manager host, or another client computer using an ssh connection. You can script CLI commands using any programming or scripting language of your choice. A few Expect scripts are provided for your reference to get you started. This Guide gives you an overview of how to connect to the CLI, examples on how to use it to set up your environment, and a complete syntax reference.

Audience

This document is intended for Oracle VM administrators with privileged access to the physical and virtual resources of the Oracle VM environment. This guide assumes that you have an in depth knowledge of Oracle VM (see the Oracle VM Manager User's Guide), and that you are familiar with Oracle Linux system administration and Linux command line operation.

Related Documents

For more information, see the following documents in the Oracle VM documentation set:

• Oracle VM Release Notes
• Oracle VM Installation and Upgrade Guide
• Oracle VM Concepts Guide
• Oracle VM Manager Getting Started Guide
• Oracle VM Manager User's Guide
• Oracle VM Manager Command Line Interface User's Guide
• Oracle VM Administrator's Guide
• Oracle VM Paravirtual Drivers for Microsoft Windows Guide
• Oracle VM Web Services API Developer's Guide
• Oracle VM Security Guide
• Oracle VM Licensing Information User Manual

You can also get the latest information on Oracle VM by going to the Oracle VM Web site:


Command Syntax

Oracle Linux command syntax appears in monospace font. The dollar character ($), number sign (#), or percent character (%) are Oracle Linux command prompts. Do not enter them as part of the command. The following command syntax conventions are used in this guide:
<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>backslash \</td>
<td>A backslash is the Oracle Linux command continuation character. It is used in command examples that are too long to fit on a single line. Enter the command as displayed (with a backslash) or enter it on a single line without a backslash:</td>
</tr>
<tr>
<td>braces { }</td>
<td>Braces indicate required items:</td>
</tr>
<tr>
<td>brackets [ ]</td>
<td>Brackets indicate optional items:</td>
</tr>
<tr>
<td>ellipses ...</td>
<td>Ellipses indicate an arbitrary number of similar items:</td>
</tr>
<tr>
<td>italics</td>
<td>Italic type indicates a variable. Substitute a value for the variable:</td>
</tr>
<tr>
<td>vertical line</td>
<td>A vertical line indicates a choice within braces or brackets:</td>
</tr>
<tr>
<td>forward slash /</td>
<td>A forward slash is used to escape special characters within single or double quotes in the Oracle VM Manager Command Line Interface, for example:</td>
</tr>
</tbody>
</table>

## Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td>italic</td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td>monospace</td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>

## Documentation Accessibility


## Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit [http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info](http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info) or visit [http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs](http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs) if you are hearing impaired.
Chapter 1 What's New in the Oracle VM Manager Command Line Interface (CLI)?

This section lists the syntax changes in the Oracle VM Manager Command Line Interface (CLI) from previous releases. Included in this section are any changes or deletions to existing commands and options, as well as new commands or options. This section may be useful when migrating your CLI scripts from previous releases of the CLI.

1.1 CLI Changes in Release 3.4

This section lists the changes and new features to the CLI in Release 3.4.

1.1.1 CLI Modifications

This section lists the modifications made to existing CLI commands.

Table 1.1 CLI Modifications in Release 3.4

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| N/A           | You can set some command options to be empty strings. You may want to do this to clear Oracle VM Server roles, network roles, clear NTP servers, or clear an object's description. No syntax changes have been made in commands, but the logic behind string options has changed. Where these changes are appropriate, they are noted in the commands in Appendix A, CLI Command Reference. For example, to clear Oracle VM Server roles you would use:   
|               | edit Server name=MyServer roles=""                                                                                                       |
| N/A           | You can mask attribute values on screen, as they are entered, by appending an asterisk (*) to the attribute name. This security measure is described in Section 2.6.2, “Masking Sensitive Data”. For example:   
|               | OVM> sendVmMessage Vm name=MyVM key=com.oracle.linux.root-password message=*...

<table>
<thead>
<tr>
<th>AccessGroup</th>
<th>The edit AccessGroup command now includes a nameOnArray option to change the name on the storage array.</th>
</tr>
</thead>
</table>
| Assembly      | The Assembly object, and the commands related to the Assembly object, have been deprecated. From this release, you should instead use the new VirtualAppliance object and associated commands. The commands related to the Assembly object are honored by the CLI, but there will be no future improvements to those commands. The following commands have been deprecated:   
|               | • The Assembly, AssemblyVirtualDisk, and AssemblyVm options of the list command.                                                               |
|               | • The Assembly, AssemblyVirtualDisk, and AssemblyVm options of the show command.                                                              |
|               | • The Assembly option of the delete command.                                                                                                  |
|               | • The Assembly option of the refresh command.                                                                                                 |
|               | • The Assembly option of the getDescriptor command.                                                                                           |
|               | • The edit Assembly (Deprecated) command.                                                                                                     |
### CLI New Features

This section lists the new commands introduced in Release 3.4.

#### Table 1.2 CLI New Features in Release 3.4

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>A new <code>setArchiveConfig</code> command configures how Oracle VM Manager manages archived statistics.</td>
</tr>
<tr>
<td>N/A</td>
<td>A new <code>getArchiveConfig</code> command shows the configuration for managing archived statistics.</td>
</tr>
<tr>
<td>N/A</td>
<td>A new <code>showClisession</code> command lists the CLI session options set using the <code>set</code> command.</td>
</tr>
<tr>
<td>N/A</td>
<td>The <code>set</code> command now includes an <code>AlphabetizeAttributes</code> option to display the output of the <code>show</code> command in alphabetical order.</td>
</tr>
<tr>
<td>Event</td>
<td>The <code>getEventListByQuery</code> command lists events for objects using a query to narrow results.</td>
</tr>
<tr>
<td>Event</td>
<td>The <code>getTriageEventSeverityList</code> command lists the highest severity event for one or more objects.</td>
</tr>
<tr>
<td>Manager</td>
<td>A new <code>getManagerTime</code> command displays the Oracle VM Manager time.</td>
</tr>
<tr>
<td>Manager</td>
<td>There are two new commands to list and set Oracle VM Manager automated database backup configuration parameters. The new commands are:</td>
</tr>
<tr>
<td></td>
<td>• <code>getDbBackupConfig</code> : Lists the values for configuration parameters related to database backup configuration.</td>
</tr>
<tr>
<td></td>
<td>• <code>setDbBackupConfig</code> : Sets the values for configuration parameters related to database backup configuration.</td>
</tr>
<tr>
<td>StorageArray</td>
<td>The <code>edit StorageArray</code> command now has a <code>lipScan</code> option that allows you to enable a Loop Initialization (LIP) for an Unmanaged FibreChannel Storage Array.</td>
</tr>
<tr>
<td>ServerPool, Vm</td>
<td>The <code>startPolicy</code> command now includes a <code>BALANCE_SERVER</code> option that can be used when defining the default start-up policy for all of your virtual machines. This option applies to the following commands:</td>
</tr>
<tr>
<td></td>
<td>• <code>create ServerPool</code></td>
</tr>
<tr>
<td></td>
<td>• <code>edit ServerPool</code></td>
</tr>
<tr>
<td>Object Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Statistic</strong></td>
<td>The <code>getStatListByQuery</code> command lists statistic for objects using a query to narrow results.</td>
</tr>
<tr>
<td><strong>Statistic, Server, Vm</strong></td>
<td>There are four new commands to list statistics for an Oracle VM Server or virtual machine. The new commands are:</td>
</tr>
<tr>
<td></td>
<td>- <code>getAverageStat</code>: Lists the average value of a statistic type for an object during a time range.</td>
</tr>
<tr>
<td></td>
<td>- <code>getLatestStat</code>: Lists the latest value of a statistic type for an object.</td>
</tr>
<tr>
<td></td>
<td>- <code>getStatListForList</code>: Lists the latest value of a statistic type for a list of objects.</td>
</tr>
<tr>
<td></td>
<td>- <code>getStatList</code>: Lists statistic types for an object.</td>
</tr>
<tr>
<td><strong>VirtualAppliance</strong></td>
<td>A new VirtualAppliance object has been added. This should be used instead of the Assembly object, which has been deprecated in this release. The new commands related to the VirtualAppliance object are:</td>
</tr>
<tr>
<td></td>
<td>- New <code>VirtualAppliance</code>, <code>VirtualApplianceVirtualDisk</code>, and <code>VirtualApplianceVm</code> options of the <code>list</code> command.</td>
</tr>
<tr>
<td></td>
<td>- New <code>VirtualAppliance</code>, <code>VirtualApplianceVirtualDisk</code>, and <code>VirtualApplianceVm</code> options of the <code>show</code> command.</td>
</tr>
<tr>
<td></td>
<td>- A new <code>VirtualAppliance</code> option of the <code>delete</code> command.</td>
</tr>
<tr>
<td></td>
<td>- A new <code>VirtualAppliance</code> option of the <code>refresh</code> command.</td>
</tr>
<tr>
<td></td>
<td>- A new <code>VirtualAppliance</code> option of the <code>getDescriptor</code> command.</td>
</tr>
<tr>
<td></td>
<td>- The <code>edit VirtualAppliance</code> command.</td>
</tr>
<tr>
<td></td>
<td>- The <code>edit VirtualApplianceVirtualDisk</code> command.</td>
</tr>
<tr>
<td></td>
<td>- The <code>edit VirtualApplianceVm</code> command.</td>
</tr>
<tr>
<td></td>
<td>- The <code>createVmFromVirtualAppliance</code> command.</td>
</tr>
<tr>
<td></td>
<td>- The <code>createVmFromVirtualApplianceVm</code> command.</td>
</tr>
<tr>
<td></td>
<td>- The <code>importVirtualAppliance</code> command.</td>
</tr>
<tr>
<td></td>
<td>- The <code>exportVirtualAppliance</code> command.</td>
</tr>
<tr>
<td><strong>Vm</strong></td>
<td>The <code>create Vm</code> and <code>edit Vm</code> commands have a new <code>viridian</code> option that enables the exposure of Windows virtualization compatible entitlements to Microsoft Windows guest operating systems.</td>
</tr>
<tr>
<td><strong>Vm</strong></td>
<td>The <code>create Vm</code> and <code>edit Vm</code> commands have a new <code>restartActionOnCrash</code> option to set the action to perform in the case where a virtual machine crashes.</td>
</tr>
</tbody>
</table>
### CLI Object Attribute Changes

This section lists the changes to the attributes of CLI objects in Release 3.4. The attribute changes affect the output of the `show` command, and the options available with `create object` and `edit object` commands.

#### Table 1.3 CLI Object Attributes Changes in Release 3.4

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Attributes Added</th>
<th>Attributes Removed</th>
<th>Attributes Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>PhysicalDisk</td>
<td><code>absolutePaths</code></td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td><code>storageArray</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Repository</td>
<td><code>apparentSize</code></td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td><code>oversubscription</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ServerAbility</td>
<td><code>vmRestartActions</code></td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td><code>vmLiveStorageMigration</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>abilities</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ServerPool</td>
<td><code>BALANCE_SERVER</code></td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>StorageArray</td>
<td><code>lipScan</code></td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>VirtualApplianceVirtualDisk, which replaces the deprecated AssemblyVirtualDisk object.</td>
<td>N/A</td>
<td>N/A</td>
<td>Display names of attributes have changed: From <code>Populated Size</code> to <code>Populated Size (Bytes)</code>. From <code>Capacity</code> to <code>Capacity (Bytes)</code>.</td>
</tr>
<tr>
<td>VirtualDisk</td>
<td><code>absolutePath</code></td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td><code>mountedPath</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vm</td>
<td><code>vmConfigFileAbsolutePath</code></td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td><code>vmConfigFileMountedPath</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>restartActionOnCrash</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>viridian</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>BALANCE_SERVER</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 2 Introduction to the Oracle VM Manager Command Line Interface (CLI)

The Oracle VM Manager Command Line Interface (CLI) provides a command line interface to communicate with Oracle VM Manager. You can use the CLI to perform many of the same functions as Oracle VM Manager, such as managing all your server pools and guest virtual machines. The CLI commands can be scripted to enable flexibility to help you deploy and manage your Oracle VM environment.

The CLI is installed when you install Oracle VM Manager, so you must have a working copy of Oracle VM Manager to use the CLI. The architecture of the CLI is shown in Oracle VM Architecture in the Oracle VM Concepts Guide.

When you make changes to the Oracle VM environment using the CLI, these changes are reflected in real time in the Oracle VM Manager Web Interface.

The Oracle VM Manager Web Interface includes additional logic over the CLI to make sure that actions performed within Oracle VM Manager do not result in configurations that may cause runtime errors. This additional logic is not available within the CLI, which provides greater flexibility, but requires a deeper understanding of Oracle VM object relationships.

The CLI does not replace the Oracle VM Utilities. The Oracle VM Utilities are complementary to the CLI. For information on the Oracle VM Utilities, see the Oracle VM Administrator's Guide.

2.1 Starting and Stopping the CLI

The CLI is automatically started when you first install Oracle VM Manager. You can also stop and start it separately to Oracle VM Manager as required. To use the CLI, the Oracle VM Manager service should first be started. See Starting and Stopping Oracle VM Manager for information on starting and stopping Oracle VM Manager.

When the CLI is installed (as part of Oracle VM Manager), it is set to automatically start when the operating system starts. If you want to disable this, enter:

```
# chkconfig ovmcli off
```

To manually start or stop the CLI, and to verify its status, log in as the root user, and use the syntax:

```
/sbin/service ovmcli [start|stop|status|restart]
```

For example, to start the CLI:

```
# /sbin/service ovmcli start
```

To stop the CLI:

```
# /sbin/service ovmcli stop
```

To restart the CLI:

```
# /sbin/service ovmcli restart
```

To check the status of the CLI:

```
# /sbin/service ovmcli status
```
Connecting to the CLI

# /sbin/service ovmcli status

The `status` option returns whether the CLI service is running or stopped.

If the Oracle VM Manager host computer runs a full graphical desktop environment, you can also use the Services dialog to start and stop the CLI. From the Applications menu, select System Settings > Server Settings > Services. Or by running the following command in a terminal to display the Services dialog:

# /usr/bin/system-config-services

In the Service Configuration dialog, select `ovmcli` to check the status, and start or stop it.

2.2 Connecting to the CLI

Multiple CLI connections can be made to a single instance of Oracle VM Manager at any time. The connection to the CLI is an SSH connection. To connect to the CLI, use an SSH client or command line interface and connect to the Oracle VM Manager host using the syntax:

```
ssh -l manager_username {manager_IP | manager_hostname} -p port
```

The default port for the CLI is 10000.

For example:

```
$ ssh -l admin 10.172.76.146 -p 10000
```

To connect to the CLI from the Oracle VM Manager host, enter:

```
$ ssh -l admin localhost -p 10000
```

You can also use the abbreviated connection syntax (without the `-l` option), for example:

```
$ ssh admin@localhost -p 10000
```

**Tip**

To keep your ssh session from disconnecting due to inactivity, you can use the ssh ServerAliveInterval option to send a null packet to the CLI to keep the connection alive. You can use this option, either on the command line when you enter the ssh command, or in the ~/.ssh/config file, for example to use this on the command line, enter:

```
$ ssh admin@localhost -p 10000 -o ServerAliveInterval=40
```

The interval length is in seconds, so this example will keep the connection alive for 40 seconds. Setting this value to 0 disables the feature. The syntax to use when adding this to the ~/.ssh/config file is:

```
Host *manager_hostname
  ServerAliveInterval 40
```

When you establish a connection to the CLI, you are prompted to enter a password for the Oracle VM Manager admin user. Enter the password and the CLI prompt is displayed, ready for you to begin entering Oracle VM CLI commands.

```
Using username "admin".
admin@10.172.76.146's password: password
```
To exit the CLI, enter `exit`, or end the SSH session.

**2.3 SSH Host Keys**

On some operating systems, when you first log in to the CLI, you may be prompted to add the key fingerprint of the Oracle VM Manager host to the `~/.ssh/known_hosts` file, for example:

```
$ ssh -l admin hostname -p 10000
The authenticity of host 'hostname (IP_address)' can't be established.
DSA key fingerprint is fingerprint.
Warning: Permanent adding 'hostname' (DSA) to the list of known hosts.
admin@hostname's password:
```

If you want to avoid this message and have host keys automatically added to the `known_hosts` file, you can turn off strict checking of SSH host keys using the following command:

```
$ ssh -o 'StrictHostKeyChecking no' admin@hostname
```

If you have upgraded or reinstalled Oracle VM Manager, you may be prompted that the host identification has changed when connecting to the CLI, for example:

```
$ ssh -l admin hostname -p 10000
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

To clear this message and allow connections to the CLI, use the `ssh-keygen` utility to remove the entry for the Oracle VM Manager host, for example:

```
$ ssh-keygen -R hostname:10000
```

Alternatively, edit the `~/.ssh/known_hosts` file and remove the entry for the Oracle VM Manager host. Connect to the CLI again and you are prompted to add the new fingerprint if strict checking of SSH host keys is enabled.

**2.4 Key-Based Authentication**

You can use public key-based SSH authentication to connect to the CLI. When you set up key-based authentication, you can log in to the CLI without being prompted for a password. Using key-based authentication does not effect the existing authentication mechanism using a username and password.
Key-Based Authentication

Keys are set up between a local system and the Oracle VM Manager host. After the keys are set up, you log in for the first time with the CLI admin password, as you normally would, then a connection is established using public key authentication and the channel is identified to the CLI Server by the client IP address and user name. Subsequent log ins do not require the password, for as long as the channel remains open. If the channel is closed, or the admin user's password is changed, key-based authentication is terminated and you are required to enter the admin user's password again. To reestablish the connection using key-based authentication, log in again using the standard SSH connection, enter the admin user’s password, and the channel is opened again.

To set up key-based authentication:

1. Make sure the ssh-agent is running on your local host:

```bash
$ eval `ssh-agent`
Agent pid number
```

If the ssh-agent is not running, you may encounter the following error when you perform the next step:

```
Could not open a connection to your authentication agent.
```

2. On your local host, generate a public/private key pair to log in to the CLI. Make sure you remember the passphrase that you enter.

```bash
$ ssh-keygen -t rsa -f ~/.ssh/admin
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase): passphrase
Enter same passphrase again: passphrase
Your identification has been saved in /user/.ssh/admin.
Your public key has been saved in /user/.ssh/admin.pub.
The key fingerprint is:
fingerprint user@hostname
```

The two keys are generated in ~/.ssh/: admin (the private key) and admin.pub (the public key).

3. Add the private key to the authentication agent, using the same passphrase you used to create the key pair, for example:

```bash
$ ssh-add ~/.ssh/admin
Enter passphrase for /home/user/.ssh/admin: passphrase
Identity added: /home/user/.ssh/admin (/home/user/.ssh/admin)
```

4. Copy the public key to the Oracle VM Manager host, for example:

```bash
$ scp ~/.ssh/admin.pub oracle@hostname:/home/oracle/.ssh/
```

Where hostname is the hostname of the Oracle VM Manager host. Make sure you do the copy as the oracle user.

5. Log into the Oracle VM Manager host as the oracle user and append the admin.pub public key to the CLI authorized file (ovmcli_authorized_keys).

```bash
$ ssh oracle@hostname
$ cd /home/oracle/.ssh/
$ cat admin.pub >> ovmcli_authorized_keys
$ exit
```

Where hostname is the hostname of the Oracle VM Manager host.

6. From the local machine, log in to the CLI using the command:

```bash
$ ssh -l admin hostname -p 10000
```
You are requested to enter the admin user's password. Enter it.

```
admin@hostname's password: password
```

Subsequent log ins use the newly established channel and do not require a password.

For security reasons, the channel for public key authentication expires after a designated period of time. See Section 2.10, "Configuring the Oracle VM CLI" for information on how to change the public key authentication expiry time.

### 2.5 Using SSH to Run Background Processes

If you choose to run CLI commands in the background via the standard SSH interface, it is possible that keyboard breaks interrupt running processes. This is a standard behaviour if the TTY setting has been set to `sane`, and is remedied by providing the SSH command with the `-n` option to automatically send keyboard input to `/dev/null`. This type of command can only be used with key-based authentication, since at the point that it is run, the command is run in the background so you can no longer interact with it.

```
$ ssh -l admin localhost -p 10000 -n "importVirtualAppliance Repository name=MyRepository url=http://example.com/myappliance.ova" &
```

Note that using the `-n` flag does not make sense if you intend to maintain an interactive session within the CLI.

### 2.6 Using the Oracle VM CLI and Getting Help

Enter `?` or `help` to display help on a token. You can also enter `?` after a token to display the possible options based on context. For example, if you want to display information about an Oracle VM Server, you can work your way through the command options to find the commands to perform this action.

```
OVM>?
add
create
deleteedit
embeddedcreate
embeddeddelete
embeddededit
exit
help
list
Perhaps this is the command? Let's drill down further.
remove
set
```

**Note**

To keep the output to a minimum in the examples in this book, we have set the output mode to sparse using the following command:

```
OVM> set OutputMode=Sparse
```

Your output may vary depending on which setting you use for this command; see Section A.165, "set" for more information.

To find the command to list Oracle VM Servers, start with the `?` option and work your way through the commands:
Using the Oracle VM CLI and Getting Help

```
show
showallcustomcmds
showclisession
showcustomcmds
showobjtypes
showversion

OVM> list ?
  AccessGroup
  AntiAffinityGroup
  Assembly
  ...
  Port
  Repository
  RepositoryExport

Server
  This looks like the command to use to list Oracle VM Servers
  ServerController
  ServerPool
  ServerPoolNetworkPolicy
  ...
  VmDiskMapping
  Vnic
  VolumeGroup

OVM> list Server
  No more options can be entered so the results are automatically displayed
  id:00:e0:81:4d:5f:2f:00:e0:81:4d:29:ee:00:e0:81:4d
  name:MyServer1
  id:00:e0:81:4d:5e:16:00:e0:81:4d:5e:17:ff:ff:ff:ff
  name:MyServer2
  id:00:e0:81:4d:40:16:00:e0:81:4d:40:17:ff:ff:ff:ff
  name:MyServer3

OVM>
```

Now you have a list of the Oracle VM Servers, you can display information about them with another command. To find the command to display information about an Oracle VM Server, drill down again through the commands to find the most appropriate command using the `?` option:

```
OVM> ?
  add
  create
  delete
  edit
  embeddedcreate
  embeddeddelete
  embeddededit
  exit
  help
  list
  remove
  set
  show
    This looks like the command to use to show information
  showallcustomcmds  commands available for all objects
  showclisession
  showcustomcmds     commands specific to an object (requires object as argument)
  showobjtypes
  showversion

OVM> show ?
  AccessGroup
  AntiAffinityGroup
  Assembly
  ...
  Port
  Repository
  RepositoryExport

Server
  This looks like the command to use to show information about an Oracle VM Server
  ServerController
  ServerPool
  ServerPoolNetworkPolicy
```
If you have forgotten the name of the Oracle VM Server, use the up arrow to scroll through the history until you see the `list Server` command and press `Enter`. Then use the `show Server name=` option to display information about an Oracle VM Server.

The CLI is a self-learning tool; built in help and tab auto-completion guide you when working with the commands. The following commands assist you to use the CLI.

### Table 2.1 Helpful CLI commands

<table>
<thead>
<tr>
<th>Command/Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Context sensitive help, for example, <code>show ?</code>, <code>clone ?</code>. If you do not know the format of a command, enter the command followed by <code>?</code> to see the options for that command. Enter <code>?</code> on its own to see a list of all the top level commands.</td>
</tr>
<tr>
<td>help</td>
<td>Displays the syntax to use for the top level commands.</td>
</tr>
<tr>
<td>showallcustomcmds</td>
<td>Displays a list of the all custom commands for all object types.</td>
</tr>
<tr>
<td>showclisession</td>
<td>Displays the CLI session settings. These settings are set using the <code>set</code> command.</td>
</tr>
<tr>
<td>showcustomcmds [object type]</td>
<td>Displays a list of the custom commands for a specific object type provided as a parameter.</td>
</tr>
<tr>
<td>showobjtypes</td>
<td>Displays a list of the object types.</td>
</tr>
<tr>
<td>tab completion</td>
<td>Press the Tab key to auto-complete the command.</td>
</tr>
<tr>
<td>history</td>
<td>Use the up or down arrow keys to step through the history of commands entered in the current session. Up to 50 commands are listed.</td>
</tr>
<tr>
<td>masking sensitive data</td>
<td>Append an asterisk (*) to an attribute name to mask the content following the equals sign (=) when setting the value of an attribute. See <a href="#">Masking Sensitive Data</a> for more information on this.</td>
</tr>
</tbody>
</table>

You can configure the end of line characters used by your SSH client, for example, if your SSH client adds a line feed (double spacing) to the end of a line, you can set the endline characters to `CR`. Set the end of line characters using the `set` command.

You can configure the output mode to define how the CLI returns results, for example in plain text or in XML. Set the output mode using the `set` command.
Return Status Values

Use the  showclisession command to show the values for the options set with the Section A.165, “set” command.

The values you enter for parameters are case sensitive, unless explicitly stated in this Guide. For example, entering name=MyServer is not the same as entering name=myserver. The CLI treats these parameter strings as case sensitive, and are considered different.

Special characters are considered any of these: ",',?\,/<>. You can escape special characters within a set of quotes to make sure they are treated as a literal string using a / (forward slash) before the character. For example:

```
OVM> create Tag name=MyTag description="HR's VMs from http://example.com/vms/" id:0004fb0000260000b351e52e3abbe192 name:MyTag
OVM> show Tag name=MyTag
  Name = MyTag
  Id = 0004fb0000260000b351e52e3abbe192
  description = 'HR's VMs from http://example.com/vms/'<Delete/>
```

Parameters are unique for each time you run a command. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence. Therefore, a command similar to the following succeeds, but the values of these repeated parameters override each other:

```
OVM> discoverServer ipAddress=server1 ipAddress=server2 takeOwnership=No\takeOwnership=Yes password=** password=******
Command: discoverServer ipAddress=server1 ipAddress=server2 takeOwnership=No
takeOwnership=Yes password=** password=******
Status: Success
Time: 2013-12-23 00:34:38,398 PST
JobId: 138778765552
OVM> list server
Command: list server
Status: Success
Time: 2013-12-23 00:34:43,602 PST
  name:server2
```

In this example, only one Oracle VM Server is discovered. The second parameter in each instance of the command overrides the first. Therefore, server2 is discovered, and ownership is set to Yes, and the password in this instance is the second one specified.

If a command is issued, but no changes to an object are performed, a success message is displayed. For example, if you change the name of an object to the same name, Status: Success is returned and displayed.

### 2.6.1 Return Status Values

Operations that trigger job creation within Oracle VM Manager, such as create or modify operations, always return the status of the job. The status is referred to as the Return Status. If a job is successful, the status returned is set to Success. If a job fails, the status returned is set to Failure.

Job status is obtained by consistently polling Oracle VM Manager with the generated Job ID to query the job status. For operations that may take too long, the CLI may timeout the polling period. In this case, the status returned is set to Running to indicate that the job is still running. In this case, you may need to query the job status manually before continuing with any other operations.

Since all commands that result in some form of change within Oracle VM Manager trigger the creation of a job, these commands also return the job ID in the response output. The value for this appears in the JobId
2.6.2 Masking Sensitive Data

In some scenarios you may need to mask sensitive data as it is entered into the CLI to avoid it displaying to screen. This functionality is applied by default on some known commands such as the `discoverServer`, where the password value is automatically substituted with a series of asterisk (`*`) characters on the screen as it is typed. However, in other situations, such as when sending messages to VMs, there are some additional values that you may wish to obscure on the screen that would not be hidden by default. Masking functionality has been built into the CLI to allow you to automatically mask the data for any attribute by simply appending the asterisk (`*`) character to the attribute name before entering its value. This is illustrated in the example below:

```
OVM> sendVmMessage Vm name=MyVM key=com.oracle.linux.root-password message*=******* log=No
```

In the example above, the `message` is purposefully masked and the actual value is substituted on screen with `*******`. For additional security, in this example, we also set the `log` parameter to No, to prevent the message from being logged.

The masking function can be applied to any attribute where a value is entered for a command, with the exception of the `name` or `id` attributes. For instance, it is possible to hide the `description` attribute as it is entered on screen:

```
OVM> edit Server name=MyServer description*=***********
```

The masking feature expects a single asterisk following the attribute name. Any attempt to enter multiple asterisks or other characters following the first asterisk causes the validation to fail and the attribute name is unrecognized.

### 2.7 Passing in a CLI Command at the Command Line

If you want to run the CLI and pass in a simple command, append the command in quotes after the SSH login credentials, for example:

```
# ssh admin@hostname -p 10000 "list Server"
```

You can submit multiple commands using a semicolon (`;`) as the command delimiter, for example:

```
# ssh admin@hostname -p 10000 "list Server; show Server name=MyServer"
```

If you have enabled key-based authentication, you are not prompted for the password to access the CLI, and the results are displayed. If you have not, you must enter the CLI password before the results are returned. See Section 2.4, “Key-Based Authentication” for information on setting up key-based authentication. You can also use any other programming language to write scripts using the CLI as discussed in Section 2.9, “Integrating the CLI into Your Applications”.

### 2.8 Sample Scripts

There are a number of shell scripts provided with Oracle VM to help you use the CLI. If a password is required for any of these scripts, you are prompted at the command line to enter it. The shell scripts use key-based authentication, which is available after the first log in to the CLI.
Sample Scripts

These scripts are located in:

/u01/app/oracle.ovm-manager-3/ovm_cli/scripts

Table 2.2 Sample scripts

<table>
<thead>
<tr>
<th>Script Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ackAllEvents</td>
<td>A shell script that acknowledges all events that a user can acknowledge. This script does not acknowledge system events.</td>
</tr>
<tr>
<td></td>
<td>This script uses localhost as the default host. If you run this script from another host, you can specify the Oracle VM Manager location using the syntax:</td>
</tr>
<tr>
<td></td>
<td>ackAllEvents -l user@host</td>
</tr>
<tr>
<td>inventory</td>
<td>A shell script that displays an inventory of all objects managed by Oracle VM Manager.</td>
</tr>
</tbody>
</table>

There are also a set of shell scripts provided to get you started scripting in the CLI. These scripts are located in:

/u01/app/oracle.ovm-manager-3/ovm_cli/scripts/createdeletescripts

To use any of these scripts, you should first overwrite the default values in the associated *.properties file with values that match those of your environment. Each parameter within the associated properties file is commented to provide guidelines on the information that must be provided within these files. Additional information on how to run the scripts are contained as comments at the start of each script.

Table 2.3 Sample create/delete scripts

<table>
<thead>
<tr>
<th>Script Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>create-fc-based-VM.sh</td>
<td>A shell script that sets up a complete Oracle VM environment, including storage, server pool, networking, virtual machine resources and a virtual machine, using fibre channel-based storage.</td>
</tr>
<tr>
<td></td>
<td>To use this script, edit the fc-based-VM.properties file in the same directory and change the parameters to suit your environment.</td>
</tr>
<tr>
<td>delete-fc-based-VM.sh</td>
<td>A shell script that deletes the set up created using the create-fc-based-VM.sh script.</td>
</tr>
<tr>
<td></td>
<td>To use this script, edit the fc-based-VM.properties file in the same directory and change the parameters to suit your environment.</td>
</tr>
<tr>
<td>create-iscsi-based-VM.sh</td>
<td>A shell script that sets up a complete Oracle VM environment, including storage, server pool, networking, virtual machine resources and a virtual machine, using ISCSI-based storage.</td>
</tr>
<tr>
<td></td>
<td>This script uses a fibre channel LUN for the server pool's clustered file system. The Oracle VM Server must have access to an unmanaged fibre channel storage array. This script creates a storage array using the Oracle NetApp Filer plug-in.</td>
</tr>
<tr>
<td></td>
<td>To use this script, edit the iscsi-based-VM.properties file in the same directory and change the parameters to suit your environment.</td>
</tr>
</tbody>
</table>
### Script Name | Purpose
--- | ---
**delete-iscsi-based-VM.sh** | A shell script that deletes the set up created using the `create-iscsi-based-VM.sh` script.

To use this script, edit the `iscsi-based-VM.properties` file in the same directory and change the parameters to suit your environment.

**create-nfs-based-VM.sh** | A shell script that sets up a complete Oracle VM environment, including storage, server pool, networking, virtual machine resources and a virtual machine, using NFS-based storage.

To use this script, edit the `nfs-based-VM.properties` file in the same directory and change the parameters to suit your environment.

**delete-nfs-based-VM.sh** | A shell script that deletes the set up created using the `create-nfs-based-VM.sh` script.

To use this script, edit the `nfs-based-VM.properties` file in the same directory and change the parameters to suit your environment.

**common.sh** | A shell script that contains the common functions used by other scripts in this directory. Do not run this script directly, instead call it from other scripts. You can see a list of the common functions in this script in Table 2.4, “Sample scripts common functions”.

To customize any of these scripts for your environment, copy them to a directory outside of the Oracle VM Manager installation tree, edit the script with a text editor and change the variables to suit your environment, or extend further to provide additional functionality.

When you run any of these scripts, the output is displayed to the screen; spool it to a file or other process or script as required.

The **common.sh** file provides some common functions used by the sample scripts. The functions available in this script are listed in Table 2.4, “Sample scripts common functions”.

### Table 2.4 Sample scripts common functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getPass</strong></td>
<td>Prompts for a password. You may want to prompt for an Oracle VM Server, or storage server password. When prompted, the input is not echoed on screen. The password can then be securely used with other CLI commands, such as discoverServer. No password is required for authentication with Oracle VM Manager as this uses certificate authentication.</td>
</tr>
<tr>
<td><strong>validateCLI command</strong></td>
<td>Script continues if the CLI command returns success, and terminates if the command returns a failure.</td>
</tr>
<tr>
<td><strong>runCLI command</strong></td>
<td>Script continues, regardless of the outcome of the CLI command.</td>
</tr>
<tr>
<td><strong>deleteRepo repository</strong></td>
<td>Deletes a storage repository and all of its contents.</td>
</tr>
<tr>
<td><strong>deleteServerpool serverpool</strong></td>
<td>Remove all Oracle VM Servers from a server pool, and deletes the server pool.</td>
</tr>
<tr>
<td><strong>addVnicFromClonedVmToNetwork vm network</strong></td>
<td>Use this command when cloning a virtual machine that includes a VNIC. This command associates the VNIC with a network, so enables the virtual machine to be started.</td>
</tr>
</tbody>
</table>
Performance Monitoring, Script Logging and Debugging

Optional timer functionality is built into the validateCLI and runCLI functions, which are used by all of the sample scripts to initiate connections to the CLI. This allows you to gather useful performance data, when running any of the sample scripts, by accurately timing and logging how long each of the commands within the scripts take to run. This tool also provides a useful log of every single command run for any of the scripts, providing debugging information that can be used to determine script failure or to understand the sequence of commands being issued by the script.

By default, the timer functionality is disabled, however it can be enabled by using the command line switch: `-t filename.txt`. For instance:

```
./create-nfs-based-VM.sh -t report.txt
```

Output generated within the specified file for a script where the timer functionality has been enabled may look similar to the following:

```
TIME(Seconds) Command
18.621        discoverServer ipAddress='10.172.76.80' password=******* takeOwnership=yes
5.587         create serverpool virtualIP='0.0.0.0' clusterEnable=no name='MyServerPool' \
              description='this is nfs pool MyServerPool'
...
```

It is worth noting that output logged for the timer functionality is appended to the file specified. This means that if you re-use the same filename, the file is not overwritten with the new data. Instead, the output is simply appended to the bottom of the file.

If no filename is specified for the timer switch, the output is dumped to STDOUT.

2.9 Integrating the CLI into Your Applications

CLI-based integration is a popular and inexpensive way to integrate Oracle VM into your own applications. It is far easier and faster than using the Web Services API, and for smaller integrations is a very good option.

Since it is possible to configure the CLI to return information in XML format, it is easy to develop applications that are capable of parsing the results of any command issued to the CLI. Set the output mode using the `set` command.

The connection information is cached on the Oracle VM Manager host for 15 minutes, so subsequent calls with the same login credentials are faster than the initial connection.

You can write your own CLI scripts and take advantage of all the rich constructs that the operating system shell provides: variables, looping, conditional execution, parsing, and so on.

An easy way to integrate with Java is to use the `exec` method within the `Runtime` class, then parse the XML output with an XML parser such as SAX.

Equally, scripts may be written using Python along with an XML parser such as LXML. An SSH library such as Paramiko may assist in the development of tools that can be used remotely.

The CLI does not support regular expressions. If you want to use regular expressions, you should use them in your script that calls the CLI.

2.10 Configuring the Oracle VM CLI

When Oracle VM Manager starts, it reads the CLI configuration file. The configuration file is located at:
You can change a number of options in the configuration file. These options are listed in this section.

Before you change any options in the configuration file, you should back up the original and change the permissions of the file to make it writeable:

```
# chmod +w /u01/app/oracle/ovm-manager-3/ovm_cli/config/CLIConfigParams.xml
```

Any changes to this configuration file require a restart of the CLI. See Section 2.1, “Starting and Stopping the CLI” for information on restarting the CLI.

### 2.10.1 SSH Port

By default, SSH connections to the CLI are allowed on port 10000. To change the port on which the CLI accepts connections, edit the `sshPort="10000"` line in the configuration file and change it to the port number you require.

### 2.10.2 Timeout

By default, connections to the CLI time out after 45 minutes. To change the timeout period for connections to the CLI client, edit the `clientInactivityTimeout="45"` line in the configuration file. Set the timeout to be between 1-1440 minutes. For security reasons, there is no option to disable the timeout period. Setting the value to less than 1 or greater than 1440 (1 day) causes the timeout value to revert to the default of 45 minutes.

In addition to this timeout option, you can use the `ssh ServerAliveInterval` option to stop an ssh client from timing out before the set timeout period. See the Tip in Section 2.2, “Connecting to the CLI” for more information on using this option.

### 2.10.3 Case Sensitivity

CLI commands are *not* case sensitive, so you can enter `list vm`, `List VM`, or any other variation in case.

Data values you use for objects in the CLI are case sensitive, so if a virtual machine has a name of `MyVM`, then you must use the same case when identifying it in the CLI and cannot use variations such as `myvm`.

### 2.10.4 Public Key Authentication Expiry

The connection channel for public key authentication expires after a designated period of time, or if Oracle VM Manager is restarted. The default for keeping the channel open is 1 week (10080 minutes). You can modify this setting by editing the `publicKeyAuthChannelTimeout="10080"` option in the configuration file. A value of `-1` keeps the channel open indefinitely.

### 2.10.5 Lock Exceptions

When executing multiple concurrent CLI scripts, you may experience lock exceptions around shared resources such as file systems and storage repositories. By default, connections to the CLI time out after 20 seconds if an object that is being requested by the CLI command is locked. If a command fails due to a lock exception, the command is resubmitted 12 times to see if the lock has become free and the command can be executed. To help you tune the CLI to reduce lock exceptions, you can configure the amount of time a command waits for the lock to become free, and the number of times the command is resubmitted with the following options in the configuration file:
• The lock time out period is defined by the `lockExceptionRetryInterval="20"` option. This option must be a value between 6 and 119 seconds, with the default being 20.

• The number of retry attempts is defined by the `lockExceptionRetryCount="12"` option. This option must be greater than 0.

2.11 CLI Logs

A log of the CLI application is available in:

/u01/app/oracle/ovm-manager-3/domains/ovm_domain/servers/AdminServer/logs/CLI.log

A log of commands submitted to the CLI is available in:

/u01/app/oracle/ovm-manager-3/domains/ovm_domain/servers/AdminServer/logs/CLIAudit.log

This log contains the following information about the command submitted to the CLI:

• Timestamp

• Client IP Address

• Username

• Command

The log files are rotated when the file size reaches 5 MB with up to 10 rotations, in the same way as the other Oracle VM log files.
Part I Using the CLI

The examples used in this section follow closely the format and flow used in the Oracle VM Manager Getting Started Guide. We have provided an abbreviated version of the steps and commands you need to get you started with using the CLI. If you need more information about the steps you are performing in this part, see the corresponding section in the Oracle VM Manager Getting Started Guide for overview information. If you need more detailed information on a step, see the Oracle VM Manager User's Guide.
Chapter 3 Discovering Oracle VM Servers

This section gives you the syntax and examples to discover an Oracle VM Server.

3.1 Discovering an Oracle VM Server

To discover an Oracle VM Server use the following syntax:

```
discoverServer ipAddress=value password=value takeOwnership={ Yes | No }
```

For example:

```
OVM> discoverServer ipAddress=10.172.76.73 password=password takeOwnership=Yes
```

For more information on the syntax and usage, see Section A.60, “discoverServer”.
Chapter 4 Discovering Storage

This section gives you the syntax and examples to use to discover storage using the CLI.

4.1 Discovering a file server

1. To discover a file server use the following syntax:

   ```
   create FileServer plugin=value accessHost=value adminServers=value 
   [refreshServers=value][uniformedExports= {Yes|No}][name=value][
   description=value]
   ```

   For example:

   ```
   OVM> create FileServer plugin="Oracle Generic Network File System" \
   accessHost=10.172.76.125 adminServers="MyServer1.virtlab.info,MyServer2.virtlab.info" \
   name=MyNFSServer
   ```

   For more information on the syntax and usage, see Section A.38, “create FileServer”.

2. After discovering a file server, you should attach an admin server to it so that administrative operations can be performed on the file server by Oracle VM Manager. To add an admin server to the file server, use the syntax:

   ```
   addAdminServer {FileServer | StorageArray } instance server=value
   ```

   For example:

   ```
   OVM> addAdminServer FileServer name=MyNFSServer server=MyServer
   ```

   For more information on the syntax and usage, see Section A.4, “addAdminServer”.

3. You should also attach a refresh server to the file server which is used to refresh the file system. To add a refresh server to the file server, use the syntax:

   ```
   addRefreshServer FileServer instance server=value
   ```

   For example:

   ```
   OVM> addRefreshServer FileServer name=MyNFSServer server=MyServer
   ```

   For more information on the syntax and usage, see Section A.6, “addRefreshServer”.

4. The final step is to refresh the file server so Oracle VM Manager has the most current information about the file server. To refresh the file server, use the syntax:

   ```
   refresh {AccessGroup | Assembly | FileServer | FileSystem | PhysicalDisk | 
   Repository | Server | StorageArray | VirtualAppliance } instance
   ```

   For example:

   ```
   OVM> refresh FileServer name=MyNFSServer
   ```

   For more information on the syntax and usage, see Section A.142, “refresh”.

4.2 Discovering a storage array

1. To discover a storage array use the following syntax:
Discovering a storage array

create StorageArray plugin=value storageType={FIBRECHANNEL | ISCSI | LOCAL | UNKNOWN} [storageName=value] [accessHost=value] [accessPort=value] [accessUsername=value accessPassword=value] [useChap={Yes | No}] [adminHost=value adminUsername=value adminPassword=value] [pluginPrivateData=value] name=value [description=value] [lipScan={Yes | No}]

For example:

```
OVM> create StorageArray plugin="Oracle Generic SCSI Plugin" name=MyISCSIServer 
 storageType=ISCSI accessHost=10.172.76.130 accessPort=3260
```

For more information on the syntax and usage, see Section A.49, “create StorageArray”.

2. After discovering a storage array, you should attach at least one admin server to it so that administrative operations can be performed on the storage array by Oracle VM Manager. To add an admin server to the storage array, use the syntax:

```
addAdminServer {FileServer | StorageArray} instance server=value
```

For example:

```
OVM> addAdminServer StorageArray name=MyISCSIServer server=MyServer
```

For more information on the syntax and usage, see Section A.4, “addAdminServer”.

3. Next you should add the storage initiators to an access group for each Oracle VM Server that is to be granted access to the storage. In this example we add the storage initiators for each Oracle VM Server to the default access group that is created when a storage array is discovered. First, find the name of the access group by listing the access groups for the server using the `show StorageArray` command, for example:

```
OVM> show StorageArray name=MyISCSIServer

...  
Access Group 1 = Default access group @ MyISCSIServer @ 0004fb00000900005264cefc5b9a1cb8 
[Default access group @ MyISCSIServer] Access group name 
Storage Plug-in = oracle.generic.SCSIPlugin.GenericPlugin (1.1.0) 
[Oracle Generic SCSI Plugin] 
...
```

Next, find the storage initiator name for each Oracle VM Server using the `show Server` command, for example:

```
OVM> show Server name=MyServer1

...  
Storage Initiator 1 = iqn.1988-12.com.oracle:e774e056fd3 
[Storage initiator name] 
Storage Initiator 2 = storage.LocalStorageInitiator in 
00:e0:81:4d:40:ff:00:e0:81:4d:40:be:00:e0:81:4d 
...
```

Then add the storage initiator for each Oracle VM Server to the default access group using the syntax:

```
add StorageInitiator instance to AccessGroup instance
```

For example:

```
OVM> add StorageInitiator name=iqn.1988-12.com.oracle:d72d82d0817f to AccessGroup \n name='Default access group @ MyISCSIServer'
```

For more information on the syntax and usage, see Section A.13, “add StorageInitiator”.

24
4. Validate the storage array. Validation should be performed when any changes are made to the storage. To validate the storage array, use the syntax:

```
validate StorageArray instance
```

For example:

```
OVM> validate StorageArray name=MyISCSIServer
```

For more information on the syntax and usage, see Section A.182, “validate”.

5. Refresh the storage array so Oracle VM Manager has the most current information about the storage. To refresh the storage array, use the syntax:

```
refresh { AccessGroup | Assembly | FileServer | FileSystem | PhysicalDisk | Repository | Server | StorageArray | VirtualAppliance } instance
```

For example:

```
OVM> refresh StorageArray name=MyISCSIServer
```

For more information on the syntax and usage, see Section A.142, “refresh”.
Chapter 5 Creating a Network

This section gives you the syntax and examples to create a network with the Virtual Machine role.

5.1 Creating a virtual machine network

1. Create an Ethernet-based network with the Virtual Machine role using the syntax:

   ```
   create Network [roles= {MANAGEMENT | LIVE_MIGRATE | CLUSTER_HEARTBEAT | VIRTUAL_MACHINE | STORAGE } ] name=value [description=value] [on Server instance]
   ```

   For example:

   ```
   OVM> create Network name=MyVMNetwork roles=VIRTUAL_MACHINE
   ```

   For more information on the syntax and usage, see Section A.40, “create Network”.

2. Next, find an Ethernet port from each Oracle VM Server to add to the network. First, find the ID of an Ethernet port using the `show Server` command, for example:

   ```
   OVM> show Server name=MyServer1
   ...
   Ethernet Port 1 = 0004fb000000007711332ff75857ee [eth0 on MyServer3.virtlab.info]
   Ethernet Port 2 = 0004fb0000200000d2e7d2d352a6654e [eth1 on MyServer3.virtlab.info]
   Ethernet Port 3 = 0004fb0000200000c12192a8f2236e4 [eth2 on MyServer3.virtlab.info]
   ...
   OVM>
   ```

   Then, add a port from each Oracle VM Server to the network using the syntax:

   ```
   add Port instance to { BondPort | Network } instance
   ```

   For example:

   ```
   OVM> add Port id=0004fb0000200000d2e7d2d352a6654e to Network name=MyVMNetwork
   ```

   For more information on the syntax and usage, see Section A.10, “add Port”.

---

27
Chapter 6 Creating a Server Pool

This section gives you the syntax and examples to create a server pool and add Oracle VM Server to it.

6.1 Creating a server pool

1. If you are creating a clustered server pool you must provide a file system or physical disk to use for the server pool file system. To find a file system or physical disk, use the list command, for example:

```bash
OVM> list FileSystem
id:66a61958-e61a-44fe-b0e0-9dd64abef7e3 name:nfs on 10.172.76.125:/mnt/vol1/poolfs03
id:0004fb0000050000b85745f78b0c4b61 name:fs on 350014ee2568cc0cf
id:4ebb1575-e611-4662-87b9-a88b40ce3db7 name:nfs on 10.172.76.125:/mnt/vol1/poolfs04
id:858d98c5-3d8b-460e-9160-3415cbdda738 name:
id:0dea4818-20e6-4d3a-958b-b12cf91588b5 name:nfs on 10.172.76.125:/mnt/vol1/poolfs02
id:35b4f1c6-182b-4ea5-9746-51393f3b515c name:nfs on 10.172.76.125:/mnt/vol2/repo03
id:aeb6143d-0a96-4845-9690-740bbf1e225e name:nfs on 10.172.76.125:/mnt/vol1/repo01
id:05e8536f-8d9c-4d7c-bbb2-29b3ffafe011 name:nfs on 10.172.76.125:/mnt/vol2/repo02
id:0004fb00000500000809e28f4fab56b1 name:nfs on 10.172.76.125:/mnt/vol1/repo01
id:0004fb00000500006a46a8dbd2461939 name:MyServerPool_cluster_heartbeat
id:0004fb00000500000809e28f4fab56b1 name:nfs on 10.172.76.125:/mnt/vol1/repo01
```

Before you create a clustered server pool you must refresh the file system or physical disk to be used for the server pool file system. To refresh a file system or physical disk, use the refresh command, for example:

```bash
refresh {AccessGroup|Assembly|FileServer|FileSystem|PhysicalDisk|Repository|Server|StorageArray|VirtualAppliance} instance
```

For example, to refresh a physical disk:

```bash
OVM> refresh PhysicalDisk id=0004fb000018000035ce16ee4d58dc4d
```

And to refresh a file system:

```bash
OVM> refresh FileSystem name="nfs on 10.172.76.125://mnt//vol1//repo01"
```

For more information on the syntax and usage, see Section A.142, “refresh”.

2. To create a server pool use the following syntax:

```bash
create ServerPool clusterEnable={Yes|No} [clusterTimeout=value][fileSystem=value][physicalDisk=value][keymapName={en-us|ar|da|de|de-ch|en-gb|es|et|fi|fo|fr|fr-be|fr-ca|fr-ch|hr|hu|is|it|ja|lt|lv|mk|nl|nl-be|No|pl|pt|pt-br|ru|sl|sv|th|tr}][migrateUsingSsl={Yes|No}][startPolicy={BEST_SERVER|BALANCE_SERVER|CURRENT_SERVER}][policyMode={OFF|DRS|DPM}][policyCpuEnable={Yes|No}][policyCpuThreshold=value][policyCpuPeriod=value][name=value][description=value][virtualIP=value]
```
For example to create a clustered server pool:

```
OVM> create ServerPool clusterEnable=Yes \
    filesystem="nfs on 10.172.76.125://mnt//vol1//poolfs01" name=MyServerPool \
    description='Clustered server pool'
```

And to create an unclustered server pool:

```
OVM> create ServerPool clusterEnable=No name=MyServerPool \
    description='Unclustered server pool'
```

For more information on the syntax and usage, see Section A.45, “create ServerPool”.

### 6.2 Adding Oracle VM Servers to a server pool

To add Oracle VM Servers to a server pool use the following syntax:

```
add Server instance to {AccessGroup | CpuCompatibilityGroup | Repository | ServerPool } instance
```

For example:

```
OVM> add Server name=MyServer to ServerPool name=MyServerPool
```

**Note**

It is not possible to add an Oracle VM Server to a server pool if the Oracle VM Server is running software prior to the 3.4 release and the server pool has not been configured with a virtual IP address. See What is a Master Server and a Virtual IP Address? in the Oracle VM Concepts Guide for more information.

For more information on the syntax and usage, see Section A.11, “add Server”.
Chapter 7 Creating a Storage Repository

This section gives you the syntax and examples to use to create a storage repository.

7.1 Creating a storage repository on a file server

1. Find the file system you want to use to create the storage repository with the list command, for example:

   ```
   OVM> list FileSystem
   id:66a61958-e61a-44fe-b0e0-9dd64abe7e3  name:nfs on 10.172.76.125:/mnt/vol1/poolfs03
   id:9d9b576-e61a-44fe-b0e0-9dd64abe7e3  name:nfs on 10.172.76.125:/mnt/vol1/poolfs04
   id:858d98c5-3d8b-460e-9160-3415cbda73b  name:nfs on 10.172.76.125:/mnt/vol1/poolfs01
   id:0dea4818-20ee-43a9-958b-b12cf91588b5 name:nfs on 10.172.76.125:/mnt/vol1/poolfs02
   id:35b4f1c6-182b-4e5f-9746-5135cbda73b5 name:nfs on 10.172.76.125:/mnt/vol2/repo01
   id:aeb6143d-0a96-4845-9690-740bbf1e225e name:nfs on 10.172.76.125:/mnt/vol1/repo02
   id:35b4f1c6-182b-4e5f-9746-5135cbda73b5 name:nfs on 10.172.76.125:/mnt/vol2/repo02
   id:35b4f1c6-182b-4e5f-9746-5135cbda73b5 name:nfs on 10.172.76.125:/mnt/vol1/repo03
   id:35b4f1c6-182b-4e5f-9746-5135cbda73b5 name:nfs on 10.172.76.125:/mnt/vol1/repo03
   ``

   Then, refresh the file system you intend to use for the storage repository. To refresh the file system, use the syntax:

   ```
   refresh {AccessGroup | Assembly | FileServer | FileSystem | PhysicalDisk | Repository | Server | StorageArray | VirtualAppliance} instance
   ``

   For example:

   ```
   OVM> refresh FileSystem name="nfs on 10.172.76.125://mnt//vol1//repo01"
   ``

   For more information on the syntax and usage, see Section A.142, “refresh”.

2. Create the storage repository. Use the syntax:

   ```
   create Repository[sharePath=value] name=value[description=value] on FileSystem instance
   ``

   For example:

   ```
   OVM> create Repository name=MyRepository on FileSystem \\name="nfs on 10.172.76.125://mnt//vol2//repo01"
   ``

   For more information on the syntax and usage, see Section A.42, “create Repository”.

3. To grant access to the storage repository to a server pool, you must present the repository. To present the storage repository to server pool, use the syntax:

   ```
   add ServerPool instance to {AccessGroup | Repository} instance
   ``

   For example:

   ```
   OVM> add ServerPool name=MyServerPool to Repository name=MyRepository
   ``

   For more information on the syntax and usage, see Section A.12, “add ServerPool”.

4. Finally, refresh the storage repository using the syntax:
To create a storage repository on a storage array

```
refresh {AccessGroup|Assembly|FileServer|FileSystem|PhysicalDisk|Repository|Server|StorageArray|VirtualAppliance} instance
```

For example:

```
OVM> refresh Repository name=MyRepository
```

For more information on the syntax and usage, see Section A.142, “refresh”.

### 7.2 To create a storage repository on a storage array

1. Find the physical disk (LUN) you want to use to create the storage repository with the `list` command, for example:

```
OVM> list PhysicalDisk
```

```
  id:0004fb000018000067ce80973e18374e  name:MyLUN1
  id:0004fb000018000035ce16ee4d58dc4d  name:MyLUN2
  id:0004fb0000180000a9c7a87ba52ce5ec  name:MyLUN3
```

```
OVM>
```

2. Find a local file system on an Oracle VM Server that has access to the LUN with the `list` command, for example:

```
OVM> list FileServer
```

```
  id:0004fb0000090000d773cb3fe655865a  name:Local FS MyServer1
  id:0004fb000009000014baa666cdf62317  name:Local FS MyServer2
  id:0004fb00000900008ae3eb6203f5646c  name:MyNFSStorage
  id:0004fb00000900001d523e2a6ce1f8c8  name:Local FS MyServer3
```

```
OVM>
```

3. Create an OCFS2 file system on the LUN using the local file system on the Oracle VM Server as the FileServer. Use the syntax:

```
create FileSystem physicalDisk=value name=value [description=value] on FileServer instance
```

For example:

```
OVM> create FileSystem name=MyRepoFileSystem physicalDisk="MyLUN1" on FileServer \ name="Local FS MyServer1"
```

For more information on the syntax and usage, see Section A.39, “create FileSystem”.

4. Create the storage repository. Use the syntax:

```
create Repository [sharePath=value] name=value [description=value] on FileSystem instance
```

For example:

```
OVM> create Repository name=MyRepository on FileSystem name=MyRepoFileSystem
```

For more information on the syntax and usage, see Section A.42, “create Repository”.

5. To grant access to the storage repository to a server pool, you must `present` the repository. To present the storage repository to server pool, use the syntax:

```
add ServerPool instance to {AccessGroup | Repository} instance
```

For example:
Importing resources to a storage repository

OVM> add ServerPool name=MyServerPool to Repository name=MyRepository

For more information on the syntax and usage, see Section A.12, “add ServerPool”.

6. Finally, refresh the storage repository using the syntax:

```
refresh { AccessGroup | Assembly | FileServer | FileSystem | PhysicalDisk | Repository | Server | StorageArray | VirtualAppliance } instance
```

For example:

```
OVM> refresh Repository name=MyRepository
```

For more information on the syntax and usage, see Section A.142, “refresh”.

7.3 Importing resources to a storage repository

7.3.1 Importing a virtual appliance

To import a virtual appliance to the storage repository, use the following syntax:

```
importVirtualAppliance Repository instance url=value[ proxy=value ]
```

For example:

```
OVM> importVirtualAppliance Repository name=MyRepository url="http://example.com//myvirtualappliance.ova"
```

For more information on the syntax and usage, see Section A.133, “importVirtualAppliance”.

7.3.2 Importing a virtual machine template

To import a virtual machine template to the storage repository, use the following syntax:

```
importTemplate Repository instance url=value[ proxy=value ]
```

For example:

```
OVM> importTemplate Repository name=MyRepository url="http://example.com//mytemplate.tgz"
```

For more information on the syntax and usage, see Section A.132, “importTemplate”.

7.3.3 Importing an ISO file

To import an ISO file to the storage repository, use the following syntax:

```
importVirtualCdrom Repository instance url=value[ proxy=value ]
```

For example:

```
OVM> importVirtualCdrom Repository name=MyRepository url="http://example.com//myiso.iso"
```

For more information on the syntax and usage, see Section A.134, “importVirtualCdrom”.


Chapter 8 Creating a Virtual Machine

This section gives you the syntax and examples to use to create a virtual machine from a number of sources.

8.1 Creating a virtual machine from a virtual appliance

1. When you import a virtual appliance, each virtual machine in the virtual appliance file is unpacked and stored as a VirtualApplianceVm object. The VirtualApplianceVm object is then used to create a virtual machine, not the virtual appliance itself. Use the `show VirtualAppliance` command to find the name or ID of the new VirtualApplianceVm objects in a virtual appliance.

   ```
   OVM> show VirtualAppliance name=myappliance.ova
   Origin = http://example.com/myappliance.ova
   Repository =
   ID | Name
   Virtual Appliance Vm 1 = ID | Name
   Assembly VirtualDisk 1 = ID | Name
   Id = ID | Name
   Name = myappliance.ova
   Description = Import URL: http://example.com/myappliance.ova
   Locked = false
   ```

2. Create a virtual machine from a VirtualApplianceVm object using the syntax:

   ```
   createVmFromVirtualApplianceVm VirtualApplianceVm instance
   ```

   Make sure you use the name of the VirtualApplianceVm object created when you imported the virtual appliance, not the name of the VirtualAppliance object.

   For example:

   ```
   OVM> createVmFromVirtualApplianceVm VirtualApplianceVm name=MyVm
   ```

   For more information on the syntax and usage, see Section A.33, “createVmFromVirtualApplianceVm”.

3. The previous step creates a virtual machine with the name `virtualappliance_name_vm_name`, for example:

   `myappliance.ova_myvm`

8.2 Creating a virtual machine from a template

Clone a virtual machine from a template, using the syntax:

```
clone Vm instance destType= {Vm | VmTemplate} [destName=value] serverPool=value [cloneCustomizer=value] [targetRepository=value]
``` 

For example:

```
OVM> clone Vm name=MyVMTemplate.tgz destType=Vm destName=MyNewVM serverPool=MyServerPool
``` 

For more information on the syntax and usage, see Section A.24, “clone Vm”.

8.3 Creating a virtual machine from an ISO

1. Create a virtual machine using the syntax:
Creating a virtual machine from an ISO

create Vm[memory=value][memoryLimit=value][cpuCount=value][
cpuCountLimit=value][cpuPriority=value][cpuUtilizationCap=value][
highAvailability= {Yes | No}][hugePagesEnabled= {Yes | No}][osType=value]
[restartActionOnCrash= {RESTART | STOP | RESTART_AFTER_DUMP | STOP_AFTER_DUMP}
][mouseType= {OS_DEFAULT | PS2_MOUSE | USB_MOUSE | USB_TABLET}]
domainType= {XEN_HVM | XEN_HVM_PV_DRIVERS | XEN_PVM | LDOMS_PVM | UNKNOWN}
[keymapName= {en-us | ar | da | de | de-ch | en-gb | es | et | fi | fo | fr | fr-be | fr-ca | fr-ch | hr | hu | is |
   it | ja | lt | lv | mk | nl | nl-be | No | pl | pt | pt-br | ru | sl | sv | th | tr}][bootOrder= {PXE | DISK | CDROM}][networkInstallPath=value][repository=value][server=value]
[startPolicy= {BEST_SERVER | BALANCE_SERVER | CURRENT_SERVER | USE_POOL_POLICY}][
   viridian= {Yes | No}][name=value][description=value] on ServerPool instance

For example:

OVM> create Vm name=MyVM repository=MyRepository domainType=XEN_HVM \ 
   server=MyServer startPolicy=USE_POOL_POLICY on ServerPool name=MyServerPool

For more information on the syntax and usage, see Section A.53, “create Vm”.

2. Create a virtual disk to use as the boot disk using the syntax:

create VirtualDisk size=value shareable= {Yes | No} sparse= {Yes | No} 
   name=value[description=value] on Repository instance

For example:

OVM> create VirtualDisk name=MyVMDisk size=10 sparse=Yes shareable=No on Repository \ 
   name=MyRepository

For more information on the syntax and usage, see Section A.51, “create VirtualDisk”.

3. Map the virtual disk to the virtual machine using the syntax:

create VmDiskMapping slot=value{physicalDisk=value|virtualDisk=value| 
   virtualCd={value|EMPTY_CDROM}} name=value[description=value] on Vm instance

For example:

OVM> create VmDiskMapping slot=0 virtualDisk=MyVMDisk name="Boot Disk" on Vm name=MyVM

For more information on the syntax and usage, see Section A.57, “create VmDiskMapping”.

4. Map an ISO file to the virtual machine using the syntax:

create VmDiskMapping slot=value{physicalDisk=value|virtualDisk=value| 
   virtualCd={value|EMPTY_CDROM}} name=value[description=value] on Vm instance

For example:

OVM> create VmDiskMapping slot=1 virtualCd=OracleLinux-dvd.iso \ 
   name="CDROM Drive" on Vm name=MyVM

For more information on the syntax and usage, see Section A.57, “create VmDiskMapping”.

5. Set up the disk boot order as the CDROM (ISO file) as the first disk, then the virtual disk as the secondary disk, using the syntax:

create Vm virtualCd=OracleLinux-dvd.iso, MyVMDisk size=10 sparse=Yes shareable=No on Repository \ 
   name=MyServerPool instance

startPolicy=USE_POOL_POLICY
Starting a virtual machine

edit Vm instance [memory=value] [memoryLimit=value] [cpuCount=value] [cpuCountLimit=value] [cpuPriority=value] [cpuUtilizationCap=value] [highAvailability={Yes|No}] [hugePagesEnabled={Yes|No}] [osType=value] [restartActionOnCrash={RESTART|STOP|RESTART_AFTER_DUMP|STOP_AFTER_DUMP}] [mouseType={OS_DEFAULT|PS2_MOUSE|USB_MOUSE|USB_TABLET}] [domainType={XEN_HVM|XEN_HVM_PV_DRIVERS|XEN_PVM|LDOMS_PVM|UNKNOWN}] [keymapName={en-us|ar|da|de|de-ch|en-gb|es|et|fi|fo|fr|fr-be|fr-ca|fr-ch|hr|hu|is|it|ja|lt|lv|mk|nl|nl-be|No|pl|pt|pt-br|ru|sl|sv|th|tr}] [bootOrder={PXE|DISK|CDROM}] [networkInstallpath=value] [startPolicy={BEST_SERVER|BALANCE_SERVER|CURRENT_SERVER|USE_POOL_POLICY}] [viridian={Yes|No}] [name=value] [description=value]

For example:

```
OVM> edit Vm name=MyVM bootOrder='CDROM,DISK' startPolicy=BEST_SERVER
```

For more information on the syntax and usage, see Section A.95, "edit Vm".

6. Create a VNIC and add it to the virtual machine using the syntax:

```
create Vnic network=value name=value [macAddress=value] [description=value] on Vm instance
```

For example:

```
OVM> create Vnic name=MyVNIC network=MyVMNetwork on Vm name=MyVM
```

For more information on the syntax and usage, see Section A.58, "create Vnic".

**8.4 Starting a virtual machine**

Start the virtual machine, using the syntax:

```
start {Server | Vm} instance
```

For example:

```
OVM> start Vm name=MyVM
```

For more information on the syntax and usage, see Section A.177, "start".
Part II CLI Command Reference

This part gives the full syntax of each CLI command with examples.

In some commands such as any command that edits an object, you may see slight differences between the syntax in the CLI syntax help, and that documented here. This is because the CLI syntax help uses an asterisk to mark options that are mandatory for an object and maps directly to mandatory options when using the Oracle VM Manager user interface, but not necessarily mandatory when entering a command in the CLI. The syntax documented in this section instead shows you what is optional or mandatory when using that command in the CLI.
Appendix A CLI Command Reference

This appendix gives the full syntax of each CLI command, with usage examples.

A.1 abort Job

Aborts a job.

Syntax

    abort Job instance

Where instance is:

    {id=value|name=value}

Description

This command aborts a running job.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.1 Aborting a job

    OVM> abort job id=1342399962239

See Also

- Section A.110, “getDebugTranscript”
- Section A.115, “getJobs”
- Section A.138, “list”
- Section A.171, “show”

A.2 ackEvent

Acknowledges an event.

Syntax

    ackEvent eventId=value
Description

This command acknowledges an event.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventId=value</td>
<td>The ID of the event. To get the ID of an event, use the getEvents or getEventsForObject command.</td>
</tr>
</tbody>
</table>

Examples

Example A.2 Acknowledging an event

```
OVM> ackEvent eventId=1342155856562
```

See Also

- Section A.113, “getEvents”
- Section A.114, “getEventsForObject”

A.3 addAccessHost

Adds an access host to an iSCSI server.

Syntax

```
addAccessHost StorageArray instance accessHost=value[accessPort=value][accessUsername=value accessPassword=value]
```

Where `instance` is:

```
{id=value|name=value}
```

Description

This command adds an access host to an iSCSI storage array. Adding more than one access host provides multiple network paths to the storage. Create an access host for each path when using multipathing. At least one access host must be set. This is not applicable to fibre channel storage. To remove an access host, use the removeAccessHost command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accessHost=value</td>
<td>The hostname or IP address for the access host.</td>
</tr>
<tr>
<td>accessPort=value</td>
<td>The port on which to connect to the access host. The default port of 3260 is used if no value is provided.</td>
</tr>
<tr>
<td>accessUsername=value</td>
<td>The username to use when using CHAP authentication.</td>
</tr>
</tbody>
</table>
### Examples

**Example A.3 Adding a storage array access host**

```
OVM> addAccessHost StorageArray name=MyISCSIServer accessHost=10.172.76.131
```

### See Also

- Section A.146, "removeAccessHost"
- Section A.49, “create StorageArray”
- Section A.86, “edit StorageArray”
- Section A.138, “list”
- Section A.171, “show”

### A.4 addAdminServer

Adds an administrative Oracle VM Server to a file server or storage array.

**Syntax**

```
addAdminServer {FileServer | StorageArray } instance server=value
```

Where `instance` is:

```
{id=value | name=value}
```

**Description**

This command adds an administrative Oracle VM Server to a file server or storage array.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server=value</td>
<td>The name or ID of the administrative Oracle VM Server.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

### Examples

**Example A.4 Adding an admin server to a file server**

```
OVM> addAdminServer FileServer name=MyNFSserver server=MyServer
```
Example A.5 Adding an admin server to a storage array

OVMM> addAdminServer StorageArray name=MyISCSIServer server=MyServer

See Also

- Section A.147, “removeAdminServer”
- Section A.38, “create FileServer”
- Section A.49, “create StorageArray”
- Section A.70, “edit FileServer”
- Section A.86, “edit StorageArray”
- Section A.154, “remove Server”
- Section A.11, “add Server”
- Section A.6, “addRefreshServer”
- Section A.149, “removeRefreshServer”
- Section A.86, “edit StorageArray”

A.5 addPolicyServer

Adds a server pool policy to a server.

Syntax

addPolicyServer ServerPool instance server=value

Where instance is:

{ id=value | name=value }

Description

This command adds a server pool policy to a server.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
<tr>
<td>server=value</td>
<td>The name or ID of an Oracle VM Server to which the policy is added.</td>
</tr>
</tbody>
</table>

Examples

Example A.6 Adding a policy to a server

OVMM> addPolicyServer ServerPool name=MyServerPool server=MyServer
See Also

• Section A.148, “removePolicyServer”
• Section A.46, “create ServerPoolNetworkPolicy”
• Section A.83, “edit ServerPoolNetworkPolicy”

A.6 addRefreshServer

Adds a refresh server to a file server.

Syntax

addRefreshServer FileServer instance server=value

Where instance is:

{ id=value | name=value }

Description

This command adds a refresh server to a file server. The refresh server is an Oracle VM Server that is used to refresh the file systems on an NFS file server. You can add multiple refresh servers to a file server. A file server must have at least one refresh server assigned to it.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server=value</td>
<td>The name or ID of the Oracle VM Server to be used as a refresh server.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

Example A.7 Adding a refresh server to file server

OVM> addRefreshServer FileServer name=MyNFSServer server=MyServer

See Also

• Section A.149, “removeRefreshServer”
• Section A.38, “create FileServer”
• Section A.70, “edit FileServer”
• Section A.154, “remove Server”
• Section A.11, “add Server”
A.7 add BondPort

Adds an bonded port to a network object.

Syntax

```
add BondPort instance to Network instance
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command adds a bonded port to a network object.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Examples

Example A.8 Adding a bonded port to a network

```
OVM> add BondPort id=0004fb000020000065822cb7bb9ec296 to Network name=MyVMNetwork
```

See Also

- Section A.36, “create BondPort”
- Section A.66, “edit BondPort”
- Section A.150, “remove BondPort”
- Section A.153, “remove Port”
- Section A.102, “embeddedCreate”
- Section A.103, “embeddedDelete”
- Section A.104, “embeddedEdit”
- Section A.59, “delete”
- Section A.138, “list”
A.8 add FileSystem

Adds a file system to an access group.

Syntax

```
add FileSystem instance to AccessGroup instance
```

Where `instance` is:

```
{ id=value | name=value }
```

Note that if the instance name contains forward slashes, these must be escaped using an additional forward slash. This is illustrated in the examples for this command.

Description

This command adds a file system to an access group.

A file system may only be associated with one access group. If you create a new access group for a file system that is already associated with an existing access group, the file system is disassociated from the original access group.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Examples

**Example A.9 Adding a file system to an access group**

```
OVM> add FileSystem name="nfs on 10.172.76.125://mnt//vol2//repo03" to AccessGroup \
       name=MyAccessGroup
```

See Also

- Section A.39, “create FileSystem”
- Section A.72, “edit FileSystem”
- Section A.151, “remove FileSystem”
- Section A.34, “create AccessGroup”
- Section A.61, “edit AccessGroup”
- Section A.138, “list”
- Section A.171, “show”
A.9 add PhysicalDisk

Adds a physical disk to a SAN storage access group.

Syntax

```
add PhysicalDisk instance to AccessGroup instance
```

Where `instance` is:

```
{id=value|name=value}
```

Description

This command adds a physical disk to a SAN storage access group. `Local storage` and generic storage plug-ins are not supported with this command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

Examples

Example A.10 Adding a physical disk to a SAN storage access group

```
OVM> add PhysicalDisk id=0004fb00001800007ee6dbda7b4461cb to AccessGroup  "name='Default access group @ MyISCSIServer'
```

See Also

- Section A.41, “create PhysicalDisk”
- Section A.76, “edit PhysicalDisk”
- Section A.152, “remove PhysicalDisk”
- Section A.59, “delete”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

A.10 add Port
Syntax

Adds an Ethernet port to a network object.

**Syntax**

```plaintext
add Port instance to {BondPort | Network} instance
```

Where `instance` is:

```plaintext
{ id=value | name=value }
```

**Description**

This command adds an Ethernet port to a network object.

To configure the IP address for a port, use the `embeddedCreate` command.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{BondPort</td>
<td>Network}`</td>
</tr>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.11 Adding an Ethernet port to a network**

```plaintext
OVMD> add Port id=0004fb0000200000d2e7d2d352a6654e to Network name=MyVMNetwork
```

**See Also**

- Section A.77, “edit Port”
- Section A.153, “remove Port”
- Section A.40, “create Network”
- Section A.74, “edit Network”
- Section A.36, “create BondPort”
- Section A.102, “embeddedCreate”
- Section A.103, “embeddedDelete”
- Section A.104, “embeddedEdit”
- Section A.138, “list”
- Section A.171, “show”

A.11 add Server
Adds an Oracle VM Server to an object.

Syntax

```
add Server instance to { AccessGroup | CpuCompatibilityGroup | Repository | ServerPool } instance
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command adds an Oracle VM Server to either a CPU compatibility group, server pool, storage repository or access group.

When you add an Oracle VM Server to a storage repository, you are making that Oracle VM Server available to perform admin duties for that storage object.

To present a storage repository to all Oracle VM Servers in a server pool, use the `add ServerPool` command.

To add admin servers to a file server or storage array, use the `addAdminServer` command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{ AccessGroup</td>
<td>CpuCompatibilityGroup</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Examples

**Example A.12 Adding an Oracle VM Server to a CPU compatibility group**

```
OVM> add Server name=MyServer to CpuCompatibilityGroup name=MyCPUGroup
```

**Example A.13 Adding an Oracle VM Server to a server pool**

```
OVM> add Server name=MyServer to ServerPool name=MyServerPool
```

**Note**

It is not possible to add an Oracle VM Server to a server pool if the Oracle VM Server is running software prior to the 3.4 release and the server pool has not been configured with a virtual IP address. See What is a Master Server and a Virtual IP Address? in the Oracle VM Concepts Guide for more information.

**Example A.14 Adding an Oracle VM Server to an access group**

```
OVM> add Server name=MyServer to AccessGroup name=MyAccessGroup
```
Example A.15 Adding (presenting) an Oracle VM Server to a storage repository

```
OVM> add Server name=MyServer to Repository name=MyRepository
```

See Also

- Section A.12, “add ServerPool”
- Section A.60, “discoverServer”
- Section A.80, “edit Server”
- Section A.142, “refresh”
- Section A.177, “start”
- Section A.178, “stop”
- Section A.162, “restart”
- Section A.137, “kill”
- Section A.181, “upgrade”
- Section A.4, “addAdminServer”
- Section A.147, “removeAdminServer”
- Section A.6, “addRefreshServer”
- Section A.149, “removeRefreshServer”
- Section A.154, “remove Server”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

A.12 add ServerPool

Adds a server pool to a storage repository or to an access group.

Syntax

```
add ServerPool instance to { AccessGroup | Repository } instance
```

Where `instance` is:

{ `id=value` | `name=value` }

Description

This command presents a storage repository to all Oracle VM Servers in a server pool. To present a storage repository to an individual Oracle VM Server, use the `add Server` command.
This command also adds a server pool to an access group.

**Important**

The option to add or present an entire server pool is a convenience that automatically selects all of the servers that belong to the specified pool and then performs the action on those servers. There is no actual relationship between the server pool and the repository or access group stored within Oracle VM Manager. This means that if you add a server to a server pool after having presented a repository to the server pool, the repository is not automatically presented to the new server. Equally, removing a server from the server pool does not automatically update the configuration.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{AccessGroup</td>
<td>Repository}`</td>
</tr>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

**Examples**

Example A.16 Presenting a storage repository to a server pool

```
OVM> add ServerPool name=MyServerPool to Repository name=MyRepository
```

Example A.17 Adding a server pool to an access group

```
OVM> add ServerPool name=MyServerPool to AccessGroup name=MyAccessGroup
```

**See Also**

- Section A.155, “remove ServerPool”
- Section A.11, “add Server”
- Section A.138, “list”
- Section A.171, “show”

**A.13 add StoragelInitiator**

Adds a storage initiator to an access group for a SAN storage server.

**Syntax**

```
add StoragelInitiator instance to AccessGroup instance
```

Where `instance` is:

```
{ id=value | name=value }
```
Description

This command adds a storage initiator to an access group for a SAN storage server.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.18 Adding a storage initiator

```bash
OVM> add StorageInitiator name=iqn.1988-12.com.oracle:d72d82d0817f to AccessGroup \
    name='Default access group @ MyISCSIServer'
```

See Also

- Section A.156, “remove StorageInitiator”
- Section A.138, “list”
- Section A.171, “show”

A.14 add Tag

Adds a tag to an object.

Syntax

```
add Tag instance to {Server | ServerPool | Vm} instance
```

Where instance is:

{ id=value | name=value }

Description

This command adds a tag used to identify and group objects to an object.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Server</td>
<td>ServerPool</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>
Examples

Example A.19 Adding a tag to a server pool

```
OVM> add Tag name=MyTag to ServerPool name=MyServerPool
```

See Also

- Section A.50, “create Tag”
- Section A.88, “edit Tag”
- Section A.157, “remove Tag”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

A.15 add VlanInterface

Adds a VLAN interface to a network.

Syntax

```
add VlanInterface instance to Network instance
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command adds a VLAN interface to a network. To create a VLAN interface, use the `create VlanInterface` command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

Examples

Example A.20 Adding a VLAN interface to a network

```
OVM> add VlanInterface name=MyVLANInterface to Network name=MyNetwork
```

See Also

- Section A.52, “create VlanInterface”
A.16 add Vm

Add a virtual machine to an Oracle VM Server, server pool, or anti affinity group.

Syntax

```
add Vm instance to { AntiAffinityGroup | Server | ServerPool } instance
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command adds a virtual machine to an Oracle VM Server, server pool, or anti affinity group. The virtual machine cannot be running, and must be stopped before using this command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ AntiAffinityGroup</td>
<td>Server</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

**Example A.21 Adding a virtual machine to a server pool**

```
OVM> add Vm name=MyVM to ServerPool name=MyServerPool
```

**Example A.22 Adding a virtual machine to an Oracle VM Server**

```
OVM> add Vm name=MyVM to Server name=MyServer
```

**Example A.23 Adding a virtual machine to an anti affinity group**

```
OVM> add Vm name=MyVM to AntiAffinityGroup name=MyAAGroup
```
See Also

• Section A.53, “create Vm”
• Section A.136, “importVirtualMachine”
• Section A.35, “create AntiAffinityGroup”
• Section A.159, “remove Vm”
• Section A.139, “migrate Vm”
• Section A.140, “migrateWithLocalStorage Vm”
• Section A.24, “clone Vm”
• Section A.177, “start”
• Section A.171, “show”
• Section A.138, “list”

A.17 add Vnic

Add a VNIC to a network.

Syntax

```
add Vnic instance to Network instance
```

Where `instance` is:

{ `id=value` | `name=value` }

Description

This command adds a VNIC to a network.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ <code>id=value</code></td>
<td><code>name=value</code> }</td>
</tr>
</tbody>
</table>

Examples

Example A.24 Adding a VNIC to a network

```
OVMD> add Vnic name=00:21:f6:00:00:00 to Network name=MyNetwork
```

See Also

• Section A.58, “create Vnic”
A.18 changeServerAgentPassword

Changes the Oracle VM Agent password on an Oracle VM Server.

Syntax

changeServerAgentPassword Server instance oldPassword=value newPassword=value confirmPassword=value

Where instance is:

{ id=value | name=value }

Description

This command changes the Oracle VM Agent password on an Oracle VM Server.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oldPassword=value</td>
<td>The existing password for the Oracle VM Agent on the Oracle VM Server. The password is displayed as asterisks.</td>
</tr>
<tr>
<td>newPassword=value</td>
<td>The new password for the Oracle VM Agent on the Oracle VM Server. The password is displayed as asterisks.</td>
</tr>
<tr>
<td>confirmPassword=value</td>
<td>The new password for the Oracle VM Agent on the Oracle VM Server. The password is displayed as asterisks.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

Example A.25 Changing the Oracle VM Agent password on an Oracle VM Server

```
OVM> changeServerAgentPassword Server name=MyServer oldPassword=***** \
    newPassword=******* confirmPassword=******
```

See Also

• Section A.138, “list”
• Section A.171, “show”
A.19 checkUpToDate

Checks whether the Oracle VM Server software is up-to-date according to the server update repository.

Syntax

```
checkUpToDate Server instance
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command checks whether the Oracle VM Server software is up-to-date according to the server update repository. This command sets the `Up To Date` parameter of the Server object, and does not display any output other than a success or failure message. This command may be useful to check whether an Oracle VM Server is up-to-date in between any regular checking by the recurring job that checks for available updates. To see the value of the Server object's `Up To Date` parameter, use the `show Server` command. To update an Oracle VM Server, use the `upgrade` command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Examples

**Example A.26 Checking whether an Oracle VM Server is up-to-date**

```
OVM> checkUpToDate Server name=MyServer
```

See Also

- Section A.48, “create ServerUpdateRepository”
- Section A.85, “edit ServerUpdateRepository”
- Section A.47, “create ServerUpdateGroup”
- Section A.84, “edit ServerUpdateGroup”
- Section A.19, “checkUpToDate”
- Section A.181, “upgrade”
- Section A.171, “show”

A.20 clearVmAllRcvdMessages
Clears all the key/value pair messages received by a running virtual machine.

**Syntax**

```plaintext
clearVmAllRcvdMessages Vm instance
```

Where `instance` is:

```plaintext
{ id=value | name=value }
```

**Description**

This command clears all the key/value pair messages received by a running virtual machine.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`id=value</td>
<td>name=value`</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.27 Clearing all messages received by a virtual machine**

```plaintext
OVM> clearVmAllRcvdMessages Vm name=MyVm
```

**See Also**

- Section A.164, “sendVmMessage”
- Section A.128, “getVmSentMessages”
- Section A.127, “getVmReceivedMessages”
- Section A.22, “clearVmRcvdMessage”
- Section A.23, “clearVmSentMessage”
- Section A.21, “clearVmAllSentMessages”

### A.21 clearVmAllSentMessages

Clears all the key/value pair messages sent to a running virtual machine.

**Syntax**

```plaintext
clearVmAllSentMessages Vm instance
```

Where `instance` is:

```plaintext
{ id=value | name=value }
```
Description

This command clears all the key/value pair messages sent to a running virtual machine.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.28 Clearing all messages sent to a virtual machine

```
OVM> clearVmAllSentMessages Vm name=MyVm
```

See Also

- Section A.164, “sendVmMessage”
- Section A.128, “getVmSentMessages”
- Section A.127, “getVmReceivedMessages”
- Section A.22, “clearVmRcvdMessage”
- Section A.23, “clearVmSentMessage”
- Section A.20, “clearVmAllRcvdMessages”

A.22 clearVmRcvdMessage

Clears a key/value pair message received by a running virtual machine.

Syntax

```
clearVmRcvdMessage Vm instance key=value
```

Where instance is:

{ id=value | name=value }

Description

This command clears a key/value pair message received by a running virtual machine.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key=value</td>
<td>The name or ID of the message key.</td>
</tr>
</tbody>
</table>
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

### Examples

**Example A.29 Clearing a message received by a virtual machine**

```bash
OVM> clearVmRcvdMessage Vm name=MyVm key="com.oracle.linux.network.device.0"
```

### See Also

- Section A.164, “sendVmMessage”
- Section A.128, “getVmSentMessages”
- Section A.127, “getVmReceivedMessages”
- Section A.23, “clearVmSentMessage”
- Section A.21, “clearVmAllSentMessages”
- Section A.20, “clearVmAllRcvdMessages”

### A.23 clearVmSentMessage

Clears a key/value pair message sent to a running virtual machine.

**Syntax**

```bash
clearVmSentMessage Vm instance key=value
```

Where `instance` is:

```bash
{ id=value | name=value }
```

**Description**

This command clears a key/value pair message sent to a running virtual machine.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key=value</td>
<td>The name or ID of the message key.</td>
</tr>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

### Examples

**Example A.30 Clearing a message sent to a virtual machine**

```bash
OVM> clearVmSentMessage Vm name=MyVm key="com.oracle.linux.network.device.0"
```
A.24 clone Vm

Clones a virtual machine or template to a new virtual machine or template.

Syntax

```
clone Vm instance destType= \{ Vm | VmTemplate \}[destName=value] serverPool=value[cloneCustomizer=value][targetRepository=value]
```

Where `instance` is:

```
\{ id=value | name=value \}
```

Description

This command clones a virtual machine or template to a new virtual machine or template.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`destType= { Vm</td>
<td>VmTemplate }`</td>
</tr>
<tr>
<td><code>destName=value</code></td>
<td>The name of the cloned virtual machine or template.</td>
</tr>
<tr>
<td><code>serverPool=value</code></td>
<td>The name or ID of the server pool on which to deploy the cloned virtual machine.</td>
</tr>
<tr>
<td><code>cloneCustomizer=value</code></td>
<td>The name or ID of the cloneCustomizer that should be used when deploying the cloned virtual machine or template.</td>
</tr>
<tr>
<td><code>targetRepository=value</code></td>
<td>The name or ID of the repository that should be used when deploying the cloned virtual machine or template.</td>
</tr>
</tbody>
</table>

Note

Although you must enter this when cloning a virtual machine to a template, the template is not deployed to a server pool, it is located in the storage repository.
Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.31 Cloning a virtual machine to a virtual machine

```
OVM> clone Vm name=MyVM destType=Vm destName=MyNewVM serverPool=MyServerPool
```

Example A.32 Cloning a virtual machine to a template

```
OVM> clone Vm name=MyVM destType=VmTemplate destName=MyVMTemplate serverPool=MyServerPool
```

Example A.33 Cloning a template to a virtual machine

```
OVM> clone Vm name=MyVMTemplate.tgz destType=Vm destName=MyNewVM serverPool=MyServerPool
```

Example A.34 Cloning a template to a template

```
OVM> clone Vm name=MyVMTemplate.tgz destType=VmTemplate destName=MyVMTemplate serverPool=MyServerPool
```

Example A.35 Cloning a virtual machine to a virtual machine using a clone customizer

```
OVM> clone Vm name=MyVM destType=Vm destName=MyNewVM serverPool=MyServerPool cloneCustomizer=MyCloneCustomizer targetRepository=MyRepository
```

See Also

- Section A.53, “create Vm”
- Section A.16, “add Vm”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

A.25 cloneCdToRepo

Clones a virtual CDROM to a storage repository.

Syntax

```
cloneCdToRepo VirtualCdrom instance target=value cloneType= { SPARSE_COPY | NON_SPARSE_COPY | THIN_CLONE }
```

Where instance is:

```
{ id=value | name=value }
```

Description

This command clones a virtual CDROM to a target repository.
Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target=value</td>
<td>The repository on which to locate the cloned physical disk.</td>
</tr>
<tr>
<td>cloneType=</td>
<td>Whether to clone a sparse, non-sparse or thin-clone virtual cdrom.</td>
</tr>
<tr>
<td>{ SPARSE_COPY</td>
<td>NON_SPARSE_COPY</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

Example A.36 Cloning a virtual CDROM to a repository

```
OVMM> cloneCdToRepo VirtualCdrom name=MyCD.iso target=MyRepository cloneType=SPARSE_COPY
```

See Also

- Section A.134, “importVirtualCdrom”
- Section A.92, “edit VirtualCdrom”
- Section A.26, “clonePdToPd”
- Section A.28, “clonePdToStorageArray”
- Section A.41, “create PhysicalDisk”
- Section A.76, “edit PhysicalDisk”
- Section A.59, “delete”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

A.26 clonePdToPd

Clones a physical disk to a physical disk.

Syntax

```
clonePdToPd PhysicalDisk instance target=value cloneType= { SPARSE_COPY | NON_SPARSE_COPY | THIN_CLONE }
```
Description

Where instance is:

\{ id=value | name=value \}

Description

This command clones a physical disk to a target physical disk. You cannot clone a physical disk using this command if the disk contains a file system or storage repository.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target=value</td>
<td>The physical disk on which to locate the cloned physical disk.</td>
</tr>
<tr>
<td>cloneType= { SPARSE_COPY</td>
<td>NON_SPARSE_COPY</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

Example A.37 Cloning a physical disk to a physical disk

OVM> clonePdToPd PhysicalDisk name=MyDisk1 target=MyRepository cloneType=SPARSE_COPY

See Also

- Section A.29, "cloneVdToPd"
- Section A.28, "clonePdToStorageArray"
- Section A.41, "create PhysicalDisk"
- Section A.76, "edit PhysicalDisk"
- Section A.59, "delete"
- Section A.142, "refresh"
- Section A.138, "list"
- Section A.171, "show"
- Section A.113, "getEvents"

A.27 clonePdToRepo

Clones a physical disk to a repository.
**Syntax**

```
clonPdToRepo PhysicalDisk instance target=value cloneType= {SPARSE_COPY | NON_SPARSE_COPY | THIN_CLONE}
```

Where `instance` is:

```
{id=value | name=value}
```

**Description**

This command clones a physical disk to a target repository. The physical disk data is copied to a virtual disk image (.img file) file in the storage repository.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target=value</td>
<td>The repository on which to locate the cloned physical disk. The target repository must be on a physical disk.</td>
</tr>
<tr>
<td>cloneType= {SPARSE_COPY</td>
<td>NON_SPARSE_COPY</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.38 Cloning a physical disk to a repository**

```
OVM> clonePdToRepo PhysicalDisk name=MyDisk1 target=MyRepository cloneType=SPARSE_COPY
```

**See Also**

- Section A.26, “clonePdToPd”
- Section A.28, “clonePdToStorageArray”
- Section A.41, “create PhysicalDisk”
- Section A.76, “edit PhysicalDisk”
- Section A.59, “delete”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”
A.28 clonePdToStorageArray

Clones a physical disk to a storage array.

Syntax

```
clonePdToStorageArray 
| PhysicalDisk instance target=value cloneType= { SPARSE_COPY 
| NON_SPARSE_COPY | THIN_CLONE } userFriendlyName=value
```

Where instance is:

```
{ id=value | name=value }
```

Description

This command clones a physical disk to a target storage array.

This command is not supported with a generic ISCSI Oracle VM Storage Connect plug-in. The clone target must be on the same storage array as the source. You cannot clone a disk from one storage array to another.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target=value</td>
<td>The storage array on which to locate the cloned physical disk.</td>
</tr>
<tr>
<td>cloneType= { SPARSE_COPY</td>
<td>NON_SPARSE_COPY</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
<tr>
<td>userFriendlyName=value</td>
<td>An optional parameter to specify a user-friendly name that can be used to identify the physical disk object.</td>
</tr>
</tbody>
</table>

Examples

**Example A.39 Cloning a physical disk to a storage array**

```
OVM> clonePdToStorageArray PhysicalDisk name=MyDisk1 target=MyRepository cloneType=SPARSE_COPY
```

See Also

- Section A.26, "clonePdToPd"
- Section A.28, "clonePdToStorageArray"
- Section A.41, "create PhysicalDisk"
- Section A.76, "edit PhysicalDisk"
- Section A.59, "delete"
A.29 cloneVdToPd

Clones a virtual disk to a physical disk.

Syntax

```
cloneVdToPd VirtualDisk instance target=value cloneType= \{ SPARSE_COPY | NON_SPARSE_COPY | THIN_CLONE \}
```

Where `instance` is:

```
\{ id=value | name=value \}
```

Description

This command clones a virtual disk to a target physical disk.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>target=value</code></td>
<td>The physical disk on which to locate the cloned virtual disk.</td>
</tr>
<tr>
<td>`cloneType= { SPARSE_COPY</td>
<td>NON_SPARSE_COPY</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Examples

**Example A.40 Cloning a virtual disk to a physical disk**

```
OVM> cloneVdToPd VirtualDisk name=MyVMDisk target=MyDisk1 cloneType=SPARSE_COPY
```

See Also

- Section A.30, “cloneVdToRepo”
- Section A.135, “importVirtualDisk”
- Section A.51, “create VirtualDisk”
- Section A.93, “edit VirtualDisk”
- Section A.59, “delete”
A.30 cloneVdToRepo

Clones a virtual disk to a repository.

Syntax

```
cloneVdToRepo VirtualDisk instance target=value cloneType= { SPARSE_COPY | NON_SPARSE_COPY | THIN_CLONE }
```

Where instance is:

```
{ id=value | name=value }
```

Description

This command clones a virtual disk to a target repository.

The virtual disk instance must be in a storage repository on a physical disk.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target=value</td>
<td>The repository on which to locate the cloned virtual disk. The target repository must be on a physical disk.</td>
</tr>
<tr>
<td>cloneType= { SPARSE_COPY</td>
<td>NON_SPARSE_COPY</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

Example A.41 Cloning a virtual disk to a repository

```
OVM> cloneVdToRepo VirtualDisk name=MyVirtualDisk target=MyRepository cloneType=SPARSE_COPY
```

See Also

- Section A.29, "cloneVdToPd"
createVmFromAssembly (Deprecated)

- Section A.135, “importVirtualDisk”
- Section A.51, “create VirtualDisk”
- Section A.93, “edit VirtualDisk”
- Section A.59, “delete”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

A.31 createVmFromAssembly (Deprecated)

Creates a virtual machine template from an assembly file.

Important

This command has been deprecated. Instead, use the createVmFromVirtualAppliance command.

Syntax

createVmFromAssembly AssemblyVm instance

Where instance is:

{ id=value | name=value }

Description

This command creates a virtual machine template from an AssemblyVm object. An AssemblyVm object is created for each virtual machine in an assembly file when an assembly file is imported using the importAssembly (Deprecated) command. The virtual machine template files are created in the same storage repository as the original AssemblyVm object. To create a virtual machine from the template, use the clone Vm command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Note

Any create command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.
Examples

Example A.42 Creating virtual machines from an assembly

OVM> createVmFromAssembly AssemblyVm name=myassembly.ova

See Also

- Section A.131, “importAssembly (Deprecated)”
- Section A.24, “clone Vm”
- Section A.63, “edit Assembly (Deprecated)”
- Section A.64, “edit AssemblyVirtualDisk (Deprecated)”
- Section A.65, “edit AssemblyVm (Deprecated)”
- Section A.111, “getDescriptor”
- Section A.59, “delete”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”

A.32 createVmFromVirtualAppliance

Creates a virtual machine (template) from a virtual machine in a virtual appliance.

Syntax

createVmFromVirtualAppliance VirtualApplianceVm instance

Where instance is:

{ id=value | name=value }

Description

This command creates a virtual machine from a VirtualApplianceVm object. A VirtualApplianceVm object is created for each virtual machine in a virtual appliance file when a virtual appliance file is imported using the importVirtualAppliance command. The virtual machine files are created in the same storage repository as the original VirtualApplianceVm object. To see a list of VirtualApplianceVm objects, use the list VirtualApplianceVm command.

The virtual machine is created and listed in the VM Templates folder in the Repositories tab in Oracle VM Manager.

Options

The following table shows the available options for this command.
Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

Note

Any `create` command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

Examples

Example A.43 Creating a virtual machine (template) from a virtual appliance

```bash
OVM> createVmFromVirtualAppliance VirtualApplianceVm name=MyVm
```

See Also

- Section A.33, “createVmFromVirtualApplianceVm”
- Section A.91, “edit VirtualApplianceVm”
- Section A.89, “edit VirtualAppliance”
- Section A.90, “edit VirtualApplianceVirtualDisk”
- Section A.133, “importVirtualAppliance”
- Section A.106, “exportVirtualAppliance”
- Section A.111, “getDescriptor”
- Section A.59, “delete”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”

A.33 createVmFromVirtualApplianceVm

Creates a virtual machine from a virtual machine in a virtual appliance.

Syntax

```bash
createVmFromVirtualApplianceVm VirtualApplianceVm instance
```

Where `instance` is:

```bash
{id=value|name=value}
```

Description

This command creates a virtual machine from a VirtualApplianceVm object. A VirtualApplianceVm object is created for each virtual machine in a virtual appliance file when a virtual appliance file is imported using
the `importVirtualAppliance` command. The virtual machine files are created in the same storage repository as the original VirtualApplianceVm object. To see a list of VirtualApplianceVm objects, use the `list VirtualApplianceVm` command.

The virtual machine is created and deployed to the **Unassigned Virtual Machines** folder in the **Servers and VMs** tab in Oracle VM Manager.

### Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

**Note**

Any `create` command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

### Examples

**Example A.44 Creating a virtual machine from a virtual appliance**

```
OVM> createVmFromVirtualApplianceVm VirtualApplianceVm name=MyVm
```

### See Also

- Section A.32, "createVmFromVirtualAppliance"
- Section A.91, "edit VirtualApplianceVm"
- Section A.89, "edit VirtualAppliance"
- Section A.90, "edit VirtualApplianceVirtualDisk"
- Section A.133, "importVirtualAppliance"
- Section A.106, "exportVirtualAppliance"
- Section A.111, "getDescriptor"
- Section A.59, "delete"
- Section A.142, "refresh"
- Section A.138, "list"
- Section A.171, "show"

### A.34 create AccessGroup

Creates an access group.
Syntax

```plaintext
create AccessGroup name=value [description=value] on {FileServer | StorageArray } instance

Where instance is:

{ id=value | name=value }
```

Description

This command creates an access group for either a file server or storage array. Generic storage array plug-ins are not supported with this command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name for the access group.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the access group. value is a maximum of 4,000 characters. To set an empty description, use description=&quot;&quot;.</td>
</tr>
</tbody>
</table>
| { id=value | name=value } | The instance of the object using either the id or name option, for example name=MyFileServer.

Note

Any create command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

Examples

Example A.45 Creating an access group on a storage array

```
OVM> create AccessGroup name=MyAccessGroup on StorageArray name=MyISCSIServer
```

Example A.46 Creating an access group on a file system

```
OVM> create AccessGroup name=MyAccessGroup on FileServer name=MyNFSServer
```

See Also

- Section A.61, “edit AccessGroup”
- Section A.9, “add PhysicalDisk”
- Section A.152, “remove PhysicalDisk”
- Section A.39, “create FileSystem”
create AntiAffinityGroup

- Section A.72, “edit FileSystem”
- Section A.8, “add FileSystem”
- Section A.151, “remove FileSystem”
- Section A.11, “add Server”
- Section A.154, “remove Server”
- Section A.138, “list”
- Section A.171, “show”
- Section A.59, “delete”
- Section A.113, “getEvents”

A.35 create AntiAffinityGroup

Creates an anti affinity group in a server pool.

Syntax

```
create AntiAffinityGroup name=value [description=value] on ServerPool instance
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command creates an anti affinity group in a server pool. To add a virtual machine to an anti affinity group, use the `add Vm` command. To remove a virtual machine from an anti affinity group, use the `remove Vm` command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the anti affinity group.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the anti affinity group. value is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Note

Any `create` command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one
object of the same attribute type as a parameter always results in the last attribute value taking precedence.

Examples

Example A.47 Creating an anti affinity group

OVM> create AntiAffinityGroup name=MyAAGroup on ServerPool name=MyServerPool

See Also

• Section A.62, “edit AntiAffinityGroup”
• Section A.16, “add Vm”
• Section A.159, “remove Vm”
• Section A.59, “delete”
• Section A.138, “list”
• Section A.171, “show”

A.36 create BondPort

Creates a bond port on an Oracle VM Server.

Syntax

create BondPort ethernetPorts=value mode= {ACTIVE_PASSIVE | LINK_AGGREGATION | LOAD_BALANCED } mtu=value [interfaceName=value] name=value [description=value] on Server instance

Where instance is:

{ id=value | name=value }

Description

This command creates a bond port on an Oracle VM Server.

To configure the IP address for a bond port, use the embeddedCreate command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ethernetPorts=value</td>
<td>The name or ID of at least two Ethernet ports in a comma separated list. The name or ID must match the name or ID as it is stored for each port within Oracle VM Manager.</td>
</tr>
<tr>
<td>mode= {ACTIVE_PASSIVE</td>
<td>LINK_AGGREGATION</td>
</tr>
</tbody>
</table>
### Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mtu=value</td>
<td>The MTU value. May be an integer between 1500 and 64000.</td>
</tr>
<tr>
<td>interfaceName=value</td>
<td>An optional name for the bond in the format bondN, for example bond1, or bond2. If you do not enter a name, the default of bondN is used, where N is the next available bond number. This cannot be changed after the bond is created.</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the bond.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the bond. value is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

**Note**

Any `create` command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

#### Examples

**Example A.48 Creating a bond port**

```
OVM> create BondPort mode=LINK_AGGREGATION mtu=1500 name=MyPortBond \\
    ethernetPorts="0004fb000200000d9349f2f8ba06d4,0004fb000200000c7a5f26641825be5" \\
    on Server name=MyServer
```

```
OVM> create bondPort name=MyPortBond2 mode=ACTIVE_PASSIVE mtu=1500 \\
    ethernetPorts="eth3 on MyServer,eth5 on MyServer" on server name=MyServer
```

**See Also**

- Section A.7, “add BondPort”
- Section A.66, “edit BondPort”
- Section A.150, “remove BondPort”
- Section A.153, “remove Port”
- Section A.102, “embeddedCreate”
- Section A.103, “embeddedDelete”
- Section A.104, “embeddedEdit”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

### A.37 create CpuCompatibilityGroup
Syntax

create CpuCompatibilityGroup name=value [description=value]

Description

This command creates a CPU compatibility group to which Oracle VM Servers sharing a common processor can be assigned.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name for the CPU compatibility group.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the CPU compatibility group. value is a maximum of 4,000 characters. To set an empty description, use description=&quot;&quot;.</td>
</tr>
</tbody>
</table>

Note

Any create command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

Examples

Example A.49 Creating a CPU compatibility group

OVM> create CpuCompatibilityGroup name=MyCPUGroup

See Also

- Section A.69, “edit CpuCompatibilityGroup”
- Section A.11, “add Server”
- Section A.154, “remove Server”
- Section A.138, “list”
- Section A.171, “show”
- Section A.59, “delete”
- Section A.113, “getEvents”

A.38 create FileServer
Discover a file server.

Syntax

create FileServer plugin=value accessHost=value adminServers=value
[refreshServers=value][uniformedExports= {Yes|No}][name=value][
description=value]

Description

This command discovers a file server.

After discovering a file server, you should:

- Optionally, if you are using non-uniformed file system exports, you can create an access group using the create AccessGroup command. Add file systems to the access group using the add FileSystem command. Add Oracle VM Servers to the access group using the add Server command.
- Refresh the file server and file systems using the refresh command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plugin=value</td>
<td>The name or ID of storage plug-in to use for the file server to be discovered. To obtain a list of existing plug-ins enter: OVM&gt; list FileServerPlugin If a vendor specific plug-in is configured it is made available as an option here.</td>
</tr>
<tr>
<td>accessHost=value</td>
<td>The host name or IP address for the file server to be discovered.</td>
</tr>
<tr>
<td>adminServers=value</td>
<td>The names or IDs of the Oracle VM Servers to perform administration on the file server, in a comma separated list.</td>
</tr>
<tr>
<td>refreshServers=value</td>
<td>Optionally, the host names or IP addresses of the Oracle VM Servers to perform refresh jobs on the file server, in a comma separated list.</td>
</tr>
<tr>
<td>uniformedExports=</td>
<td>Whether the file server has uniformed file system exports. The default is Yes.</td>
</tr>
<tr>
<td></td>
<td>{Yes</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the file server.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the file server. value is a maximum of 4,000 characters. To set an empty description, use description=&quot;&quot;.</td>
</tr>
</tbody>
</table>

Note

Any create command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one
Examples

object of the same attribute type as a parameter always results in the last attribute value taking precedence.

Examples

Example A.50 Discovering a file server

```
OVMe create FileServer plugin="Oracle Generic Network File System" \
  accessHost=10.172.76.125 adminServers="MyServer1.virtlab.info,MyServer2.virtlab.info" \
  name=MyNFSServer
```

See Also

- Section A.70, “edit FileServer”
- Section A.11, “add Server”
- Section A.154, “remove Server”
- Section A.4, “addAdminServer”
- Section A.147, “removeAdminServer”
- Section A.6, “addRefreshServer”
- Section A.149, “removeRefreshServer”
- Section A.142, “refresh”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

A.39 create FileSystem

Creates an OCFS2 file system on a physical disk on a file server.

Syntax

```
create FileSystem physicalDisk=value name=value[ description=value ] on FileServer instance
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command creates an OCFS2 file system on a physical disk on a file server. When creating a file system on an NFS file server, you can use the file server itself to create the file system. When creating an OCFS2 file system on a LUN, you should use a local file server on an Oracle VM Server that has access to the LUN to create the file system. See Section 7.2, “To create a storage repository on a storage array” for an example.
Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>physicalDisk=value</td>
<td>The name or ID of the physical disk on which to create the OCFS2 file system.</td>
</tr>
<tr>
<td>name=value</td>
<td>A name for the file system.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the file system. value is a maximum of 4,000 characters. To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Note

Any create command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

Examples

Example A.51 Creating an OCFS2 file system on physical disk on a file server

```bash
OVM> create FileSystem physicalDisk=0004fb0000180000e3c93dc542901b7a name=MyRepoFileSystem 
     on FileServer id=0004fb00000900007e1ce0c83b3f136f
```

Example A.52 Creating an OCFS2 file system on a LUN

```bash
OVM> create FileSystem name=MyRepoFileSystem physicalDisk="MyLUN1" on FileServer 
     name="Local FS MyServer1"
```

See Also

- Section A.72, "edit FileSystem"
- Section A.8, "add FileSystem"
- Section A.151, "remove FileSystem"
- Section A.34, "create AccessGroup"
- Section A.61, "edit AccessGroup"
- Section A.11, "add Server"
- Section A.154, "remove Server"
- Section A.138, "list"
- Section A.171, "show"
- Section A.59, "delete"
A.40 create Network

Creates an Ethernet-based network.

Syntax

```
create Network [roles= { MANAGEMENT | LIVE_MIGRATE | CLUSTER_HEARTBEAT |
VIRTUAL_MACHINE | STORAGE }] name=value [description=value] [on Server instance]
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command creates an Ethernet-based network. To create a local network on an Oracle VM Server, use the `on Server instance` option. You can only create a local network for virtual machine networks (using the `roles=VIRTUAL_MACHINE` option), as shown in Example A.54, "Creating a local network on an Oracle VM Server". You cannot use a local network for traffic such as storage, or cluster heartbeat. If you attempt to create a local network and do not specify the role, its value is automatically set to `VIRTUAL_MACHINE`.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`roles= { MANAGEMENT</td>
<td>LIVE_MIGRATE</td>
</tr>
<tr>
<td><code>name=value</code></td>
<td>A name to identify the network.</td>
</tr>
<tr>
<td><code>description=value</code></td>
<td>Optional description for the network. <code>value</code> is a maximum of 4,000 characters. To set an empty description, use <code>description='''</code>.</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Note

Any `create` command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.
Examples

Example A.53 Creating a network

OVM> create Network name=MyVMNetwork roles=VIRTUAL_MACHINE

Example A.54 Creating a local network on an Oracle VM Server

OVM> create Network name=MyLocalNetwork roles=VIRTUAL_MACHINE on Server name=MyServer

See Also

- Section A.10, “add Port”
- Section A.77, “edit Port”
- Section A.153, “remove Port”
- Section A.74, “edit Network”
- Section A.36, “create BondPort”
- Section A.138, “list”
- Section A.171, “show”

A.41 create PhysicalDisk

Creates a physical disk on a volume group.

Syntax

create PhysicalDisk size=value [extraInfo=value] shareable= {Yes|No} thinProvision= {Yes|No} userFriendlyName=value name=value [description=value] on VolumeGroup instance

Where instance is:

{ id=value | name=value }

Description

This command creates a physical disk on a volume group. Local storage and generic storage plug-ins are not supported with this command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>size=value</td>
<td>The size of the physical disk in GiB. The value can be from 1 to 2048.</td>
</tr>
<tr>
<td>extraInfo=value</td>
<td>The value of the extra information field. Oracle VM Manager does not use the extra information field. However, the value that you set might be passed to a storage array plug-in. Refer</td>
</tr>
</tbody>
</table>
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>shareable= { Yes</td>
<td>No }</td>
</tr>
<tr>
<td>thinProvision= { Yes</td>
<td>No }</td>
</tr>
<tr>
<td>userFriendlyName=value</td>
<td>A user friendly name to identify the disk.</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the physical disk.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the physical disk. value is a maximum of 4,000 characters. To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

### Note

Any create command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

### Examples

**Example A.55 Creating a physical disk on a volume group**

```
OVM> create PhysicalDisk size=10 name=MyPhysicalDisk shareable=No \ 
thinProvision=No userFriendlyName="My Disk" on VolumeGroup \ 
id='Storage_Volume_Group 0004fb0000090000325a36dad3b3b7d8'
```

### See Also

- Section A.9, “add PhysicalDisk”
- Section A.76, “edit PhysicalDisk”
- Section A.152, “remove PhysicalDisk”
- Section A.161, “resize”
- Section A.59, “delete”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

## A.42 create Repository

Creates a storage repository.
Syntax

create Repository [sharePath=value] name=value [description=value] on FileSystem instance

Where instance is:

{ id=value | name=value }

Note that if the instance name contains forward slashes and you need to quote the name, you must escape forward slashes by using additional forward slashes. This is illustrated in the example.

Description

This command creates a storage repository on a file system. To create a repository on a LUN, you should first create an OCFS2 file system on it using the create FileSystem command.

After you create a repository, you should refresh it.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sharePath=value</td>
<td>A path to a subdirectory on the selected file system.</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the storage repository.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the storage repository. value is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Note

Any create command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

Examples

Example A.56 Creating a storage repository on a file server

OVM> create Repository name=MyRepository on FileSystem \name="nfs on 10.172.76.125://mnt//vol2//repo01"

Example A.57 Creating a storage repository on a SAN server

OVM> create Repository name=MyRepository on FileSystem name=MyRepoFileSystem

See Also

- Section A.11, “add Server”
- Section A.142, “refresh”
A.43 create RepositoryExport

Creates a repository export.

Syntax

```
create RepositoryExport clientHostName=value name=value repository=value [description=value] options=value on Serverinstance
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command creates an export on an Oracle VM Server to enable access for a third party back up tool to back up the contents of an OCFS2-based storage repository.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clientHostName=value</td>
<td>The hostname or IP address of the computer for which to grant access to the storage repository contents. This is likely to be the machine on which the third party back up and restore software is running.</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the export on the file server.</td>
</tr>
<tr>
<td>repository=value</td>
<td>An OCFS2-based storage repository presented to the Oracle VM Server. This is the repository to configure for back up.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the export on the file server. <code>value</code> is a maximum of 4,000 characters. To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>options=value</td>
<td>The parameters to include in the NFS mount configuration, for example: <code>rw, async, no_root_squash, wdelay</code>. When no <code>options</code> are specified, the default NFS options on the Oracle VM Server are used.</td>
</tr>
<tr>
<td>Serverinstance</td>
<td>An Oracle VM Server on which the storage repository is presented.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

**Example A.58 Creating a repository export on an Oracle VM Server**

```
OVM> create RepositoryExport clientHostName=10.172.76.146 name="My NFS Export" \
```
A.44 create ServerController

Creates a server controller object to configure IPMI on an Oracle VM Server.

Syntax

```
create ServerController ipAddress=value userName=value [password=value] name=value [description=value] on Server instance
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command creates a server controller object to configure IPMI on an Oracle VM Server.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>virtualIP=value</code></td>
<td>The IP address of the IPMI.</td>
</tr>
<tr>
<td><code>userName=value</code></td>
<td>The user name for the IPMI.</td>
</tr>
<tr>
<td><code>password=value</code></td>
<td>An optional password for the IPMI.</td>
</tr>
<tr>
<td><code>name=value</code></td>
<td>A name to identify the server control object.</td>
</tr>
<tr>
<td><code>description=value</code></td>
<td>Optional description for the server control object. <code>value</code> is a maximum of 4,000 characters. To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
</tbody>
</table>

Examples

**Example A.59 Configuring IPMI for an Oracle VM Server**

```
OVM> create ServerController ipAddress=192.168.10.3 userName=admin password=password
```
A.45 create ServerPool

Creates a server pool.

Syntax

```plaintext
create ServerPool
```

```plaintext
  clusterEnable= { Yes | No }[[
  clusterTimeout= value][
  fileSystem=value][
  physicalDisk=value][
  keymapName= { en-us|ar|da|de|de-ch|en-gb|es|et|fi|fo|fr|fr-be|fr-ca|fr-ch|hr|hu|is|it|ja|lt|lv|mk|nl|nl-be|
  No|pl|pt|pt-br|ru|sl|sv|th|tr}][
  migrateUsingSsl= { Yes | No }][
  startPolicy= { BEST_SERVER|BALANCE_SERVER|CURRENT_SERVER }][
  policyMode= { OFF|DRS|DPM }][
  policyCpuEnable= { Yes | No }][
  policyPeriod=value][
  policyCpuThreshold=value][
  name=value [description=value][
  virtualIP=value]
```

Description

This command creates a server pool.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clusterEnable= { Yes</td>
<td>No }</td>
</tr>
<tr>
<td>clusterTimeout= value</td>
<td>Set the timeout value for the cluster in seconds. value is an integer between 30 and 300. SPARC-based server pools do not recognise the cluster timeout parameter. Setting this value for a server pool consisting of SPARC servers has no effect and the parameter is ignored by the Oracle VM Agent for SPARC.</td>
</tr>
<tr>
<td>fileSystem=value</td>
<td>The file system to use for the server pool file system. Note that if you specify the name as the value for the file system, and the name is specified in quotes, any forward slashes in the name must be escaped using additional forward slashes. This is illustrated in the examples for this command.</td>
</tr>
<tr>
<td>physicalDisk=value</td>
<td>The physical disk to use for the server pool file system.</td>
</tr>
</tbody>
</table>
## Option | Description
--- | ---

You cannot create a server pool file system on a local physical disk as the server pool file system needs to be accessible by all Oracle VM Servers in the server pool.

| keymapName= {en-us | ar | da | de | de-ch | en-gb | es | et | fi | fo | fr | fr-be | fr-ca | fr-ch | hr | hu | it | ja | lt | lv | mk | nl | nl-be | No | pl | pt | pt-br | ru | sl | sv | th | tr} | The key mapping to be used when connecting to a virtual machine's console.

| migrateUsingSsl= {Yes | No} | Whether to enable secure migration of virtual machines using SSL.

| startPolicy= {BEST_SERVER | BALANCE_SERVER | CURRENT_SERVER} | The policy by which virtual machines are located when created in the server pool. If none is provided, the CURRENT_SERVER option is used by default.

| policyMode= {OFF | DRS | DPM} | Set the policy to use for the server pool.

| policyCpuEnable= {Yes | No} | Set whether to enable the policy set in the policyMode option for the server pool.

| policyPeriod=value | The time period for the policy job to run. This sets the policy job to run every $n$ minutes, for example, 10 sets the policy job to run every 10 minutes. $value$ can be an integer between 2 and 60.

| policyCpuThreshold=value | The maximum amount of CPU percentage usage allowed before the policy must be enacted. $value$ can be an integer between 25 and 99.

| virtualIP=value | An optional virtual IP address for the server pool. This is a deprecated parameter. Only specify a virtual IP address for a server pool if you intend to add Oracle VM Servers running a version of the software from prior to a 3.4 release.

| name=value | A name to identify the server pool.

| description=value | Optional description for the server pool. $value$ is a maximum of 4,000 characters.

| Note | Any create command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

## Examples

### Example A.60 Creating a clustered server pool

```
OVM> create ServerPool clusterEnable=Yes \
```
filesystem="nfs on 10.172.76.125://mnt//vol1//poolfs01" name=MyServerPool \
  description='Clustered server pool'

Example A.61 Creating an unclustered server pool

OVMe create ServerPool clusterEnable=No name=MyServerPool \
  description='Unclustered server pool'

See Also

- Section A.82, “edit ServerPool”
- Section A.46, “create ServerPoolNetworkPolicy”
- Section A.83, “edit ServerPoolNetworkPolicy”
- Section A.11, “add Server”
- Section A.154, “remove Server”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

A.46 create ServerPoolNetworkPolicy

Creates a server pool network policy.

Syntax

create ServerPoolNetworkPolicy[policyEnable= {Yes | No}][policyThreshold=value] 
  network=value name=value [description=value] on ServerPool instance

Where instance is:

{ id=value | name=value }

Description

This command creates a server pool network policy. A server pool network policy is the object that controls
DPM/DRS behavior of the virtual machines associated with the server pool based on network bandwidth
usage.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>policyEnable= {Yes</td>
<td>No }</td>
</tr>
</tbody>
</table>
### Option | Description
---|---
`policyThreshold=value` | The percentage (%) of network bandwidth the policy uses to move virtual machines. `value` can be an integer between 0 and 100.

`network=value` | The name or ID of the network associated with the policy, which is used to administer the policy for the server pool.

`name=value` | A name to identify the server pool network policy.

`description=value` | Optional description for the server pool network policy. `value` is a maximum of 4,000 characters. To set an empty description, use `description=""`.

### Note
Any `create` command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

### Examples

#### Example A.62 Creating a server pool network policy

```shell
OVMD> create ServerPoolNetworkPolicy network=MyNetwork policyEnable=Yes name=MyNetworkPolicy on ServerPool name=MyServerPool
```

### See Also

- Section A.83, “edit ServerPoolNetworkPolicy”
- Section A.5, “addPolicyServer”
- Section A.148, “removePolicyServer”
- Section A.45, “create ServerPool”
- Section A.82, “edit ServerPool”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

### A.47 create ServerUpdateGroup

Creates an Oracle VM Server update group in a server pool.

#### Syntax

```sql
create ServerUpdateGroup name=value [description=value] on ServerPool instance
```

Where `instance` is:
This command creates an Oracle VM Server update group in a server pool. This allows you to override the default update group for a server pool. The default update groups:

- For x86-based server pools, have the ID of GlobalX86ServerUpdateConfiguration.
- For SPARC-based server pools, have the ID of GlobalSparcServerUpdateConfiguration.

When you have created a server update group for a server pool, you should then assign the update repository using the create ServerUpdateRepository command. To update an Oracle VM Server, use the upgrade command.

### Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the server update group.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the server update group. value is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td>name=MyserverPool</td>
<td>To set an empty description, use description=&quot;&quot;.</td>
</tr>
</tbody>
</table>

### Examples

**Example A.63 Creating an update group for a server pool**

```bash
OVMM> create ServerUpdateGroup name=MyServerUpdateGroup on ServerPool name=MyServerPool
```

### See Also

- Section A.84, “edit ServerUpdateGroup”
- Section A.48, “create ServerUpdateRepository”
- Section A.85, “edit ServerUpdateRepository”
- Section A.19, “checkUpToDate”
- Section A.181, “upgrade”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

**A.48 create ServerUpdateRepository**
Creates an Oracle VM Server update repository.

Syntax

```
create ServerUpdateRepository repositoryName=value url=value enabled= { Yes | No } pkgSignatureType= { NONE | GPG | CA } [ pkgSignatureKey=value ] name=value [ description=value ] on ServerUpdateGroup instance
```

Where instance is:

```
{ id=value | name=value }
```

Description

This command sets the location for the YUM (for x86) or IPS (for SPARC) repository that contains the files to update Oracle VM Servers. This command creates an Oracle VM Server update repository and assigns it to a server update group. The default server update groups:

- For x86-based Oracle VM Servers, have the ID of `GlobalX86ServerUpdateConfiguration`.
- For SPARC-based Oracle VM Servers, have the ID of `GlobalSparcServerUpdateConfiguration`.

To override either of these default repositories and create a repository that is restricted to a particular server pool, you should first create a server update group for the server pool using the `create ServerUpdateGroup` command, then create an update repository and assign it to the new server update group. To update an Oracle VM Server, use the `upgrade` command.

Note that when a new server update repository is created, the repository is added to each of the servers that belong to the server update group. If there is a problem adding the repository to a server in the server update group, an error event is generated for that server within Oracle VM Manager. Oracle VM Manager does not attempt to validate the repository before it is added to each server. If the repository is invalid or, in the case of a SPARC repository, the repository name does not match a valid publisher at the URL specified, an error event is generated for the servers affected.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>repositoryName=value</td>
<td>A name for the server update repository. This may only contain alphanumeric characters and underscores. No spaces are permitted. For SPARC repositories, this must match a valid publisher for the repository, hosted at the provided URL.</td>
</tr>
<tr>
<td>url=value</td>
<td>The URL to access the repository. If you enclose the URL in quotes, you must escape each forward slash (/) with another, for example:</td>
</tr>
<tr>
<td></td>
<td><code>url=&quot;http://10.172.77.200//ovs&quot;</code></td>
</tr>
<tr>
<td>enabled= { Yes</td>
<td>No }</td>
</tr>
<tr>
<td>pkgSignatureType= { NONE</td>
<td>GPG</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pkgSignatureKey=value</td>
<td>The verification signature for the repository, for example, the location of the GPG key using any of the HTTP, FTP, FILE or HTTPS protocols.</td>
</tr>
<tr>
<td></td>
<td>If you enclose the value for the option in quotes, you must escape each forward slash (/) with another, for example:</td>
</tr>
<tr>
<td></td>
<td>pkgSignatureKey=&quot;http:///10.172.77.200//ovs//RPM-GPG-KEY&quot;</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the server update repository.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the server update repository. value is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use description=&quot;&quot;</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.64 Creating an Oracle VM Server update repository**

```
OVM> create ServerUpdateRepository repositoryName=MyUpdateRepository url=http://10.172.77.200/ovs \
   enabled=Yes pkgSignatureType=GPG pkgSignatureKey=http://10.172.77.200/ovs/RPM-GPG-KEY \ 
   name=Myx86Repository on ServerUpdateGroup id=GlobalX86ServerUpdateConfiguration
```

**See Also**

- Section A.85, “edit ServerUpdateRepository”
- Section A.47, “create ServerUpdateGroup”
- Section A.84, “edit ServerUpdateGroup”
- Section A.19, “checkUpToDate”
- Section A.181, “upgrade”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

**A.49 create StorageArray**

Discovers a storage array.

**Syntax**

```
create StorageArray plugin=value storageType={FIBRECHANNEL | ISCSI | LOCAL | UNKNOWN} [ storageName=value] [accessHost=value] [accessPort=value] [accessUsername=value accessPassword=value] [useChap={Yes | No}] [adminHost=value adminUserName=value]
```
Description

This command discovers a storage array and adds it to Oracle VM Manager. If you are adding a non-generic storage array also enter the additional plug-in options to enable Oracle VM Manager to access the storage array's configuration management functions using the adminHost option. To add more access hosts to enable multipathing on ISCSI servers, use the addAccessHost command.

After discovering a storage array, you should add storage initiators to it, add admin servers to it, validate it, then refresh it.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plugin=value</td>
<td>The name or ID of the Oracle VM Storage Connect plug-in to use for the storage array.</td>
</tr>
<tr>
<td>storageType={FIBRECHANNEL</td>
<td>ISCSI</td>
</tr>
<tr>
<td>storageName=value</td>
<td>A name to identify the storage for a storage array behind a concentrator.</td>
</tr>
<tr>
<td>accessHost=value</td>
<td>The hostname or IP address for the storage array. This is not applicable to fibre channel storage arrays.</td>
</tr>
<tr>
<td>accessPort=value</td>
<td>The port on which access to the storage array is allowed. When adding iSCSI storage, add the access port as well. The default access port for iSCSI is 3260. If not specified, the default port is used automatically.</td>
</tr>
<tr>
<td>accessUsername=value</td>
<td>A username with administrative access to the storage array, used with accesshostname. This option should only be used where CHAP is enabled on the storage array.</td>
</tr>
<tr>
<td>accessPassword=value</td>
<td>The password for the accessusername user. This option should only be used where CHAP is enabled on the storage array.</td>
</tr>
<tr>
<td>useChap={Yes</td>
<td>No}</td>
</tr>
<tr>
<td>adminHost=value</td>
<td>The host name or IP address where administrative access to the storage array is allowed.</td>
</tr>
<tr>
<td>adminUserName=value</td>
<td>A user name with administrative access to the storage array, used with adminHost.</td>
</tr>
<tr>
<td>adminPassword=value</td>
<td>The administrator password for the adminUserName user.</td>
</tr>
<tr>
<td>pluginPrivateData=value</td>
<td>This option is used to pass additional parameters that a non-generic Oracle VM Storage Connect plug-in may accept to control functionality. For instance, in the case of an Oracle NetApp file system, you can enable SSL using this parameter in the following way: pluginPrivateData=&quot;ssl=Yes&quot;</td>
</tr>
</tbody>
</table>

Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the storage array.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the storage array. value is a maximum of 4,000 characters. To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>lipScan={Yes</td>
<td>No}</td>
</tr>
</tbody>
</table>

Note

Any create command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

Examples

Example A.65 Discovering a storage array

```
OVM> create StorageArray plugin="Oracle Generic SCSI Plugin" name=MyISCSIServer \
    storageType=ISCSI accessHost=10.172.76.130 accessPort=3260
```

See Also

- Section A.3, “addAccessHost”
- Section A.86, “edit StorageArray”
- Section A.11, “add Server”
- Section A.154, “remove Server”
- Section A.4, “addAdminServer”
- Section A.147, “removeAdminServer”
- Section A.142, “refresh”
- Section A.34, “create AccessGroup”
- Section A.13, “add StorageInitiator”
- Section A.182, “validate”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”
A.50 create Tag

Creates a tag.

Syntax

```
create Tag name=value[description=value]
```

Description

This command creates a tag to identify and group objects.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the tag.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the tag. value is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use description=&quot;&quot;.</td>
</tr>
</tbody>
</table>

Note

Any `create` command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

Examples

Example A.66 Creating a tag

```
OVM> create Tag name=MyTag description="My tag."
```

See Also

- Section A.88, “edit Tag”
- Section A.14, “add Tag”
- Section A.157, “remove Tag”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

A.51 create VirtualDisk

Creates a virtual disk.
Syntax

```
create VirtualDisk size=value shareable= {Yes | No} sparse= {Yes | No} name=value [description=value] on Repository instance
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command creates a virtual disk in a storage repository.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>size=value</code></td>
<td>The size of the virtual disk in GiB. <code>value</code> can be an integer between 1 and 1000000.</td>
</tr>
<tr>
<td>`shareable= {Yes</td>
<td>No}`</td>
</tr>
<tr>
<td>`sparse= {Yes</td>
<td>No}`</td>
</tr>
<tr>
<td><code>name=value</code></td>
<td>A name to identify the virtual disk.</td>
</tr>
<tr>
<td><code>description=value</code></td>
<td>Optional description for the virtual disk. <code>value</code> is a maximum of 4,000 characters. To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

Note

Any `create` command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

Examples

Example A.67 Creating a virtual disk in a storage repository

```
OVM> create VirtualDisk name=MyVMDisk size=10 sparse=Yes shareable=No on Repository \name=MyRepository
```

See Also

- Section A.135, “importVirtualDisk”
- Section A.93, “edit VirtualDisk”
A.52 create VlanInterface

Creates a VLAN interface.

Syntax

```plaintext
create VlanInterface vlanId=value [mtu=value] name=value [description=value] on {Port | BondPort} instance
```

Where `instance` is:

```plaintext
{ id=value | name=value }
```

Description

This command creates a VLAN interface on either a port or a bond port.

To configure the IP address for a VLAN interface, use the `embeddedCreate` command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>vlanId=value</code></td>
<td>The ID of the VLAN interface. May be an integer between 2 and 4095.</td>
</tr>
<tr>
<td><code>mtu=value</code></td>
<td>The MTU value. May be an integer between 1500 and 64000.</td>
</tr>
<tr>
<td><code>name=value</code></td>
<td>A name to identify the VLAN interface.</td>
</tr>
<tr>
<td><code>description=value</code></td>
<td>Optional description for the VLAN interface. <code>value</code> is a maximum of 4,000 characters. To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Note

Any `create` command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.
Examples

Example A.68 Creating a VLAN interface

```
OVM> create VlanInterface vlanId=20 mtu=1500 name=MyVLANInterface on Port id=0004fb00000000229dbcccf1?efec5
```

See Also

- Section A.94, “edit VlanInterface”
- Section A.15, “add VlanInterface”
- Section A.7, “add BondPort”
- Section A.66, “edit BondPort”
- Section A.150, “remove BondPort”
- Section A.153, “remove Port”
- Section A.102, “embeddedCreate”
- Section A.103, “embeddedDelete”
- Section A.104, “embeddedEdit”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

A.53 create Vm

Creates a virtual machine.

Syntax

```
create Vm [memory=value] [memoryLimit=value] [cpuCount=value] [cpuCountLimit=value] [cpuPriority=value] [cpuUtilizationCap=value] [highAvailability={Yes|No}] [hugePagesEnabled={Yes|No}] [osType=value] [restartActionOnCrash={RESTART|STOP|RESTART_AFTER_DUMP|STOP_AFTER_DUMP}] [mouseType={OS_DEFAULT|PS2_MOUSE|USB_MOUSE|USB_TABLET}] [domainType={XEN_HVM|XEN_PVM|LDOMS_PVM|UNKNOWN}] [keymapName={en-us|ar|da|de|de-ch|en-gb|es|et|fi|fo|fr|fr-be|fr-ca|fr-ch|hr|hu|is|it|ja|lt|lv|mk|nl|nl-be|No|pl|pt|pt-br|ru|sl|sv|th|tr}] [bootOrder={PXE|DISK|CDROM}] [networkInstallPath=value] [repository=value] [server=value] [startPolicy={BEST_SERVER|BALANCE_SERVER|CURRENT_SERVER|USE_POOL_POLICY}] [viridian={Yes|No}] name=value [description=value] on ServerPool instance
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command creates a virtual machine.
The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>memory=value</code></td>
<td>The memory size the virtual machine is allocated in MB. May be an integer between 32 and 512000. The default is 512.</td>
</tr>
<tr>
<td><code>memoryLimit=value</code></td>
<td>The maximum memory size the virtual machine can be allocated in MB. May be an integer between 32 and 512000. The default is 512.</td>
</tr>
<tr>
<td><code>cpuCount=value</code></td>
<td>The number of processors the virtual machine is allocated. May be an integer between 1 and 999999, but cannot exceed the maximum limit for the domain type, as follows:</td>
</tr>
<tr>
<td></td>
<td>• PVM: 256.</td>
</tr>
<tr>
<td></td>
<td>Important</td>
</tr>
<tr>
<td></td>
<td>As of Oracle VM Release 3.4.6, support for PVM guests is removed.</td>
</tr>
<tr>
<td></td>
<td>For more information, see Disabling Paravirtualized Guests on Oracle VM Server in the Oracle VM Administrator's Guide.</td>
</tr>
<tr>
<td></td>
<td>• HVM: 128; or 32 for Microsoft Windows guests.</td>
</tr>
<tr>
<td></td>
<td>Note</td>
</tr>
<tr>
<td></td>
<td>As of Release 3.4.6, the HVM limit is 64 for Microsoft Windows guests.</td>
</tr>
<tr>
<td></td>
<td>• PVHVM: 128; or 32 if using Oracle VM Paravirtual Drivers for Microsoft Windows.</td>
</tr>
<tr>
<td></td>
<td>Note</td>
</tr>
<tr>
<td></td>
<td>As of Release 3.4.6, the PVHVM limit is 64 if using Oracle VM Paravirtual Drivers for Microsoft Windows.</td>
</tr>
<tr>
<td></td>
<td>• LDOMS_PVM: Equivalent to the number of available CPUs on the server.</td>
</tr>
<tr>
<td><code>cpuCountLimit=value</code></td>
<td>The maximum number of processors the virtual machine can be allocated. May be an integer between 1 and 999999, but cannot exceed the maximum limit for the domain type, as follows:</td>
</tr>
<tr>
<td></td>
<td>• PVM: 256.</td>
</tr>
<tr>
<td></td>
<td>Important</td>
</tr>
<tr>
<td></td>
<td>As of Oracle VM Release 3.4.6, support for PVM guests is removed.</td>
</tr>
</tbody>
</table>
### Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>For more information, see <a href="https://docs.oracle.com/en-us/ovmg/doc/ovmg/3.4.6.Disabling.html">Disabling Paravirtualized Guests on Oracle VM Server</a> in the Oracle VM Administrator's Guide.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>As of Release 3.4.6, the HVM limit is 64 for Microsoft Windows guests.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>As of Release 3.4.6, the PVHVM limit is 64 if using Oracle VM Paravirtual Drivers for Microsoft Windows.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>The HugePages feature is deprecated in Oracle VM Release 3.4.1. You should not enable HugePages when creating or editing virtual machines in the Oracle VM Manager Web Interface or Oracle VM Manager Command Line Interface. This feature will be removed in a future release of Oracle VM.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>If you have HugePages enabled for any PVM guests, Oracle recommends that you change the domain type for virtual machines from Paravirtualized (PVM) to Hardware virtualized, with paravirtualized drivers (PVHVM). If you cannot change the domain type for a virtual machine, you should disable the HugePages setting and then restart the virtual machine.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpuPriority=value</td>
<td>The CPU priority of the virtual machine. A value between 1 and 100; the higher the number, the more priority the CPU is given.</td>
</tr>
<tr>
<td>cpuUtilizationCap=value</td>
<td>The maximum percentage to which the virtual CPUs can receive scheduled time. A value between 10 and 100; the higher the number, the more scheduled time the CPU is given.</td>
</tr>
<tr>
<td>highAvailability= {Yes</td>
<td>Whether to enable High Availability (HA).</td>
</tr>
<tr>
<td>hugePagesEnabled= {Yes</td>
<td>Whether to enable HugePages.</td>
</tr>
<tr>
<td></td>
<td>No}</td>
</tr>
</tbody>
</table>
### Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>osType=value</code></td>
<td>The operating system of the virtual machine. To find the operating system type, use the <code>getVmOsTypes</code> command.</td>
</tr>
<tr>
<td>`[restartActionOnCrash={RESTART</td>
<td>STOP</td>
</tr>
<tr>
<td>`mouseType={OS_DEFAULT</td>
<td>PS2_MOUSE</td>
</tr>
<tr>
<td>`domainType={XEN_HVM</td>
<td>XEN_HVM_PV_DRIVERS</td>
</tr>
<tr>
<td>`keymapName={en-us</td>
<td>ar</td>
</tr>
<tr>
<td>`bootOrder={PXE</td>
<td>DISK</td>
</tr>
<tr>
<td><code>networkInstallPath=value</code></td>
<td>The location at which the installation media (mounted ISO file) is located when creating a PVM guest.</td>
</tr>
<tr>
<td><code>repository=value</code></td>
<td>The name or ID of the storage repository in which to create the virtual machine configuration file.</td>
</tr>
</tbody>
</table>
### Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server=value</td>
<td>The name or ID of the Oracle VM Server on which to create the virtual machine.</td>
</tr>
<tr>
<td>startPolicy= { BEST_SERVER</td>
<td>BALANCE_SERVER</td>
</tr>
<tr>
<td>viridian= { Yes</td>
<td>No }</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the virtual machine.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the virtual machine. value is a maximum of 4,000 characters. To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

#### Note

Any `create` command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

### Examples

#### Example A.69 Creating a virtual machine

```
OVM> create Vm name=MyVM repository=MyRepository domainType=XEN_HVM \ 
    server=MyServer startPolicy=USE_POOL_POLICY on ServerPool name=MyServerPool
```

### See Also

- Section A.126, “getVmOsTypes”
- Section A.24, “clone Vm”
- Section A.95, “edit Vm”
- Section A.16, “add Vm”
- Section A.177, “start”
- Section A.138, “list”
- Section A.171, “show”
create VmCloneCustomizer

A.54 create VmCloneCustomizer

Creates a clone customizer for a virtual machine.

Syntax

create VmCloneCustomizer name=value [description=value] on Vm instance

Where instance is:

{ id=value | name=value }

Description

This command allows you to create a new clone customizer for a virtual machine. When the clone customizer is created, you may want to create clone network and storage mappings that can be used by the clone customizer. See Section A.55, "create VmCloneNetworkMapping" and Section A.56, "create VmCloneStorageMapping" for more information.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Tip</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the clone customizer.</td>
<td>Tip: To find this name after the clone customizer is created, use the list VmCloneCustomizer command. You need the name or ID of this to delete a clone customizer from a virtual machine with the delete VmCloneCustomizer command.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the clone customizer object. value is a maximum of 4,000 characters.</td>
<td>To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value</td>
<td>The instance of the object using either the id or name option, for example name=MyVM.</td>
</tr>
</tbody>
</table>

Note

Any create command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

Examples

Example A.70 Create a virtual machine clone customizer

OVM> create VmCloneCustomizer name=MyVmCloneCustomizer on Vm name=MyVM
### See Also

- Section A.56, “create VmCloneStorageMapping”
- Section A.55, “create VmCloneNetworkMapping”
- Section A.96, “edit VmCloneCustomizer”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

### A.55 create VmCloneNetworkMapping

Maps the network that should be used by a virtual machine clone customizer.

#### Syntax

```plaintext
create VmCloneNetworkMapping network=value vnic=value name=value [ description=value ] on vmCloneCustomizer instance
```

Where `instance` is:

```plaintext
{ id=value | name=value }
```

#### Description

Creates a new network mapping for use by a virtual machine clone customizer. To create a clone network mapping, a virtual machine with an assigned VNIC must already exist within your environment.

#### Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network=value</td>
<td>The name or ID value of an existing network to which the cloned virtual machine should be connected.</td>
</tr>
<tr>
<td>vnic=value</td>
<td>The name or ID value of an existing VNIC that is attached to the existing virtual machine that is this clone customizer's parent.</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the clone network mapping.</td>
</tr>
</tbody>
</table>

#### Tip

To find this name after the network is mapped to a virtual machine clone customizer, use the `list vmCloneNetworkMapping` command. You need the name or ID of this to delete a network mapping from a virtual machine with the `delete vmCloneNetworkMapping` command.
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>description=value</td>
<td>Optional description for the clone network mapping object. value is a maximum of 4,000 characters. To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

**Note**

Any create command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

### Examples

**Example A.71 Creating a virtual machine clone network mapping**

```
OVN> create VmCloneNetworkMapping name=MyCloneNet network=MyVMNetwork \  
vnic=0004fb0000070000277ecade9b897469 on VmCloneCustomizer name=MyVMCloneCustomizer
```

### See Also

- Section A.54, “create VmCloneCustomizer”
- Section A.97, “edit VmCloneNetworkMapping”
- Section A.56, “create VmCloneStorageMapping”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

### A.56 create VmCloneStorageMapping

Maps the storage for a virtual machine disk that should be used by a virtual machine clone customizer.

**Syntax**

```
create VmCloneStorageMapping cloneType=  { SPARSE_COPY | NON_SPARSE_COPY | THIN_CLONE } vmDiskMapping=value {physicalDisk=value | repository=value | storageArray=value} name=value [description=value] on vmCloneCustomizer instance
```

Where instance is:

```
{ id=value | name=value }
```

**Description**

Maps the storage for a virtual machine disk that should be used by a virtual machine clone customizer. A disk mapping that is already used by virtual machine that is this clone customizer’s parent must already exist.
Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloneType= { SPARSE_COPY</td>
<td>The type of clone storage that should be created.</td>
</tr>
<tr>
<td></td>
<td>NON_SPARSE_COPY</td>
</tr>
<tr>
<td>vmDiskMapping=value</td>
<td>The name or ID of an existing virtual machine disk mapping.</td>
</tr>
<tr>
<td>{physicalDisk=value</td>
<td>The name or ID of either a physicalDisk, repository or storageArray object</td>
</tr>
<tr>
<td></td>
<td>repository=value</td>
</tr>
<tr>
<td></td>
<td>storageArray=value }</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the clone storage mapping.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the clone storage mapping object. value is a</td>
</tr>
<tr>
<td></td>
<td>maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value }</td>
</tr>
<tr>
<td></td>
<td>name=MyVMCloneCustomizer.</td>
</tr>
</tbody>
</table>

Tip

To find this name after virtual storage is mapped to a virtual machine clone customizer, use the list vmCloneStorageMapping command. You need the name or ID of this to delete a disk mapping from a virtual machine clone customizer with the delete vmCloneStorageMapping command.

Note

Any create command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

Examples

**Example A.72 Creating a virtual machine clone storage mapping**

```
OVM> create VmCloneStorageMapping name=BootDisk cloneType=SPARSE_COPY \
    vmDiskMapping=0004fb000013000096e5d46c5f5e6a52 repository=MyRepository \
    on VmCloneCustomizer name=MyVMCloneCustomizer
```

See Also

- Section A.54, “create VmCloneCustomizer”
- Section A.98, “edit VmCloneStorageMapping”
create VmDiskMapping

- Section A.55, “create VmCloneNetworkMapping”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

### A.57 create VmDiskMapping

Maps a virtual disk, physical disk, or CDROM to a virtual machine disk slot.

**Syntax**

```plaintext
create VmDiskMapping slot=value{physicalDisk=value|virtualDisk=value|virtualCd={value|EMPTY_CDROM}} name=value[description=value] on Vm instance
```

Where `instance` is:

```plaintext
{id=value|name=value}
```

**Description**

This command maps a virtual disk, physical disk, or CDROM to a virtual machine disk slot. To create an empty CDROM drive, use the `virtualCd=EMPTY_CDROM` option. To edit a virtual disk or eject a CDROM, remove it using the `delete VmDiskMapping` command, then use the `create VmDiskMapping` command again to remap it to a virtual machine with any changed settings.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>slot=value</td>
<td>The slot number for the disk in the virtual machine. This can be an integer between 0 and 1000000.</td>
</tr>
<tr>
<td>physicalDisk=value</td>
<td>The name or ID of the physical disk.</td>
</tr>
<tr>
<td>VirtualDisk=value</td>
<td>The name or ID of the virtual disk.</td>
</tr>
<tr>
<td>virtualCd={value</td>
<td>EMPTY_CDROM}</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the disk mapping.</td>
</tr>
</tbody>
</table>

**Tip**

To find this name after a virtual disk is mapped to a virtual machine, use the `list VmDiskMapping` command. You will need the name or ID of this to delete a disk mapping from a virtual machine with the `delete VmDiskMapping` command. Optional description for the disk mapping object. `value` is a maximum of 4,000 characters.
Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
<tr>
<td></td>
<td>The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyVM</code>.</td>
</tr>
</tbody>
</table>

Note

Any `create` command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

Examples

Example A.73 Mapping a virtual disk to a virtual machine

```bash
OVM> create VmDiskMapping slot=0 virtualDisk=MyVMDisk name="Boot Disk" on Vm name=MyVM
```

Example A.74 Mapping an ISO file (CDROM) to a virtual machine

```bash
OVM> create VmDiskMapping slot=1 virtualCd=OracleLinux-dvd.iso 
    name="CDROM Drive" on Vm name=MyVM
```

Example A.75 Mapping an empty CDROM drive to a virtual machine

```bash
OVM> create VmDiskMapping slot=2 virtualCd=EMPTY_CDROM name="CDROM Drive" on Vm name=MyVM
```

Example A.76 Mapping a physical disk to a virtual machine

```bash
OVM> create VmDiskMapping slot=3 physicalDisk=MyPhysicalDisk name="D Drive" on Vm name=MyVM
```

See Also

- Section A.99, “edit VmDiskMapping”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

A.58 create Vnic

Creates a VNIC on a virtual machine.

Syntax

```bash
create Vnic network=value name=value [macAddress=value] [description=value] on Vm instance
```

Where `instance` is:

`{id=value | name=value}`

Description

This command creates a VNIC and associates it to a virtual machine.
Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network=value</td>
<td>The name or ID of the network to which the VNIC is to be assigned.</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the VNIC.</td>
</tr>
<tr>
<td>macAddress=value</td>
<td>Optional parameter to specify the MAC address for the VNIC. If no MAC address is provided, one is automatically assigned from the default range. To change the default range of MAC addresses, use the setVnicMacAddrRange command. To display the MAC address range, use the getVnicMacAddrRange command.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the VNIC. value is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use description=&quot;&quot;.</td>
</tr>
</tbody>
</table>

Note

Any create command only creates a single instance of an object, and therefore only accepts a single object instance as an attribute. Providing more than one object of the same attribute type as a parameter always results in the last attribute value taking precedence.

Examples

Example A.77 Creating a VNIC

```
OVMe> create Vnic name=MyVNIC network=MyVMNetwork on Vm name=MyVM
```

See Also

- Section A.100, “edit Vnic”
- Section A.17, “add Vnic”
- Section A.160, “remove Vnic”
- Section A.170, “setVnicMacAddrRange”
- Section A.129, “getVnicMacAddrRange”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

A.59 delete

Deletes an object.
Syntax

```
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command deletes an object.

Virtual machines and virtual machine templates are treated as equivalent within the CLI. Therefore, to delete a virtual machine template you should use the `delete Vm` command. Since it is possible that a virtual machine template and a virtual machine may share the same name, it is recommended that you use the object's unique ID to perform this operation.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{ AccessGroup</td>
<td>AntiAffinityGroup</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Important

The `Assembly` option has been deprecated. You should instead use `VirtualAppliance`.

Examples

Example A.78 Deleting an Oracle VM Server

```
OVM> delete Server name=MyServer
```
Example A.79 Deleting a virtual machine or virtual machine template

OVM> delete Vm id=0004fb00001400007be890778aedc7b8

Example A.80 Deleting a network

OVM> delete Network name=MyVMNetwork

Example A.81 Deleting a virtual machine disk mapping

OVM> delete VmDiskMapping id=0004fb00001300009d46acbb77de919e

See Also

• Section A.138, “list”
• Section A.171, “show”

A.60 discoverServer

Discovers an Oracle VM Server.

Syntax

discoverServer ipAddress=value password=value takeOwnership={ Yes | No }

Description

This command discovers an Oracle VM Server and adds it to Oracle VM Manager. The username used to connect to the Oracle VM Agent on the Oracle VM Server is oracle. The port number on which access is made to the Oracle VM Agent on the Oracle VM Server is 8899.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipAddress=value</td>
<td>The IP address or hostname of the Oracle VM Server.</td>
</tr>
<tr>
<td>password=value</td>
<td>The password to use when connecting to the Oracle VM Agent on the Oracle VM Server.</td>
</tr>
<tr>
<td>takeOwnership={ Yes</td>
<td>No }</td>
</tr>
</tbody>
</table>

Important

The takeOwnership option should only be used if the Oracle VM Server is not already owned by an existing Oracle VM Manager. If specified for an Oracle VM Server that is already owned, the option is silently ignored.

Examples

Example A.82 Discovering an Oracle VM Server

OVM> discoverServer ipAddress=10.172.76.73 password=password takeOwnership=Yes
See Also

- Section A.80, “edit Server”
- Section A.11, “add Server”
- Section A.154, “remove Server”
- Section A.177, “start”
- Section A.162, “restart”
- Section A.178, “stop”
- Section A.137, “kill”
- Section A.181, “upgrade”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

A.61 edit AccessGroup

Edits an access group.

Syntax

```
edit AccessGroup instance [ name=value ][ nameOnArray=value ][ description=value ]
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command edits the attributes of an access group. Generic Oracle VM Storage Connect plug-ins are not supported with this command.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name for the access group.</td>
</tr>
<tr>
<td>nameOnArray=value</td>
<td>A name for the access group as known on the storage array.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the access group. value is a maximum of 4,000 characters.</td>
</tr>
</tbody>
</table>
Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

**Example A.83 Changing the name of an access group**

```
OVM> edit AccessGroup name=MyAccessGroup nameOnArray=MyNewName
```

See Also

- Section A.34, “create AccessGroup”
- Section A.9, “add PhysicalDisk”
- Section A.152, “remove PhysicalDisk”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

**A.62 edit AntiAffinityGroup**

Edits an anti affinity group in a server pool.

Syntax

```
edit AntiAffinityGroup instance [name=value][description=value]
```

Where `instance` is:

{id=value | name=value}

Description

This command edits the attributes of an anti affinity group in a server pool. To add a virtual machine to an anti affinity group, use the add Vm command. To remove a virtual machine from an anti affinity group, use the remove Vm command.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the anti affinity group.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the anti affinity group. value is a maximum of 4,000 characters.</td>
</tr>
</tbody>
</table>


Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
<tr>
<td>name=value</td>
<td>The instance of the object using either the id or name option, for example name=MyAAGroup.</td>
</tr>
</tbody>
</table>

Example A.84 Editing an anti affinity group

```
OVN> edit AntiAffinityGroup name=MyAAGroup description="My Anti Affinity Group"
```

See Also

- Section A.35, “create AntiAffinityGroup”
- Section A.16, “add Vm”
- Section A.159, “remove Vm”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

A.63 edit Assembly (Deprecated)

Edits an assembly.

<table>
<thead>
<tr>
<th>Important</th>
</tr>
</thead>
<tbody>
<tr>
<td>This command has been deprecated. Instead, use the edit VirtualAppliance command.</td>
</tr>
</tbody>
</table>

Syntax

```
edit Assembly instance[name=value][description=value]
```

Where instance is:

```
{id=value|name=value}
```

Description

This command edits the attributes of assembly.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the assembly.</td>
</tr>
</tbody>
</table>
### Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>description=value</td>
<td>Optional description for the assembly. <em>value</em> is a maximum of 4,000 characters. To set an empty description, use <em>description=&quot;&quot;</em>.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

#### Example A.85 Editing an assembly

```
OVM> edit Assembly name=myassembly.ova description='Oracle Linux Release 6'
```

#### See Also

- Section A.64, “edit AssemblyVirtualDisk (Deprecated)”
- Section A.65, “edit AssemblyVm (Deprecated)”
- Section A.131, “importAssembly (Deprecated)”
- Section A.111, “getDescriptor”
- Section A.59, “delete”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”

### A.64 edit AssemblyVirtualDisk (Deprecated)

Edits a virtual disk in an assembly.

**Important**

This command has been deprecated. Instead, use the **edit VirtualApplianceVirtualDisk** command.

#### Syntax

```
edit AssemblyVirtualDisk instance [name=value][description=value]
```

Where *instance* is:

```
{ id=value | name=value }
```

#### Description

This command edits the attributes of a virtual disk in an assembly. You cannot delete an AssemblyVirtualDisk object. You can only delete it by deleting the assembly itself.
Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>name=value</code></td>
<td>A name to identify the assembly virtual disk.</td>
</tr>
<tr>
<td><code>description=value</code></td>
<td>Optional description for the assembly virtual disk. <code>value</code> is a maximum of 4,000 characters. To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

Examples

Example A.86 Editing an assembly virtual disk

```
OVM> edit AssemblyVirtualDisk name=MyAssemblyVirtualDisk description='Oracle Linux Release 6'
```

See Also

- Section A.63, “edit Assembly (Deprecated)”
- Section A.65, “edit AssemblyVm (Deprecated)”
- Section A.131, “importAssembly (Deprecated)”
- Section A.111, “getDescriptor”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”

A.65 edit AssemblyVm (Deprecated)

Edits a virtual machine in an assembly.

⚠️ Important

This command has been deprecated. Instead, use the `edit VirtualApplianceVm` command.

Syntax

```
edit AssemblyVm instance [name=value] [description=value]
```

Where `instance` is:

```
{id=value | name=value}
```
**Description**

This command edits the attributes of a virtual machine in an assembly. You cannot delete an AssemblyVm object. You can only delete it by deleting the assembly itself.

Although none of the options are mandatory, you must supply at least one option.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the assembly virtual machine.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the assembly virtual machine. value is a maximum of 4,000 characters. To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.87 Editing an assembly virtual machine**

```
OVM> edit AssemblyVm name=MyAssemblyVM description='Oracle Linux Release 6'
```

**See Also**

- Section A.63, “edit Assembly (Deprecated)”
- Section A.64, “edit AssemblyVirtualDisk (Deprecated)”
- Section A.131, “importAssembly (Deprecated)”
- Section A.111, “getDescriptor”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”

**A.66 edit BondPort**

Edits a bond port.

**Syntax**

```
edit BondPort instance [mode= {ACTIVE_PASSIVE | LINK_AGGREGATION | LOAD_BALANCED }][mtu=value][name=value][description=value]
```

Where instance is:
{id=value|name=value}

**Description**

This command edits the attributes of a bond port on an Oracle VM Server.

Although none of the options are mandatory, you must supply at least one option.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode= {ACTIVE_PASSIVE</td>
<td>The network bonding mode.</td>
</tr>
<tr>
<td>LINK_AGGREGATION</td>
<td>}</td>
</tr>
<tr>
<td>mtu=value</td>
<td>The MTU value. May be an integer between 1500 and 64000.</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the bond.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the bond. value is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.88 Editing a bond port**

```bash
OVM> edit BondPort id=0004fb0000200000884da42c23947622 mode=LOAD_BALANCED
```

**See Also**

- Section A.36, “create BondPort”
- Section A.7, “add BondPort”
- Section A.150, “remove BondPort”
- Section A.153, “remove Port”
- Section A.102, “embeddedCreate”
- Section A.103, “embeddedDelete”
- Section A.104, “embeddedEdit”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”
A.67 edit ControlDomain

Edits a control domain.

Syntax

```
edit ControlDomain instance [name=value][description=value]
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command edits the attributes of a control domain. You cannot create or delete a control domain; you can only edit the name and description.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the control domain.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the control domain. <code>value</code> is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.89 Editing a control domain

```
OVM> edit ControlDomain id=0004fb0000210000308f567ae2cbdc4c description="SPARC control domain"
```

See Also

- Section A.138, “list”
- Section A.171, “show”

A.68 edit Cpu

Edits a control domain.

Syntax

```
edit Cpu instance [name=value][description=value]
```
Where *instance* is:

{ id=value | name=value }

**Description**

This command edits the attributes of a CPU. You cannot create or delete a CPU; you can only edit the name and description.

Although none of the options are mandatory, you must supply at least one option.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the CPU.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the CPU. <em>value</em> is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use <em>description</em>=&quot;&quot;.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.90 Editing a CPU**

```
OVM> edit Cpu id="Processor (1) in 00:e0:81:4d:40:c6:00:e0:81:4d:40:c7:ff:ff:ff:ff" \
  description="CPU 1 on MyServer1"
```

**See Also**

- Section A.138, “list”
- Section A.171, “show”

**A.69 edit CpuCompatibilityGroup**

Edits a *CPU compatibility group*.

**Syntax**

```
edit CpuCompatibilityGroup instance [name=value][description=value]
```

Where *instance* is:

{ id=value | name=value }

**Description**

This command edits the attributes of a CPU compatibility group.
Although none of the options are mandatory, you must supply at least one option.

## Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=</td>
<td>A name to identify the CPU compatibility group.</td>
</tr>
<tr>
<td>description=</td>
<td>Optional description for the CPU compatibility group. value is</td>
</tr>
<tr>
<td></td>
<td>a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

## Examples

### Example A.91 Editing a CPU compatibility group

```
OVM> edit CpuCompatibilityGroup name=MyCPUGroup description='SPARC-based CPU compatibility group'
```

## See Also

- Section A.11, “add Server”
- Section A.154, “remove Server”
- Section A.37, “create CpuCompatibilityGroup”
- Section A.138, “list”
- Section A.171, “show”
- Section A.59, “delete”

### A.70 edit FileServer

Edits a file server.

#### Syntax

```
edit FileServer instance[plugin=value][accessHost=value][uniformedExports= {Yes | No}][name=value][description=value]
```

Where instance is:

```
{id=value | name=value}
```

#### Description

This command edits the attributes of a file server.

Although none of the options are mandatory, you must supply at least one option.
Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plugin=value</td>
<td>The name or ID for the Oracle VM Storage Connect plug-in for the file server. To see a list of the available plug-ins, use the list StorageArrayPlugin command.</td>
</tr>
<tr>
<td>accessHost=value</td>
<td>The host name or IP address for the file server.</td>
</tr>
<tr>
<td>uniformedExports= {Yes</td>
<td>No}</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the file server.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the file server. value is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.92 Editing a file server

```
OVM> edit FileServer id=0004fb00000900000ef55b2f96a564c8 name=MyNFSServer \
  description='My NFS Server'
```

See Also

- Section A.38, “create FileServer”
- Section A.11, “add Server”
- Section A.154, “remove Server”
- Section A.4, “addAdminServer”
- Section A.147, “removeAdminServer”
- Section A.6, “addRefreshServer”
- Section A.149, “removeRefreshServer”
- Section A.142, “refresh”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

A.71 edit FileServerPlugin
Edits a file server plug-in.

**Syntax**

```
edit FileServerPlugin instance [name=value] [description=value]
```

Where **instance** is:

```
{ id=value | name=value }
```

**Description**

This command edits the attributes of a file server plug-in. You cannot create or delete a file server plug-in; you can only edit the name and description.

Although none of the options are mandatory, you must supply at least one option.

**Options**

The following table shows the available options for this command:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the file server plug-in.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the file server plug-in. value is a maximum of 4,000 characters. To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.93 Editing a file server plug-in**

```
OVM> edit FileServerPlugin id="oracle.generic.NFSPlugin.GenericNFSPlugin (1.1.0)" \
    description="File Server Plugin for NFS Server"
```

**See Also**

- Section A.138, “list”
- Section A.171, “show”

**A.72 edit FileSystem**

Edits an OCFS2 file system.

**Syntax**

```
edit FileSystem instance [name=value] [description=value]
```


Where *instance* is:

{id=value | name=value}

**Description**

This command edits the attributes of an OCFS2 file system.

Although none of the options are mandatory, you must supply at least one option.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the file system.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the file system. <em>value</em> is a maximum of 4,000 characters. To set an empty description, use <em>description=&quot;&quot;</em>.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.94 Editing a file system**

```
OVM> edit FileSystem id=0004fb0000050000002618dec56ee0e8 name=MyFileSystem 
    description='My File System'
```

**See Also**

- Section A.39, “create FileSystem”
- Section A.8, “add FileSystem”
- Section A.151, “remove FileSystem”
- Section A.34, “create AccessGroup”
- Section A.61, “edit AccessGroup”
- Section A.142, “refresh”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

**A.73 edit Manager**
Edits a Manager object.

**Syntax**

```
edit Manager instance name=value [description=value]
```

Where `instance` is:

```
{ id=value | name=value }
```

**Description**

This command edits the attributes of a Manager (Oracle VM Manager) object.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the Manager object.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the Manager object. <code>value</code> is a maximum of 4,000 characters. To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.95 Renaming a Manager object**

```
OVM> edit Manager name="OVM Manager" name="Oracle VM Manager"
```

**See Also**

- Section A.138, “list”
- Section A.171, “show”

**A.74 edit Network**

Edits an Ethernet-based network.

**Syntax**

```
edit Network instance [roles= { MANAGEMENT | LIVE_MIGRATE | CLUSTER_HEARTBEAT | VIRTUAL_MACHINE | STORAGE }] [name=value] [description=value]
```

Where `instance` is:

```
{ id=value | name=value }
```
Description

This command edits the attributes of an Ethernet-based network.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>roles= { MANAGEMENT</td>
<td>The network roles. Enter options separated by commas (,), for example:</td>
</tr>
<tr>
<td>LIVE_MIGRATE</td>
<td>roles='VIRTUAL_MACHINE, STORAGE'</td>
</tr>
<tr>
<td>CLUSTER_HEARTBEAT</td>
<td>To set an empty roles option, use roles=''.</td>
</tr>
<tr>
<td>VIRTUAL_MACHINE</td>
<td>It is not possible to edit the roles for a local network. Any attempts to do so are simply ignored.</td>
</tr>
<tr>
<td>STORAGE</td>
<td></td>
</tr>
</tbody>
</table>

| name=value             | A name to identify the network.                                           |
| description=value      | Optional description for the network. value is a maximum of 4,000 characters. |
|                        | To set an empty description, use description=''.                         |

| { id=value | name=value } | The instance of the object using either the id or name option, for example name=MyNetwork. |

Examples

Example A.96 Editing a network

```
OVM> edit Network name=MyVMNetwork roles='VIRTUAL_MACHINE,LIVE_MIGRATION'
```

See Also

- Section A.10, “add Port”
- Section A.77, “edit Port”
- Section A.153, “remove Port”
- Section A.40, “create Network”
- Section A.36, “create BondPort”
- Section A.138, “list”
- Section A.171, “show”
A.75 edit PeriodicTask

Edits a periodic task.

Syntax

```
edit PeriodicTask instance [taskInterval=value][ enabled={Yes | No}][ name=value][ description=value]
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command edits the attributes of a periodic task. A periodic task is a task (job) that can be run multiple times, with an interval in between, such as checking server updates repositories for available Oracle VM Server software updates.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>taskInterval=value</code></td>
<td>The interval between which the periodic task should run, in minutes. <code>value</code> may be an integer between 1 and 999999999.</td>
</tr>
<tr>
<td>`enabled={Yes</td>
<td>No} `</td>
</tr>
<tr>
<td><code>name=value</code></td>
<td>A name to identify the periodic task.</td>
</tr>
<tr>
<td><code>description=value</code></td>
<td>Optional description for the periodic task. <code>value</code> is a maximum of 4,000 characters. To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Examples

**Example A.97 Disabling a periodic task**

```
OVM> edit PeriodicTask name="Server Upgrade Checker Task" enabled=No
```

See Also

- Section A.138, “list”
- Section A.171, “show”

A.76 edit PhysicalDisk
Edits a physical disk.

**Syntax**

```
edit PhysicalDisk instance shareable= {Yes|No} [extraInfo=value] name=value [description=value]
```

Where *instance* is:

```
{id=value|name=value}
```

**Description**

This command edits the attributes of a physical disk. *Local storage* and generic storage plug-ins are not supported with this command. To resize a physical disk, use the `resize` command.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>shareable= {Yes</td>
<td>No}</td>
</tr>
<tr>
<td>extraInfo=value</td>
<td>Any extra parameters for your Oracle VM Storage Connect plug-in.</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the physical disk.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the physical disk. <em>value</em> is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.98 Editing a physical disk**

```
OVM> edit PhysicalDisk id=0004fb000018000034a2da375d08990e shareable=Yes
```

**See Also**

- Section A.41, “create PhysicalDisk”
- Section A.9, “add PhysicalDisk”
- Section A.152, “remove PhysicalDisk”
- Section A.161, “resize”
- Section A.59, “delete”
A.77 edit Port

Edits an Ethernet port on an Oracle VM Server.

Syntax

```
edit Port instance[ mtu=value][ name=value][ description=value]
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command edits the attributes of an Ethernet port on an Oracle VM Server.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mtu=value</td>
<td>The MTU value. May be an integer between 1500 and 64000.</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the port.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the port. value is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

**Example A.99 Editing an Ethernet port**

```
OVM> edit Port id=0004fb0000200000b0f9d86788b94a0e mtu=1500
```

See Also

- Section A.10, "add Port"
- Section A.36, "create BondPort"
- Section A.7, "add BondPort"
A.78 edit Repository

Edits a storage repository.

Syntax

```shell
edit Repository instance [name=value] [description=value]
```

Where `instance` is:

```shell
{id=value | name=value}
```

**Description**

This command edits the attributes of a storage repository.

Although none of the options are mandatory, you must supply at least one option.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>name=value</code></td>
<td>A name to identify the storage repository.</td>
</tr>
<tr>
<td><code>description=value</code></td>
<td>Optional description for the storage repository. <code>value</code> is a maximum of 4,000 characters. To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

**Examples**

Example A.100 Editing a storage repository

```shell
OVM> edit Repository id=0004fb00000300003ab65ab35e3fea7a name=MyRepository \
    description="My Storage Repository"
```
See Also

- Section A.42, “create Repository”
- Section A.11, “add Server”
- Section A.154, “remove Server”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

A.79 edit RepositoryExport

Edits a repository export.

Syntax

```
edit RepositoryExport instance name=value [description=value] [options=value]
```

Where `instance` is:

{ `id=value` | name=value }

Description

This command edits the attributes of an export on an Oracle VM Server to enable access for a third party back up tool to back up the contents of an OCFS2-based storage repository.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the export on the file server.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the export on the file server. <code>value</code> is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>options=value</td>
<td>The parameters to include in the NFS mount configuration, for example: <code>rw, async, no_root_squash, wdelay</code>.</td>
</tr>
<tr>
<td>{ <code>id=value</code></td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

Example A.101 Editing a repository export on an Oracle VM Server

```
OVM> edit RepositoryExport name="My NFS Export" name=MyExport \  
```
A.80 edit Server

Edits an Oracle VM Server.

Syntax

```
edit Server instance [ntpServers=value][roles={VM|UTILITY}][inboundMigrationLocked={Yes|No}][name=value][description=value]
```

Where `instance` is:

```
{id=value|name=value}
```

Description

This command edits the attributes of an Oracle VM Server. Use this command set the NTP servers, role, and whether to allow running of virtual machines.

To place an Oracle VM Server into or out of maintenance mode, use the `setMaintenanceMode` command.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ntpServers=value</code></td>
<td>The hostname or IP address of one or more NTP servers to set the time on the Oracle VM Server. Enter options separated by commas (,), for example: <code>ntpServers=&quot;ntp1.example.com,ntp2.example.com&quot;</code> To set an empty <code>ntpServers</code> option, use <code>ntpServers=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>`roles={VM</td>
<td>UTILITY}`</td>
</tr>
<tr>
<td></td>
<td>To set an empty <code>roles</code> option, use <code>roles=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>`inboundMigrationLocked={Yes</td>
<td>No}`</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>not prevent virtual machines running on the Oracle VM Server from being migrated to another.</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the Oracle VM Server.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the Oracle VM Server. <code>value</code> is a maximum of 4,000 characters. To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td></td>
<td>The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code>.</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.102 Adding NTP servers to an Oracle VM Server**

```
OVM> edit Server name=MyServer ntpServer="ntp1.example.com,ntp2.example.com"
```

**Example A.103 Changing the role of an Oracle VM Server**

```
OVM> edit Server name=MyServer role=UTILITY runVmsEnabled=No
```

**See Also**

- Section A.60, “discoverServer”
- Section A.11, “add Server”
- Section A.154, “remove Server”
- Section A.177, “start”
- Section A.162, “restart”
- Section A.178, “stop”
- Section A.137, “kill”
- Section A.168, “setMaintenanceMode”
- Section A.181, “upgrade”
- Section A.4, “addAdminServer”
- Section A.147, “removeAdminServer”
- Section A.6, “addRefreshServer”
- Section A.149, “removeRefreshServer”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”
**A.81 edit ServerController**

Edits a server controller object to configure IPMI on an Oracle VM Server.

**Syntax**

```
edit ServerController instance ipAddress=value userName=value [password=value] name=value [description=value]
```

Where `instance` is:

```
{ id=value | name=value }
```

**Description**

This command edits the attributes of a server controller object to configure IPMI on an Oracle VM Server.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>virtualIP=</code></td>
<td>The IP address of the IPMI.</td>
</tr>
<tr>
<td><code>userName=</code></td>
<td>The user name for the IPMI.</td>
</tr>
<tr>
<td><code>password=</code></td>
<td>An optional password for the IPMI.</td>
</tr>
<tr>
<td><code>name=</code></td>
<td>A name to identify the server control object.</td>
</tr>
<tr>
<td><code>description=</code></td>
<td>Optional description for the server control object. <code>value</code> is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.104 Editing IPMI configuration for an Oracle VM Server**

```
OVM> edit ServerController id=mgtCtl_00:e0:81:4d:40:c6:00:e0:81:4d:40:c7:ff:ff:ff:ff 
    ipAddress=192.168.10.4 userName=admin password=password name=MyServerController
```

**See Also**

- Section A.44, “create ServerController”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

**A.82 edit ServerPool**
**Syntax**

```bash
edit ServerPool instance clusterEnable= {Yes|No}[ clusterTimeout= value][
  keymapName={en-us | ar | da | de | de-ch | en-gb | es | et | fi | fo | fr | fr-be | fr-ca | fr-ch | hr | hu | is | it | ja | lt | lv | mk | nl | nl-be | No | pl | pt | pt-br | ru | sl | sv | th | tr}][
  migrateUsingSsl={Yes | No}][
  masterServer=value][
  startPolicy= {BEST_SERVER | BALANCE_SERVER | CURRENT_SERVER}][
  policyMode= {OFF | DRS | DPM}][
  policyCpuEnable= {Yes | No}][
  policyPeriod=value][
  policyCpuThreshold=value][
  name=value][
  description=value]
```

Where `instance` is:
```
{ id=value | name=value }
```

**Description**

This command edits the attributes of a server pool. Use this command to change the master Oracle VM Server, change the keyboard mapping, set whether to use secure migration of virtual machines, and to manage server pool policies.

Although none of the options are mandatory, you must supply at least one option.

It is not possible to edit the cluster enable flag for a server pool if there are Oracle VM Servers in the server pool. Attempts to edit this attribute for a server pool that already contain Oracle VM Servers fail and result in an error.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`clusterEnable= {Yes</td>
<td>No}`</td>
</tr>
<tr>
<td><code>clusterTimeout= value</code></td>
<td>Set the timeout value for the cluster in seconds. <code>value</code> is an integer between 30 and 300. Note that you can only change this value for a server pool if the pool contains no servers. SPARC-based server pools do not recognise the cluster timeout parameter. Setting this value for a server pool consisting of SPARC servers has no effect and the parameter is ignored by the Oracle VM Agent for SPARC.</td>
</tr>
<tr>
<td>`keymapName={en-us</td>
<td>ar</td>
</tr>
<tr>
<td>`migrateUsingSsl= {Yes</td>
<td>No}`</td>
</tr>
</tbody>
</table>
### Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>masterServer=value</td>
<td>The name or ID of the Oracle VM Server to use as the master server for the server pool.</td>
</tr>
<tr>
<td>startPolicy= { BEST_SERVER</td>
<td>BALANCE_SERVER</td>
</tr>
<tr>
<td>policyMode= { OFF</td>
<td>DRS</td>
</tr>
<tr>
<td>policyCpuEnable= { Yes</td>
<td>No }</td>
</tr>
<tr>
<td>policyPeriod=value</td>
<td>The time period for the policy job to run. This sets the policy job to run every ( n ) minutes, for example, 10 sets the policy job to run every 10 minutes. ( value ) can be an integer between 2 and 60.</td>
</tr>
<tr>
<td>policyCpuThreshold=value</td>
<td>The maximum amount of CPU percentage usage allowed before the policy must be enacted. ( value ) can be an integer between 25 and 99.</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the server pool.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the server pool. ( value ) is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

#### Examples

**Example A.105 Editing a server pool**

```
OVM> edit ServerPool name=MyServerPool name=MyOtherServerPool migrateUsingSsl=Yes
```

**Example A.106 Changing the master Oracle VM Server**

```
OVM> edit ServerPool id=0004fb000002000037db5e362c85a3fe masterServer=MyServer
```

**Example A.107 Changing the virtual machine start policy**

```
OVM> edit ServerPool name=MyServerPool startPolicy=BEST_SERVER
```

#### See Also

- Section A.45, “create ServerPool”
- Section A.46, “create ServerPoolNetworkPolicy”
- Section A.83, “edit ServerPoolNetworkPolicy”
- Section A.11, “add Server”
- Section A.154, “remove Server”
- Section A.59, “delete”
- Section A.138, “list”
A.83 edit ServerPoolNetworkPolicy

Edits a server pool network policy.

Syntax

edit ServerPoolNetworkPolicy instance [policyEnable= {Yes|No}][policyThreshold=value] name=value [description=value]

Where instance is:

{ id=value | name=value }

Description

This command edits the attributes of a server pool network policy.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>policyEnable= {Yes</td>
<td>No}</td>
</tr>
<tr>
<td>policyThreshold=value</td>
<td>The percentage (%) of network bandwidth the policy uses to move virtual machines. value can be an integer between 0 and 100.</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the server pool network policy.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the server pool network policy. value is a maximum of 4,000 characters. To set an empty description, use description=&quot;&quot;.</td>
</tr>
</tbody>
</table>

Examples

Example A.108 Editing a server pool network policy

OVM> edit ServerPoolNetworkPolicy name=MyNetworkPolicy policyEnable=No name="My Network Policy"

See Also

- Section A.46, “create ServerPoolNetworkPolicy”
- Section A.5, “addPolicyServer”
- Section A.148, “removePolicyServer”
- Section A.45, “create ServerPool”
A.84 edit ServerUpdateGroup

Edits an Oracle VM Server update group in a server pool.

Syntax

```
edit ServerUpdateGroup instance[name=value][description=value]
```

Where instance is:

```
{id=value | name=value}
```

Description

This command edits an Oracle VM Server update group in a server pool.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the server update group.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the server update group. value is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.109 Editing the name of an update group for a server pool

```
OVM> edit ServerUpdateGroup name=MyServerUpdateGroup name="My Server Update Group"
```

See Also

- Section A.47, “create ServerUpdateGroup”
- Section A.48, “create ServerUpdateRepository”
- Section A.85, “edit ServerUpdateRepository”
- Section A.19, “checkUpToDate”
A.85 edit ServerUpdateRepository

Edits an Oracle VM Server update repository.

Syntax

```
edit ServerUpdateRepository [repositoryName=value][ url=value][enabled= {Yes|No}][pkgSignatureType= {NONE|GPG|CA}][pkgSignatureKey=value][name=value][description=value]
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command edits an Oracle VM Server update repository.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>repositoryName=value</td>
<td>A name for the server update repository. This may only contain alphanumeric characters and underscores. No spaces are permitted.</td>
</tr>
<tr>
<td>url=value</td>
<td>The URL to access the repository.</td>
</tr>
<tr>
<td>enabled= {Yes</td>
<td>No}</td>
</tr>
<tr>
<td>pkgSignatureType= {NONE</td>
<td>GPG</td>
</tr>
<tr>
<td>pkgSignatureKey=value</td>
<td>The verification signature for the repository, for example, the location of the GPG key using any of the HTTP, FTP, FILE or HTTPS protocols.</td>
</tr>
</tbody>
</table>

If you enclose the value for the option in quotes, you must escape each forward slash (/) with another, for example:

```
url="http:///10.172.77.200//ovs"
```
### Examples

#### Example A.110 Editing an Oracle VM Server update repository

```
OVM> edit ServerUpdateRepository url=http://10.172.77.200/ovs enabled=No pkgSignatureType=GPG \
    pkgSignatureKey=http://10.172.77.200/ovs/RPM-GPG-KEY-oracle
```

### See Also

- Section A.48, “create ServerUpdateRepository”
- Section A.47, “create ServerUpdateGroup”
- Section A.84, “edit ServerUpdateGroup”
- Section A.19, “checkUpToDate”
- Section A.181, “upgrade”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

### A.86 edit StorageArray

Edits a storage array.

#### Syntax

```
edit StorageArray instance[ plugin=value][ storageName=value][ useChap={Yes | No}][ adminHost=value adminUserName=value adminPassword=value][ name=value][ description=value][ lipScan={Yes | No}]
```

Where `instance` is:

```
{id=value|name=value}
```

#### Description

This command edits the attributes of a storage array.
Although none of the options are mandatory, you must supply at least one option.

## Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plugin=value</td>
<td>The name or ID for the Oracle VM Storage Connect plug-in for the storage array. To see a list of the available plug-ins, use the list StorageArrayPlugin command.</td>
</tr>
<tr>
<td>storageName=value</td>
<td>The storage name associated with the storage array. The storage name uniquely identifies the correct storage array if multiple storage devices are concentrated behind a single endpoint.</td>
</tr>
<tr>
<td>useChap={Yes</td>
<td>No}</td>
</tr>
<tr>
<td>adminHost=value</td>
<td>The host name or IP address where administrative access to the storage array is allowed.</td>
</tr>
<tr>
<td>adminUserName=value</td>
<td>A user name with administrative access to the storage array, used with adminHost.</td>
</tr>
<tr>
<td>adminPassword=value</td>
<td>The administrator password for the adminUserName user.</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the storage array.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the storage array. value is a maximum of 4,000 characters. To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
<tr>
<td>lipScan={Yes</td>
<td>No}</td>
</tr>
</tbody>
</table>

### Note

This option is only applicable to an Unmanaged FibreChannel Storage Array.

## Examples

### Example A.111 Editing the name of a storage array

```bash
OVM> edit StorageArray name=MyISCSIServer name="My ISCI Server"
```

## See Also

- Section A.49, “create StorageArray”
- Section A.11, “add Server”
- Section A.154, “remove Server”
A.87 edit StorageArrayPlugin

Edits a storage array plug-in.

Syntax

```
edit StorageArrayPlugin instance [name=value][description=value]
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command edits the attributes of a storage array plug-in. You cannot create or delete a storage array plug-in; you can only edit the name and description.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the storage array plug-in.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the storage array plug-in. <code>value</code> is a maximum of 4,000 characters. To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.112 Editing a storage array plug-in

```
OVM> edit StorageArrayPlugin name="oracle.generic.SCSIPlugin.GenericPlugin (1.1.0)" \
    description="Generic storage array plug-in"
```
A.88 edit Tag
Edits a tag.

Syntax

```
edit Tag instance name=value [ description=value ]
```

Where `instance` is:

```
{ id=value | name=value }
```

Description
This command edits the attributes of a tag used to identify and group objects.

Options
The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the tag.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the tag. <code>value</code> is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

Example A.113 Editing a tag

```
OVM> edit Tag name=MyTag name=MyNewTagName description="My new tag name."
```

See Also

- Section A.50, “create Tag”
- Section A.14, “add Tag”
- Section A.157, “remove Tag”
- Section A.59, “delete”
- Section A.138, “list”
A.89 edit VirtualAppliance

Edits a virtual appliance.

Syntax

```
edit VirtualAppliance instance [ name=value ] [ description=value ]
```

Where `instance` is:

```
{id=value} | {name=value}
```

Description

This command edits the attributes of a VirtualAppliance object.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the virtual appliance.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the virtual appliance. <code>value</code> is a maximum of 4,000 characters. To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>`{id=value}</td>
<td>{name=value}`</td>
</tr>
</tbody>
</table>

Examples

Example A.114 Editing a virtual appliance

```
OVM> edit VirtualAppliance name=virtualappliance.ova description='Oracle Linux Release 6'
```

See Also

- Section A.90, “edit VirtualApplianceVirtualDisk”
- Section A.91, “edit VirtualApplianceVm”
- Section A.32, “createVmFromVirtualAppliance”
- Section A.33, “createVmFromVirtualApplianceVm”
- Section A.133, “importVirtualAppliance”
- Section A.106, “exportVirtualAppliance”
- Section A.111, “getDescriptor”
• Section A.59, “delete”
• Section A.142, “refresh”
• Section A.138, “list”
• Section A.171, “show”

A.90 edit VirtualApplianceVirtualDisk

Edits a virtual disk in a virtual appliance.

Syntax

```
edit VirtualApplianceVirtualDisk instance [name=value] [description=value]
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command edits the attributes of a virtual disk in a virtual appliance. You cannot delete a VirtualApplianceVirtualDisk object. You can only delete it by deleting the virtual appliance itself.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>name=value</code></td>
<td>A name to identify the virtual appliance virtual disk.</td>
</tr>
<tr>
<td><code>description=value</code></td>
<td>Optional description for the virtual appliance virtual disk.</td>
</tr>
<tr>
<td>`id=value</td>
<td>name=value`</td>
</tr>
<tr>
<td><code>value</code></td>
<td><code>value</code> is a maximum of 4,000 characters.</td>
</tr>
</tbody>
</table>

Examples

Example A.115 Editing a virtual appliance virtual disk

```
OVM> edit VirtualApplianceVirtualDisk name=MyVirtualApplianceVirtualDisk description='Oracle Linux Release 6'
```

See Also

• Section A.89, “edit VirtualAppliance”
• Section A.91, “edit VirtualApplianceVm”
• Section A.32, “createVmFromVirtualAppliance”
A.91 edit VirtualApplianceVm

Edits a virtual machine in a virtual appliance.

Syntax

edit VirtualApplianceVm instance[ name=value ][ description=value ]

Where instance is:

{ id=value | name=value }

Description

This command edits the attributes of a virtual machine in a virtual appliance. You cannot delete a VirtualApplianceVm object. You can only delete it by deleting the virtual appliance itself.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the virtual appliance virtual machine.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the virtual appliance virtual machine. value is a maximum of 4,000 characters. To set an empty description, use description=&quot;.&quot;</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

Example A.116 Editing a virtual appliance virtual machine

OVM> edit VirtualApplianceVm name=MyVirtualApplianceVm description='Oracle Linux Release 6'
A.92 edit VirtualCdrom

Edits an ISO file/CDROM.

Syntax

edit VirtualCdrom instance [sharable={Yes|No}][name=value][description=value]

Where instance is:

{ id=value | name=value }  

Description

This command edits the attributes of an ISO file/CDROM.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sharable={Yes</td>
<td>No}</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the ISO file/CDROM.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the ISO file/CDROM. value is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>
Examples

Example A.117 Editing a ISO file/CDROM

```
OVM> edit VirtualCdrom id=0004fb0000150000cd7223d8105042c5.iso name="OL6" \
    description="Oracle Linux 6"
```

See Also

- Section A.134, “importVirtualCdrom”
- Section A.25, “cloneCdToRepo”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

A.93 edit VirtualDisk

Edits a virtual disk.

Syntax

```
edit VirtualDisk instance [shareable = {Yes | No}] [name=value] [description=value]
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command edits the attributes of a virtual disk in a storage repository. To resize a virtual disk, use the `resize` command.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>shareable</code> = {Yes</td>
<td>No}</td>
</tr>
<tr>
<td><code>name=value</code></td>
<td>A name to identify the virtual disk.</td>
</tr>
</tbody>
</table>
| `description=value`       | Optional description for the virtual disk. `value` is a maximum of 4,000 characters.  
                            | To set an empty description, use `description=""`.                           |
| `{id=value | name=value}`     | The instance of the object using either the `id` or `name` option, for example `name=MyVirtualDisk`. |
Examples

Example A.118 Editing a virtual disk

```
OVM> edit VirtualDisk name=MyVMDisk name='New name for MyVMDisk' description='My virtual disk'
```

See Also

- Section A.135, “importVirtualDisk”
- Section A.51, “create VirtualDisk”
- Section A.161, “resize”
- Section A.59, “delete”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

A.94 edit VlanInterface

Edits a VLAN interface.

Syntax

```
edit VlanInterface instance[mtu=value][name=value][description=value]
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command edits the attributes of a VLAN interface.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mtu=value</td>
<td>The MTU value. May be an integer between 1500 and [64000].</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the VLAN interface.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the VLAN interface. \texttt{value} is a maximum of 4,000 characters. To set an empty description, use \texttt{description=&quot;&quot;}.</td>
</tr>
</tbody>
</table>
Examples

Example A.119 Editing a VLAN interface

```
OVM> edit VlanInterface name=MyVlanInterface mtu=1500
```

See Also

- Section A.52, “create VlanInterface”
- Section A.15, “add VlanInterface”
- Section A.158, “remove VlanInterface”
- Section A.102, “embeddedCreate”
- Section A.103, “embeddedDelete”
- Section A.104, “embeddedEdit”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

A.95 edit Vm

Edits a virtual machine.

Syntax

```
edit Vm instance [memory=value] [memoryLimit=value] [cpuCount=value] [cpuCountLimit=value] [cpuPriority=value] [cpuUtilizationCap=value] [highAvailabilty= {Yes | No}] [hugePagesEnabled= {Yes | No}] [osType=value] [restartActionOnCrash= {RESTART | STOP | RESTART_AFTER_DUMP | STOP_AFTER_DUMP}] [mouseType= {OS_DEFAULT | PS2_MOUSE | USB_MOUSE | USB_TABLET}] [domainType= {XEN_HVM | XEN_HMV_PV_DRIVERS | XEN_PVM | LDOMS_PVM | UNKNOWN}] [keymapName= {en-us | ar | da | de | de-ch | en-gb | es | et | fi | fo | fr | fr-be | fr-ca | fr-ch | hr | hu | is | it | ja | lt | lv | mk | nl | nl-be | No | pl | pt | pt-br | ru | sl | sv | th | tr}] [bootOrder= {PXE | DISK | CDROM}] [networkInstallpath=value] [startPolicy= {BEST_SERVER | BALANCE_SERVER | CURRENT_SERVER | USE_POOL_POLICY}] [viridian= {Yes | No}] [name=value] [description=value]
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command edits the attributes of a virtual machine to change the configuration options.

Although none of the options are mandatory, you must supply at least one option.
Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>memory=value</td>
<td>The memory size the virtual machine is allocated in MB. May be an integer between 32 and 512000. The default is 512.</td>
</tr>
<tr>
<td>memoryLimit=value</td>
<td>The maximum memory size the virtual machine can be allocated in MB. May be an integer between 32 and 512000. The default is 512.</td>
</tr>
<tr>
<td>cpuCount=value</td>
<td>The number of processors the virtual machine is allocated. May be an integer between 1 and 999999, but cannot exceed the maximum limit for the domain type, as follows:</td>
</tr>
<tr>
<td></td>
<td>• PVM: 256.</td>
</tr>
<tr>
<td></td>
<td>• HVM: 128; or 32 for Microsoft Windows guests.</td>
</tr>
<tr>
<td></td>
<td>• PVHVM: 128; or 32 if using Oracle VM Paravirtual Drivers for Microsoft Windows.</td>
</tr>
<tr>
<td></td>
<td>• LDoms_PVM: Equivalent to the number of available CPUs on the server.</td>
</tr>
<tr>
<td>cpuCountLimit=value</td>
<td>The maximum number of processors the virtual machine can be allocated. May be an integer between 1 and 999999, but cannot exceed the maximum limit for the domain type, as follows:</td>
</tr>
<tr>
<td></td>
<td>• PVM: 256.</td>
</tr>
</tbody>
</table>

**Important**

As of Oracle VM Release 3.4.6, support for PVM guests is removed. For more information, see Disabling Paravirtualized Guests on Oracle VM Server in the Oracle VM Administrator's Guide.

**Note**

As of Release 3.4.6, the HVM limit is 64 for Microsoft Windows guests.

As of Release 3.4.6, the PVHVM limit is 64 if using Oracle VM Paravirtual Drivers for Microsoft Windows.

As of Release 3.4.6, the LDoms_PVM limit is 512.
### Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>For more information, see <a href="https://docs.oracle.com/en/middleware/cloud/oracle-vm/oracle-virtual-machine/index.html">Disabling Paravirtualized Guests on Oracle VM Server</a> in the Oracle VM Administrator's Guide.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>As of Release 3.4.6, the HVM limit is 64 for Microsoft Windows guests.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>As of Release 3.4.6, the PVHVM limit is 64 if using Oracle VM Paravirtual Drivers for Microsoft Windows.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>The HugePages feature is deprecated in Oracle VM Release 3.4.1. You should not enable HugePages when creating or editing virtual machines in the Oracle VM Manager Web Interface or Oracle VM Manager Command Line Interface. This feature will be removed in a future release of Oracle VM.</td>
</tr>
<tr>
<td></td>
<td>If you have HugePages enabled for any PVM guests, Oracle recommends that you change the domain type for virtual machines from Paravirtualized (PVM) to Hardware virtualized, with paravirtualized drivers (PVHVM). If you cannot change the domain type for a virtual machine, you should disable the HugePages setting and then restart the virtual machine.</td>
</tr>
</tbody>
</table>

- **HVM**: 128; or 32 for Microsoft Windows guests.

- **PVHVM**: 128; or 32 if using Oracle VM Paravirtual Drivers for Microsoft Windows.

- **LDOMS_PVM**: Equivalent to the number of available CPUs on the server.

**cpuPriority=value**  
The CPU priority of the virtual machine. A value between 1 and 100; the higher the number, the more priority the CPU is given.

**cpuUtilizationCap=value**  
The maximum percentage to which the virtual CPUs can receive scheduled time. A value between 10 and 100; the higher the number, the more scheduled time the CPU is given.

**highAvailability= {Yes | No}**  
Whether to enable High Availability (HA).

**hugePagesEnabled= {Yes | No}**  
Whether to enable HugePages.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>osType= value</td>
<td>The operating system of the virtual machine. To find the operating system type, use the <code>getVmOsTypes</code> command.</td>
</tr>
<tr>
<td>[restartActionOnCrash=</td>
<td>The action to perform in the case where a virtual machine crashes. This option must only be used for virtual machines that are running on the Xen hypervisor and that have been configured to run on a particular Oracle VM Server.</td>
</tr>
<tr>
<td>{RESTART</td>
<td>STOP</td>
</tr>
<tr>
<td></td>
<td>Note that in the case where High Availability (HA) is enabled for a virtual machine, the virtual machine is always restarted in the event of a crash, since HA takes precedence over this setting. Therefore, you should be aware that if the <code>highAvailability</code> parameter is set to the value of <code>Yes</code>, any value for this option, other than <code>RESTART</code> and <code>RESTART_AFTER_DUMP</code>, is effectively ignored.</td>
</tr>
<tr>
<td></td>
<td>If you select and option to generate a core dump file, you should be aware that these are saved to <code>/var/xen/dump</code> on the Oracle VM Server where the virtual machine is hosted. Each core dump file is named uniquely so that files are not overwritten. This can use up disk space rapidly. You must ensure that there is either enough disk space available at this path on the Oracle VM Server where the virtual machine will run; or you should mount additional storage at this path to avoid using up disk space required to host dom0.</td>
</tr>
<tr>
<td>mouseType= { OS_DEFAULT</td>
<td>PS2_MOUSE</td>
</tr>
<tr>
<td>domainType= { XEN_HVM</td>
<td>XEN_HVM_PV_DRIVERS</td>
</tr>
<tr>
<td>keymapName= { en-us</td>
<td>ar</td>
</tr>
<tr>
<td>bootOrder= { PXE</td>
<td>DISK</td>
</tr>
<tr>
<td>networkInstallpath=value</td>
<td>The location at which the installation media (mounted ISO file) is located when creating a PVM guest.</td>
</tr>
<tr>
<td>startPolicy= { BEST_SERVER</td>
<td>BALANCE_SERVER</td>
</tr>
</tbody>
</table>
Option | Description
--- | ---
**viridian= { Yes | No }** | Viridian support enables the exposure of Windows virtualization compatible entitlements to Microsoft Windows guest operating systems. Enabling viridian support is strongly recommended to ensure improved performance for Microsoft Windows guest operating systems. Enabling viridian support is permitted for all Microsoft Windows guest operating system types. However, it is only effective from Microsoft Windows Vista and Microsoft Windows Server 2008 onwards.

**name=value** | A name to identify the virtual machine.

**description=value** | Optional description for the virtual machine. value is a maximum of 4,000 characters. To set an empty description, use description="".

{ **id=value | name=value** } | The instance of the object using either the id or name option, for example name=MyVM.

**Examples**

**Example A.120 Edit a virtual machine**

OVM> edit Vm name=MyVM bootOrder='CDROM,DISK' startPolicy=BEST_SERVER

**See Also**

- Section A.126, “getVmOsTypes”
- Section A.24, “clone Vm”
- Section A.16, “add Vm”
- Section A.177, “start”
- Section A.138, “list”
- Section A.171, “show”

**A.96 edit VmCloneCustomizer**

Edits an existing clone customizer for a virtual machine.

**Syntax**

```
edit VmCloneCustomizer instance [name=value] [description=value]
```

Where instance is:

{ **id=value | name=value** }

**Description**

This command allows you to edit an existing clone customizer for a virtual machine.
Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name=value</td>
<td>A name to identify the clone customizer.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the clone customizer object. value is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.121 Edit a virtual machine clone customizer

```bash
OVMD> edit VmCloneCustomizer name=MyVmCloneCustomizer name=MyCloneCustomizer \     description="A test clone customizer"
```

See Also

- Section A.98, “edit VmCloneStorageMapping”
- Section A.97, “edit VmCloneNetworkMapping”
- Section A.54, “create VmCloneCustomizer”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

A.97 edit VmCloneNetworkMapping

Edits an existing network mapping for a virtual machine clone customizer.

Syntax

```bash
edit VmCloneNetworkMapping instance [network=value] [name=value] [description=value]
```
Where \textit{instance} is:

\{ \textit{id}=\textit{value} \mid \textit{name}=\textit{value} \}

\section*{Description}

Edits an existing network mapping for a virtual machine clone customizer.

Although none of the options are mandatory, you must supply at least one option.

\section*{Options}

The following table shows the available options for this command.

\begin{center}
\begin{tabular}{|l|l|}
\hline
\textbf{Option} & \textbf{Description} \\
\hline
\texttt{network=\textit{value}} & The name or ID value of an existing network. \\
\texttt{name=\textit{value}} & A name to identify the clone network mapping. \\
\texttt{description=\textit{value}} & Optional description for the clone network mapping object. \textit{value} is a maximum of 4,000 characters. \textit{To set an empty description, use \texttt{description='\textquotedbl'}\textit{".}} \\
\texttt{\{} \textit{id}=\textit{value} \mid \textit{name}=\textit{value} \texttt{\}} & The instance of the object using either the \texttt{id} or \texttt{name} option, for example \texttt{name=MyCloneNet}. \\
\hline
\end{tabular}
\end{center}

\section*{Examples}

\textbf{Example A.122 Editing a virtual machine clone network mapping}

\texttt{OVM> edit VmCloneNetworkMapping name=MyCloneNet description="A clone network mapping"}

\section*{See Also}

\begin{itemize}
\item Section A.54, “create VmCloneCustomizer”
\item Section A.55, “create VmCloneNetworkMapping”
\item Section A.59, “delete”
\item Section A.138, “list”
\item Section A.171, “show”
\end{itemize}

\section*{A.98 edit VmCloneStorageMapping}

Edits an existing clone customizer storage mapping.

\section*{Syntax}

\texttt{edit VmCloneStorageMapping instance[ cloneType=\{ SPARSE\textunderscore COPY \mid NON\textunderscore SPARSE\textunderscore COPY \mid THIN\_CLONE \}\} [physicalDisk=\textit{value} \mid repository=\textit{value} \mid storageArray=\textit{value}] [\texttt{name=\textit{value}}] [\texttt{description=\textit{value}}]

Where \textit{instance} is:
Description

Edits an existing clone customizer storage mapping.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloneType= { SPARSE_COPY</td>
<td>NON_SPARSE_COPY</td>
</tr>
<tr>
<td>{physicalDisk=value</td>
<td>repository=value</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the clone storage mapping.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the clone storage mapping object. value is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use description=&quot;&quot;.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.123 Editing a virtual machine clone storage mapping

```
OVM> edit VmCloneStorageMapping name=MyVMCloneStorage cloneType=THIN_CLONE
```

See Also

- Section A.54, “create VmCloneCustomizer”
- Section A.98, “edit VmCloneStorageMapping”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

A.99 edit VmDiskMapping

Edits the virtual machine disk mapping object.

Syntax

```
edit VmDiskMapping instance [virtualCd=instance] [name=value] [description=value]
```
Description

This command edits the virtual machine disk mapping object. Do not use this command to change the disk mapping in a virtual machine, this is just for the object that contains the disk mapping information. To edit a virtual disk or CDROM mapped to a virtual machine, remove it using the delete VmDiskMapping command, then use the create VmDiskMapping command again to remap it to a virtual machine with any changed settings.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>virtualCd=instance</td>
<td>The name or ID of the ISO file (virtual CDROM).</td>
</tr>
<tr>
<td>name=value</td>
<td>A name to identify the disk mapping.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the disk mapping object. value is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.124 Editing a virtual disk mapping object

OVM> edit VmDiskMapping id=0004fb0000130000409cd9340443e257 name=MyDiskMap

See Also

- Section A.57, “create VmDiskMapping”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

A.100 edit Vnic
Edits a VNIC.

Syntax

```bash
eedit Vnic instance [name=value] [description=value] [macAddress=value]
```

Where `instance` is:

```bash
{id=value} | {name=value}
```

Description

This command edits the attributes of a VNIC on a network.

To change the network assignment of a VNIC, please see Section A.17, “add Vnic”.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>name=value</code></td>
<td>A name for the VNIC.</td>
</tr>
<tr>
<td><code>description=value</code></td>
<td>Optional description for the VNIC. <code>value</code> is a maximum of 4,000 characters.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td><code>macAddress=value</code></td>
<td>The MAC address that should be applied to this VNIC.</td>
</tr>
<tr>
<td>`{id=value}</td>
<td>{name=value}`</td>
</tr>
</tbody>
</table>

Examples

**Example A.125 Editing a VNIC**

```bash
OVM> edit Vnic id=0004fb000000007000007fa68ff0d2011539f name=Server1Vnic macAddress=00:21:f6:00:00:18
```

See Also

- Section A.58, “create Vnic”
- Section A.17, “add Vnic”
- Section A.160, “remove Vnic”
- Section A.170, “setVnicMacAddrRange”
- Section A.129, “getVnicMacAddrRange”
- Section A.59, “delete”
- Section A.138, “list”
A.101 edit VolumeGroup

Edits a volume group object.

Syntax

```
edit VolumeGroup instance [name=value][description=value]
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command edits the attributes of a volume group object.

Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network=value</td>
<td>The name or ID of the volume group object.</td>
</tr>
<tr>
<td>name=value</td>
<td>A name for the volume group object.</td>
</tr>
<tr>
<td>description=value</td>
<td>Optional description for the volume group object. <code>value</code> is a maximum of 4,000 characters. To set an empty description, use <code>description=&quot;&quot;</code>.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.126 Editing a volume group

```
OVM> edit VolumeGroup name=MyVolumeGroup name=MyNewName
```

See Also

- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

A.102 embeddedCreate

Creates an IP address object and adds it to a network related object.
Syntax

```
embeddedCreate {BondPort | Port | VlanInterface} instance ipAddressConfig
ipAddressConfigType={STATIC | DYNAMIC} [ipAddress=value] [ipNetmask=value]
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command adds an IP address and its associated configuration (ipAddressConfig object) to a bond port, port, or VLAN interface. Note that if the `ipAddressConfigType` is set to `DYNAMIC`, then any values specified for `ipAddress` or `ipNetmask` are ignored.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{BondPort</td>
<td>Port</td>
</tr>
<tr>
<td>ipAddressConfigType={STATIC</td>
<td>DYNAMIC}</td>
</tr>
<tr>
<td><code>[ipAddress=value]</code></td>
<td>The IP address. Note that this value is ignored if the <code>ipAddressConfigType</code> is set to <code>DYNAMIC</code>.</td>
</tr>
<tr>
<td><code>[ipNetmask=value]</code></td>
<td>The netmask. Note that this value is ignored if the <code>ipAddressConfigType</code> is set to <code>DYNAMIC</code>.</td>
</tr>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

Examples

**Example A.127 Adding an IP address to a port**

```
OVM> embeddedCreate Port id=0004fb00002000003ea1bffb91ece960 ipAddressConfig \n    ipAddressConfigType=STATIC ipAddress=10.172.76.100 ipNetmask=255.255.254.0
```

**Example A.128 Adding an IP address to a bond port**

```
OVM> embeddedCreate BondPort id=0004fb00002000000a5389824228bdf1 ipAddressConfig \n    ipAddressConfigType=STATIC ipAddress=10.172.76.100 ipNetmask=255.255.254.0
```

See Also

- Section A.103, “embeddedDelete”
- Section A.104, “embeddedEdit”
- Section A.138, “list”
- Section A.171, “show”
A.103 embeddedDelete

Deletes an IP address object and removes it from a network related object.

Syntax

```
embeddedDelete { BondPort | Port | VlanInterface } instance ipAddressConfig id=value
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command removes an IP address and its associated configuration (ipAddressConfig object) from a bond port, port, or VLAN interface.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{ BondPort</td>
<td>Port</td>
</tr>
<tr>
<td><code>id=value</code></td>
<td>The index value of the ipAddressConfig object. To find the index value, use the <code>show</code> command and look for the embedded object's IP address information. For example, to find the index value for an IP address object on a port:</td>
</tr>
<tr>
<td></td>
<td>OVM&gt; show Port id=0004fb00002000003ea1bffb91ece960</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>Ip Address Config 1 - Address Type = Ipv4</td>
</tr>
<tr>
<td></td>
<td>Ip Address Config 1 - Config Type = Static</td>
</tr>
<tr>
<td></td>
<td>Ip Address Config 1 - Address = 10.172.76.100</td>
</tr>
<tr>
<td></td>
<td>Ip Address Config 1 - Netmask = 255.255.254.0</td>
</tr>
<tr>
<td></td>
<td>Interface Name = eth2</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>The index value in this case is 1.</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Examples

**Example A.129 Deleting an IP address from a port**

```
OVM> embeddedDelete Port id=0004fb00002000003ea1bffb91ece960 ipAddressConfig id=1
```

**Example A.130 Deleting an IP address from a bond port**

```
OVM> embeddedDelete BondPort id=0004fb00002000000a5389824228bdf1 ipAddressConfig id=1
```

See Also

- Section A.102, "embeddedCreate"
A.104 embeddedEdit

Edits an IP address object on a network related object.

Syntax

```
embeddedEdit { BondPort | Port | VlanInterface } instance ipAddressConfig id=value ipAddressConfigType={ STATIC | DYNAMIC }[ ipAddress=value ][ ipNetmask=value ]
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command edits the attributes of an IP address and its associated configuration (ipAddressConfig object) on a bond port, port, or VLAN interface. Note that if the `ipAddressConfigType` is set to `DYNAMIC`, then any values specified for `ipAddress` or `ipNetmask` are ignored.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ BondPort</td>
<td>Port</td>
</tr>
<tr>
<td><code>id=value</code></td>
<td>The index value of the ipAddressConfig object. To find the index value, use the <code>show</code> command and look for the embedded object's IP address information. For example, to find the index value for an IP address object on a port:</td>
</tr>
<tr>
<td>`ipAddressConfigType={ STATIC</td>
<td>DYNAMIC }`</td>
</tr>
<tr>
<td><code>[ ipAddress=value ]</code></td>
<td>The IP address. Note that this value is ignored if the <code>ipAddressConfigType</code> is set to <code>DYNAMIC</code>.</td>
</tr>
<tr>
<td><code>[ ipNetmask=value ]</code></td>
<td>The netmask. Note that this value is ignored if the <code>ipAddressConfigType</code> is set to <code>DYNAMIC</code>.</td>
</tr>
</tbody>
</table>
Option | Description
--- | ---
{id=value | name=value} | The instance of the object using either the id or name option, for example name=MyPort.

### Examples

**Example A.131 Editing an IP address on a port**

```
OVM> embeddedEdit Port id=0004fb00002000003ea1bffb91ece960 ipAddressConfig \
    id=1 ipAddressConfigType=STATIC ipAddress=10.172.76.100 ipNetmask=255.255.254.0
```

**Example A.132 Editing an IP address on a bond port**

```
OVM> embeddedEdit BondPort id=0004fb00002000000a5389824228bdf1 ipAddressConfig \
    id=1 ipAddressConfigType=STATIC ipAddress=10.172.76.100 ipNetmask=255.255.254.0
```

### See Also

- Section A.102, "embeddedCreate"
- Section A.103, "embeddedDelete"
- Section A.138, "list"
- Section A.171, "show"

### A.105 exit

Exits/Quits the CLI.

#### Syntax

```
exit
```

#### Description

This command exits/quiats the CLI.

#### Options

This command does not take any arguments or provide any options.

#### Examples

**Example A.133 Exiting the CLI**

```
OVM> exit
```

### See Also

- Section A.130, “help”

### A.106 exportVirtualAppliance
Exports a virtual appliance to a storage repository.

### Syntax

```
exportVirtualAppliance Repository instance name=value vms=value
```

Where `instance` is:

```
{ id=value | name=value }
```

### Description

This command creates a virtual appliance from one or more virtual machines, then saves the virtual appliance in a storage repository.

### Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Repository</code></td>
<td>The storage repository in which to export the virtual appliance.</td>
</tr>
<tr>
<td><code>instance</code></td>
<td>The name of the virtual appliance.</td>
</tr>
<tr>
<td><code>name=value</code></td>
<td>The names or IDs of the virtual machines to be exported in the virtual appliance, in a comma separated list.</td>
</tr>
<tr>
<td></td>
<td>Note: Virtual machines must be in the <strong>Stopped</strong> state before you can export them.</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

### Examples

**Example A.134 Exporting a virtual appliance to a storage repository**

```
OVM> exportVirtualAppliance Repository name=MyRepository name=MyVirtualAppliance "vms=0004fb00000000004b291334d6730d57,0004fb00000000006000000600000c67aa9bf6ca0f6"
```

### See Also

- Section A.32, "createVmFromVirtualAppliance"
- Section A.33, "createVmFromVirtualApplianceVm"
- Section A.133, "importVirtualAppliance"
- Section A.89, "edit VirtualAppliance"
- Section A.90, "edit VirtualApplianceVirtualDisk"
- Section A.91, "edit VirtualApplianceVm"
A.107 getArchiveConfig

Shows the configuration for managing archived statistics.

Syntax

getArchiveConfig

Description

This command shows the configuration for how Oracle VM Manager manages archived statistics. To set the configuration, use the setArchiveConfig command.

Options

This command does not take any arguments or provide any options.

Examples

Example A.135 Showing the configuration for managing archived statistics

OVM> getArchiveConfig

See Also

- Section A.166, “setArchiveConfig”

A.108 getAverageStat

Lists the average value of a statistic type for an object during a time range.

Syntax

getAverageStat objType={Server | Vm | FileSystem} objId=value statType={CPU_UTILIZATION | CPU_COUNT | MEMORY_USED | FREE_MEMORY | MEMORY_UTILIZATION | FILE_SYSTEM_SPACE_FREE | FILE_SYSTEM_SPACE_UTILIZATION | FILE_SYSTEM_SPACE_TOTAL} startTime=value [ endTime=value ]

Description

This command lists the average statistic for an object during a time range. Statistics are held for a limited amount of time and there may be no statistics of the particular type that occurred during all or part of the specified time range.
## Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>objType=</strong></td>
<td>The object type for which to list the average statistic. The value of <strong>objType</strong> may be <strong>Server</strong>, <strong>Vm</strong>, or <strong>FileSystem</strong>. The value for <strong>objType</strong> is case insensitive.</td>
</tr>
<tr>
<td><strong>objId=value</strong></td>
<td>The name or ID of the object for which statistics should be listed. This must be the ID of a Server, Vm, or FileSystem object.</td>
</tr>
<tr>
<td><strong>statType=</strong></td>
<td>The type of statistic to list:</td>
</tr>
<tr>
<td></td>
<td>• <strong>CPU_UTILIZATION</strong>: The percentage of the total CPU power being used by the object</td>
</tr>
<tr>
<td></td>
<td>• <strong>CPU_COUNT</strong>: The number of CPUs.</td>
</tr>
<tr>
<td></td>
<td>• <strong>MEMORY_USED</strong>: The amount of memory, in MBs, used by the object.</td>
</tr>
<tr>
<td></td>
<td>• <strong>FREE_MEMORY</strong>: The amount of memory, in MBs, available on the server.</td>
</tr>
<tr>
<td></td>
<td>• <strong>MEMORY_UTILIZATION</strong>: The percentage of the total memory being used by the object.</td>
</tr>
<tr>
<td></td>
<td>• <strong>FILE_SYSTEM_SPACE_FREE</strong>: The amount of file system space, in GiB, that is not currently used.</td>
</tr>
<tr>
<td></td>
<td>• <strong>FILE_SYSTEM_SPACE_UTILIZATION</strong>: The percentage of the total file system space currently in use.</td>
</tr>
<tr>
<td></td>
<td>• <strong>FILE_SYSTEM_SPACE_TOTAL</strong>: The total size of the file system, in GiB.</td>
</tr>
<tr>
<td><strong>startTime=value</strong></td>
<td>The start date and time from which to list the average statistic. The format to use is &quot;MM-dd-yyyy HH:mm&quot;.</td>
</tr>
<tr>
<td></td>
<td>The value must not be later than the current time or after the value of <strong>endTime</strong>.</td>
</tr>
</tbody>
</table>
Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endTime=value</td>
<td>The end date and time from which to list the average statistic. The format to use is &quot;MM-dd-yyyy HH:mm&quot;. This is an optional parameter.</td>
</tr>
</tbody>
</table>

Example A.136 Listing average statistics for an Oracle VM Server

```bash
OVM> getAverageStat objType=Server objId=00:e0:81:4d:41:05:00:e0:81:4d:40:de:00:e0:81:4d \
    statType=CPU_UTILIZATION startTime="08-20-2014 00:00"
```

See Also

- Section A.116, “getLatestStat”
- Section A.117, “getLatestStatForList”
- Section A.121, “getStatList”
- Section A.122, “getStatListByQuery”
- Section A.169, “setStatsConfig”
- Section A.120, “getStatsConfig”
- Section A.138, “list”
- Section A.171, “show”

A.109 getDbBackupConfig

Shows the configuration for the automated database backup facility within Oracle VM Manager.

Syntax

```bash
getDbBackupConfig
```

Description

This command shows the configuration for the automated database backup facility within Oracle VM Manager. Further information on the automated database backup facility is provided in Backing up the MySQL Database Repository in the Oracle VM Administrator’s Guide.

Options

This command does not take any arguments or provide any options.

Examples

Example A.137 Showing the Oracle VM Manager database backup configuration

```bash
OVM> getDbBackupConfig
```

See Also

- Section A.167, “setDbBackupConfig”
A.110 getDebugTranscript

Shows the debug transcript of a job.

Syntax

getDebugTranscript Job instance

Where instance is:

{ id=value | name=value }

Description

This command shows the debug transcript of a job.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

Example A.138 Showing the debug transcript of a job

OVM> getDebugTranscript Job id=1373345941846

See Also

- Section A.1, “abort Job”
- Section A.115, “getJobs”
- Section A.138, “list”
- Section A.171, “show”

A.111 getDescriptor

Lists the descriptor file for a virtual appliance or assembly.

Syntax

getDescriptor {Assembly | VirtualAppliance} instance

Where instance is:

{ id=value | name=value }

Description

This command lists the descriptor file for a virtual appliance or assembly. The descriptor (.ovf file) is the main file in a virtual appliance or assembly package, and contains meta-data for the virtual appliance or assembly, including links to external files, such as virtual disks.
Important
The Assembly option has been deprecated. You should instead use VirtualAppliance.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

Example A.139 Listing the descriptor for a virtual appliance

```
OVM> getDescriptor VirtualAppliance name=MyVirtualAppliance.ova
```

See Also

- Section A.131, “importAssembly (Deprecated)”
- Section A.31, “createVmFromAssembly (Deprecated)”
- Section A.24, “clone Vm”
- Section A.63, “edit Assembly (Deprecated)”
- Section A.64, “edit AssemblyVirtualDisk (Deprecated)”
- Section A.65, “edit AssemblyVm (Deprecated)”
- Section A.59, “delete”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”

A.112 getEventListByQuery

Lists the events for an object using a query to narrow results.

Syntax

```
getEventListByQuery [[objType=value][objIds=value]][ severity={UNKNOWN | CRITICAL | WARNING | INFORMATIONAL}][ severityEqualityType={LT | LTE | GT | GTE | EQ | NE}][ eventTypes= value][ eventEqualityType={LIKE | NOT_LIKE }][ acknowledged={Yes | No}][[ userAckable={Yes | No}][ startTime=value][ endTime=value][ maxResults=value]
```

Description

This command lists the events for an object using a query to narrow results. To acknowledge an event, use the ackEvent command.
## Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>objType=value</strong></td>
<td>The object type for which to list the events. The <code>value</code> of <code>objType</code> is an object such as <code>FileServer</code>, <code>Repository</code>, or <code>Server</code>. To get a list of all object types, use the <code>showobjtypes</code>. The value for <code>objType</code> is case insensitive. This is an optional parameter and must be used with the <code>objIds</code> option.</td>
</tr>
<tr>
<td><strong>objIds=value</strong></td>
<td>The ID of the objects for which events should be listed. To enter one or more objects, use a comma separated list. This is an optional parameter and must be used with the <code>objType</code> option.</td>
</tr>
<tr>
<td><strong>severity=</strong></td>
<td>The severity of the event to list. Use this in combination with the <code>equalityType</code> option. The events are ranked in the order from low to high:</td>
</tr>
<tr>
<td>{UNKNOWN</td>
<td>CRITICAL</td>
</tr>
<tr>
<td></td>
<td>• <strong>INFORMATIONAL</strong>: An event which is used to convey information about the system.</td>
</tr>
<tr>
<td></td>
<td>• <strong>WARNING</strong>: An event which is only likely to have a minor impact to system functionality.</td>
</tr>
<tr>
<td></td>
<td>• <strong>CRITICAL</strong>: An event which could cause a critical loss of functionality which requires immediate attention.</td>
</tr>
<tr>
<td></td>
<td>When you use this in conjunction with the <code>severityEqualityType</code> option, the comparison is performed using this ranking. For example, if you use:</td>
</tr>
<tr>
<td></td>
<td><code>severity=WARNING severityEqualityType=GTE</code></td>
</tr>
<tr>
<td></td>
<td>All events greater than or equal to WARNING are returned. This means all WARNING and all CRITICAL events are returned.</td>
</tr>
<tr>
<td></td>
<td>This is an optional parameter.</td>
</tr>
<tr>
<td><strong>severityEqualityType=</strong></td>
<td>The equality type for the severity comparison. Use this in combination with the <code>severity</code> option. The parameters are:</td>
</tr>
<tr>
<td>{LT</td>
<td>LTE</td>
</tr>
<tr>
<td></td>
<td>• <strong>LTE</strong>: Less than or equal to (&lt;=).</td>
</tr>
<tr>
<td></td>
<td>• <strong>GT</strong>: Greater than (&gt;).</td>
</tr>
<tr>
<td></td>
<td>• <strong>GTE</strong>: Greater than or equal to (&gt;=).</td>
</tr>
<tr>
<td></td>
<td>• <strong>EQ</strong>: Equal to (=).</td>
</tr>
</tbody>
</table>
### Option  | Description
--- | ---
• **NE**: Not equal to (<>), or !=

| eventTypes=value | The event type. Use this to filter the results to a specific event or events. For example: `eventTypes=server.maintenance.mode`.

Note the period (.) at the end of the event type. This is important. You cannot truncate the event type; it must be in full. No wild cards are allowed. For a list of the available options, see Table A.1, “eventTypes Values”.

For the most up-to-date list of event types, see the Oracle VM Manager Web Services API 3.4 documentation at: [http://hostname/WSAPI/doc/api/webservices_s/com/oracle/ovm/mgr/ws/model/Event.html](http://hostname/WSAPI/doc/api/webservices_s/com/oracle/ovm/mgr/ws/model/Event.html)

This documentation is available on the Oracle VM Manager ISO/CD. You should deploy it to a web server to access the content.

| eventEqualityType= {LIKE | NOT_LIKE} | The event equality type for the event types comparison. Use this in combination with the eventTypes option. The parameters are:

• **LIKE**: Is like the value of eventTypes.

• **NOT_LIKE**: Is not like the value of eventTypes.

| acknowledged= {Yes | No} | Whether to include acknowledged events in the listing. The default is No, so all events that have been acknowledged are not displayed.

This is an optional parameter.

| userAcknowledged= {Yes | No} | Whether to include events that can be acknowledged by a user in the listing. The default is No, so all events that cannot be acknowledged by a user are removed from the results.

This is an optional parameter.

| startTime=value | The start date and time from which to list events. The format to use is "MM-dd-yyyy HH:mm:ss".

This is an optional parameter.

| endTime=value | The end date and time from which to list events. The format to use is "MM-dd-yyyy HH:mm:ss".

This is an optional parameter.

| maxResults=value | The maximum number of events to list.

This is an optional parameter.

---

**eventTypes Values**

This table shows the available options for the `eventTypes` option of this command.
Table A.1 eventTypes Values

<table>
<thead>
<tr>
<th>eventType Value</th>
<th>Event Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lifecycle Events</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lifecycle.</td>
<td>INFORMATIONAL</td>
<td>This eventType is not used directly. However, you can use this when retrieving Events by type to retrieve all lifecycle eventTypes.</td>
</tr>
<tr>
<td>lifecycle.create.</td>
<td>INFORMATIONAL</td>
<td>Object has been created.</td>
</tr>
<tr>
<td>lifecycle.modify.</td>
<td>INFORMATIONAL</td>
<td>Object has been modified.</td>
</tr>
<tr>
<td>lifecycle.delete.</td>
<td>INFORMATIONAL</td>
<td>Object has been deleted.</td>
</tr>
<tr>
<td><strong>Runstate Events</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>runstate.</td>
<td>INFORMATIONAL</td>
<td>This eventType is not used directly. However, you can use this when retrieving Events by type to retrieve all runstate eventTypes.</td>
</tr>
<tr>
<td>runstate.starting.</td>
<td>INFORMATIONAL</td>
<td>Object is in the process of starting.</td>
</tr>
<tr>
<td>runstate.running.</td>
<td>INFORMATIONAL</td>
<td>Object is running.</td>
</tr>
<tr>
<td>runstate.stopping.</td>
<td>INFORMATIONAL</td>
<td>Object is in the process of stopping.</td>
</tr>
<tr>
<td>runstate.stopped.</td>
<td>INFORMATIONAL</td>
<td>Object is stopped.</td>
</tr>
<tr>
<td>runstate.suspended.</td>
<td>INFORMATIONAL</td>
<td>Object is suspended.</td>
</tr>
<tr>
<td><strong>Discover Events</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>discover.failure.</td>
<td>INFORMATIONAL</td>
<td>There was a failure during server discovery.</td>
</tr>
<tr>
<td><strong>Virtual machine Events</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm.api.incoming.</td>
<td>INFORMATIONAL</td>
<td>Virtual machine incoming message.</td>
</tr>
<tr>
<td>vm.api.outgoing.</td>
<td>INFORMATIONAL</td>
<td>Virtual machine outgoing message.</td>
</tr>
<tr>
<td>vm.error.</td>
<td>CRITICAL</td>
<td>This eventType is not used directly. However, you can use this when retrieving Events by type to retrieve all vm.error eventTypes.</td>
</tr>
<tr>
<td>vm.error.disconnected.</td>
<td>CRITICAL</td>
<td>Virtual machine has disconnected.</td>
</tr>
<tr>
<td>vm.error.migration.</td>
<td>CRITICAL</td>
<td>Virtual machine migration failed.</td>
</tr>
<tr>
<td>vm.warn.migration.</td>
<td>WARNING</td>
<td>This eventType is not used directly. However, you can use this when retrieving Events by type to retrieve all vm.warn.migration eventTypes.</td>
</tr>
<tr>
<td>eventType Value</td>
<td>Event Severity</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>vm.warn.migration.src.</td>
<td>WARNING</td>
<td>Virtual machine migration warning from the server that holds the source (from) the virtual machine.</td>
</tr>
<tr>
<td>vm.warn.migration.tgt.</td>
<td>WARNING</td>
<td>Virtual machine migration warning from the server that is the target (to) for the virtual machine.</td>
</tr>
<tr>
<td>vm.domain.mismatch.</td>
<td>INFORMATIONAL</td>
<td>Virtual machine domain type mismatch.</td>
</tr>
<tr>
<td>vm.migrating.</td>
<td>INFORMATIONAL</td>
<td>Virtual machine is migrating.</td>
</tr>
<tr>
<td>vm.misplaced.</td>
<td>INFORMATIONAL</td>
<td>Virtual machine has been found on an incompatible server.</td>
</tr>
<tr>
<td>vm.os.mismatch.</td>
<td>INFORMATIONAL</td>
<td>Virtual machine operating system type mismatch.</td>
</tr>
<tr>
<td>vm.disk.missing.</td>
<td>CRITICAL</td>
<td>Virtual machine is missing a virtual disk.</td>
</tr>
<tr>
<td>vm.error.offline.</td>
<td>CRITICAL</td>
<td>Virtual machine is offline.</td>
</tr>
</tbody>
</table>

**Server Events**

<table>
<thead>
<tr>
<th>eventType Value</th>
<th>Event Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server.failure</td>
<td>CRITICAL</td>
<td>Server has experienced a failure.</td>
</tr>
<tr>
<td>server.discovery.failed.</td>
<td>CRITICAL</td>
<td>Server discovery has failed.</td>
</tr>
<tr>
<td>server.discovery.started.</td>
<td>INFORMATIONAL</td>
<td>Server discovery is in progress.</td>
</tr>
<tr>
<td>server.repository.error.</td>
<td>CRITICAL</td>
<td>The server repository has experienced an error.</td>
</tr>
<tr>
<td>server.cluster.failure.</td>
<td>CRITICAL</td>
<td>The server cluster has experienced a failure.</td>
</tr>
<tr>
<td>server.cluster.state.</td>
<td>INFORMATIONAL</td>
<td>This eventType is not used directly. However, you can use this when retrieving Events by type to retrieve all server.cluster.state eventTypes.</td>
</tr>
<tr>
<td>server.cluster.state.down.</td>
<td>INFORMATIONAL</td>
<td>The server's cluster is down.</td>
</tr>
<tr>
<td>server.cluster.state.up.</td>
<td>INFORMATIONAL</td>
<td>The server's cluster is up.</td>
</tr>
<tr>
<td>server.disconnected.</td>
<td>CRITICAL</td>
<td>The server has disconnected.</td>
</tr>
<tr>
<td>server.authentication.error.</td>
<td>CRITICAL</td>
<td>There was a server authentication error.</td>
</tr>
<tr>
<td>server.disk.reserved.</td>
<td>INFORMATIONAL</td>
<td>The server disk is reserved.</td>
</tr>
<tr>
<td>server.disk.unreserved.</td>
<td>INFORMATIONAL</td>
<td>The server disk is unreserved.</td>
</tr>
<tr>
<td>server.kernel.invalid.</td>
<td>CRITICAL</td>
<td>The server kernel is invalid.</td>
</tr>
<tr>
<td>server.maintenance.mode.</td>
<td>INFORMATIONAL</td>
<td>The server is in maintenance mode.</td>
</tr>
<tr>
<td>server.out.of.date.</td>
<td>INFORMATIONAL</td>
<td>The server software is out of date with regards to that found in the update repository.</td>
</tr>
<tr>
<td>eventType Value</td>
<td>Event Severity</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>server.pool.</td>
<td>INFORMATIONAL</td>
<td>This eventType is not used directly. However, you can use this when retrieving Events by type to retrieve all server.pool eventTypes.</td>
</tr>
<tr>
<td>server.pool.failure.</td>
<td>CRITICAL</td>
<td>There has been a failure with the server pool.</td>
</tr>
<tr>
<td>server.pool.file.system.missing.</td>
<td>CRITICAL</td>
<td>The server pool file system is missing.</td>
</tr>
<tr>
<td>server.pool.master.missing.</td>
<td>CRITICAL</td>
<td>Server pool is missing the master.</td>
</tr>
<tr>
<td>server.pool.vip.unavailable.</td>
<td>WARNING</td>
<td>Server pool virtual IP address is unavailable.</td>
</tr>
<tr>
<td>server.evacuate.failed.</td>
<td>WARNING</td>
<td>Server failed to evacuate.</td>
</tr>
<tr>
<td>server.mount.missing.</td>
<td>WARNING</td>
<td>Server mount is missing.</td>
</tr>
<tr>
<td>server.network.operation.failed.</td>
<td>WARNING</td>
<td>Server network operation failed.</td>
</tr>
<tr>
<td>server.offline.</td>
<td>CRITICAL</td>
<td>Server is offline.</td>
</tr>
<tr>
<td>server.ownership.mismatch.</td>
<td>WARNING</td>
<td>Server is owned by another Oracle VM Manager.</td>
</tr>
<tr>
<td>server.version.mismatch.</td>
<td>WARNING</td>
<td>There is an API version mismatch with the server.</td>
</tr>
<tr>
<td>server.update.repository.mismatch.</td>
<td>INFORMATIONAL</td>
<td>The repository information on the server is not the same as configured in Oracle VM Manager. This will trigger an update process so the information on the server is reconfigured.</td>
</tr>
<tr>
<td>server.update.repository.check.</td>
<td>INFORMATIONAL</td>
<td>Server is being checked if its software update repository is up to date.</td>
</tr>
<tr>
<td>server.update.repository.check.failed</td>
<td>INFORMATIONAL</td>
<td>There was a failure during the update repository configuration.</td>
</tr>
<tr>
<td>server.update.repository.config.failed</td>
<td>INFORMATIONAL</td>
<td>There was a failure during the repository configuration update check.</td>
</tr>
<tr>
<td>server.duplicate.storage.initiator.</td>
<td>WARNING</td>
<td>Server has a duplicate storage initiator.</td>
</tr>
<tr>
<td>server.network.mismatch.</td>
<td>WARNING</td>
<td>A networking device was found during discovery on this server with a network definition that includes the Virtual Machine role. This network already exists in Oracle VM Manager without the Virtual Machine role. This role will not be automatically added to the network and the server's view of the network is out of sync with</td>
</tr>
</tbody>
</table>
### eventTypes Values

<table>
<thead>
<tr>
<th>eventType Value</th>
<th>Event Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server.upgrade.in.progress.</td>
<td>INFORMATIONAL</td>
<td>Server upgrade in progress.</td>
</tr>
<tr>
<td>server.upgrade.config.in.progress.</td>
<td>INFORMATIONAL</td>
<td>A configuration of the server upgrade repositories is in progress.</td>
</tr>
<tr>
<td>filesystem.missing.</td>
<td>WARNING</td>
<td>File system is missing.</td>
</tr>
<tr>
<td>filesystem.size.mismatch.</td>
<td>CRITICAL</td>
<td>File system is larger than its underlying storage device.</td>
</tr>
<tr>
<td>repository.empty.</td>
<td>CRITICAL</td>
<td>Repository is empty.</td>
</tr>
<tr>
<td>repository.error.</td>
<td>CRITICAL</td>
<td>There is a repository error.</td>
</tr>
<tr>
<td>repository.missing.</td>
<td>WARNING</td>
<td>Repository is missing.</td>
</tr>
<tr>
<td>repository.unmounted.</td>
<td>CRITICAL</td>
<td>Repository is unmounted</td>
</tr>
<tr>
<td>storage.device.</td>
<td>INFORMATIONAL</td>
<td>Storage device is offline.</td>
</tr>
<tr>
<td>storage.device.offline.</td>
<td>WARNING</td>
<td>Storage device is offline.</td>
</tr>
<tr>
<td>storage.device.online.</td>
<td>INFORMATIONAL</td>
<td>Storage device is online.</td>
</tr>
<tr>
<td>storage.device.off.path.</td>
<td>WARNING</td>
<td>At least one path is missing for storage element</td>
</tr>
<tr>
<td>mgr.db.scan.error.</td>
<td>CRITICAL</td>
<td>Database scan error.</td>
</tr>
<tr>
<td>mgr.db.scan.ok.</td>
<td>INFORMATIONAL</td>
<td>Database scan succeeded.</td>
</tr>
<tr>
<td>mgr.db.backup.fail.</td>
<td>WARNING</td>
<td>Database backup failed.</td>
</tr>
<tr>
<td>mgr.db.timeshift.</td>
<td>WARNING</td>
<td>System time has gone backwards.</td>
</tr>
<tr>
<td>port.status.</td>
<td>INFORMATIONAL</td>
<td>This eventType is not used directly. However, you can use this when retrieving Events by type to retrieve all port.status eventTypes.</td>
</tr>
<tr>
<td>port.status.down.</td>
<td>INFORMATIONAL</td>
<td>Ethernet port is down.</td>
</tr>
<tr>
<td>port.status.up.</td>
<td>INFORMATIONAL</td>
<td>Ethernet port is up.</td>
</tr>
</tbody>
</table>

### File Server Events

<table>
<thead>
<tr>
<th>eventType Value</th>
<th>Event Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fileserver.error.</td>
<td>CRITICAL</td>
<td>A critical fileserver error.</td>
</tr>
<tr>
<td>fileserver.invalid.exports.</td>
<td>CRITICAL</td>
<td>Invalid exports exist.</td>
</tr>
</tbody>
</table>

### File System Events

<table>
<thead>
<tr>
<th>eventType Value</th>
<th>Event Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filesystem.missing.</td>
<td>WARNING</td>
<td>File system is missing.</td>
</tr>
<tr>
<td>filesystem.size.mismatch.</td>
<td>CRITICAL</td>
<td>File system is larger than its underlying storage device.</td>
</tr>
</tbody>
</table>

### Repository Events

<table>
<thead>
<tr>
<th>eventType Value</th>
<th>Event Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>repository.empty.</td>
<td>CRITICAL</td>
<td>Repository is empty.</td>
</tr>
<tr>
<td>repository.error.</td>
<td>CRITICAL</td>
<td>There is a repository error.</td>
</tr>
<tr>
<td>repository.missing.</td>
<td>WARNING</td>
<td>Repository is missing.</td>
</tr>
<tr>
<td>repository.unmounted.</td>
<td>CRITICAL</td>
<td>Repository is unmounted</td>
</tr>
</tbody>
</table>

### Storage Device Events

<table>
<thead>
<tr>
<th>eventType Value</th>
<th>Event Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>storage.device.</td>
<td>INFORMATIONAL</td>
<td>Storage device is offline.</td>
</tr>
<tr>
<td>storage.device.offline.</td>
<td>WARNING</td>
<td>Storage device is offline.</td>
</tr>
<tr>
<td>storage.device.online.</td>
<td>INFORMATIONAL</td>
<td>Storage device is online.</td>
</tr>
<tr>
<td>storage.device.off.path.</td>
<td>WARNING</td>
<td>At least one path is missing for storage element</td>
</tr>
</tbody>
</table>

### Manager Events

<table>
<thead>
<tr>
<th>eventType Value</th>
<th>Event Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mgr.db.scan.error.</td>
<td>CRITICAL</td>
<td>Database scan error.</td>
</tr>
<tr>
<td>mgr.db.scan.ok.</td>
<td>INFORMATIONAL</td>
<td>Database scan succeeded.</td>
</tr>
<tr>
<td>mgr.db.backup.fail.</td>
<td>WARNING</td>
<td>Database backup failed.</td>
</tr>
<tr>
<td>mgr.db.timeshift.</td>
<td>WARNING</td>
<td>System time has gone backwards.</td>
</tr>
</tbody>
</table>

### Port Events

<table>
<thead>
<tr>
<th>eventType Value</th>
<th>Event Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>port.status.</td>
<td>INFORMATIONAL</td>
<td>This eventType is not used directly. However, you can use this when retrieving Events by type to retrieve all port.status eventTypes.</td>
</tr>
<tr>
<td>port.status.down.</td>
<td>INFORMATIONAL</td>
<td>Ethernet port is down.</td>
</tr>
<tr>
<td>port.status.up.</td>
<td>INFORMATIONAL</td>
<td>Ethernet port is up.</td>
</tr>
</tbody>
</table>

### Path Events
### Examples

<table>
<thead>
<tr>
<th>eventType Value</th>
<th>Event Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>path.status.</td>
<td>INFORMATIONAL</td>
<td>This eventType is not used directly. However, you can use this when retrieving Events by type to retrieve all path.status eventTypes.</td>
</tr>
<tr>
<td>path.status.down.</td>
<td>INFORMATIONAL</td>
<td>Storage path status is down.</td>
</tr>
<tr>
<td>path.status.up.</td>
<td>INFORMATIONAL</td>
<td>Storage path status is up.</td>
</tr>
<tr>
<td>path.missing.storage.device.</td>
<td>WARNING</td>
<td>Storage path is missing a storage device.</td>
</tr>
</tbody>
</table>

**Volume Events**

<table>
<thead>
<tr>
<th>eventType Value</th>
<th>Event Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>volume.group.missing.</td>
<td>CRITICAL</td>
<td>A volume group is missing.</td>
</tr>
</tbody>
</table>

**Object Events**

<table>
<thead>
<tr>
<th>eventType Value</th>
<th>Event Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object.needs.refresh.</td>
<td>INFORMATIONAL</td>
<td>Object needs to be refreshed to obtain valid object information.</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.140 Listing all events**

```
OVM> getEventListByQuery
```

**Example A.141 Listing all events for specified Oracle VM Servers**

```
OVM> getEventListByQuery objType=Server objIds=00:e0:81:4d:41:05:00:e0:81:4d:40:de:00:e0:81:4d, 00:e0:81:4d:41:59:00:e0:81:4d:41:2c:00:e0:81:4d
```

**Example A.142 Listing events types for Oracle VM Servers**

```
OVM> getEventListByQuery eventEqualityType=LIKE eventTypes=server.disconnected.
```

**Example A.143 Listing high priority events for Oracle VM Servers**

```
OVM> getEventListByQuery severity=WARNING severityEqualityType=GTE \
    eventTypes=server.disconnected. eventEqualityType=LIKE
```

**See Also**

- Section A.175, “showobjtypes”
- Section A.113, “getEvents”
- Section A.114, “getEventsForObject”
- Section A.2, “ackEvent”

**A.113 getEvents**

Lists the events for an object.

**Syntax**

```plaintext
getEvents [[objType=value][objId=value]][severity={UNKNOWN|CRITICAL|WARNING|INFORMATIONAL}][acknowledged={Yes|No}][startTime=value][endTime=value][amount=value]
```
### Description

This command lists the event for an object. To acknowledge an event, use the `ackEvent` command.

### Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>objType=value</code></td>
<td>The object type for which to list the events. The value of <code>objType</code> is an object such as <code>FileServer</code>, <code>Repository</code>, or <code>Server</code>. To get a list of all object types, use the <code>showobjt types</code>. The value for <code>objType</code> is case insensitive. This is an optional parameter and must be used with the <code>objId</code> option.</td>
</tr>
<tr>
<td><code>objId=value</code></td>
<td>The name or ID of the object. This is an optional parameter and must be used with the <code>objType</code> option.</td>
</tr>
<tr>
<td>`severity={UNKNOWN</td>
<td>CRITICAL</td>
</tr>
<tr>
<td>`acknowledged={Yes</td>
<td>No}`</td>
</tr>
<tr>
<td><code>startTime=value</code></td>
<td>The start date and time from which to list events. The format to use is &quot;MM-dd-yyyy HH:mm:ss&quot;. This is an optional parameter.</td>
</tr>
<tr>
<td><code>endTime=value</code></td>
<td>The end date and time from which to list events. The format to use is &quot;MM-dd-yyyy HH:mm:ss&quot;. This is an optional parameter.</td>
</tr>
<tr>
<td><code>amount=value</code></td>
<td>The maximum number of events to list. This is an optional parameter.</td>
</tr>
</tbody>
</table>

### Examples

**Example A.144 Listing all events**

```
OVM> getEvents
```

**Example A.145 Listing all critical error events**

```
OVM> getEvents severity=CRITICAL
```
Example A.146 Listing Oracle VM Server events using all options

OVM> getEvents objType=Server objId=00:e0:81:4d:40:c6:00:e0:81:4d:40:c7:ff:ff:ff:ff \  
    severity=UNKNOWN acknowledged=Yes startTime="05-20-2013 00:00:00" \  
    endTime="05-21-2013 23:59:00" amount=100

Example A.147 Listing all unacknowledged critical error events for a virtual machine

OVM> getEvents objType=Vm objId=0004fb0001400003f45fc117b56c135 severity=CRITICAL \  
    acknowledged=No

See Also

- Section A.175, “showobjtypes”
- Section A.114, “getEventsForObject”
- Section A.112, “getEventListByQuery”
- Section A.2, “ackEvent”

A.114 getEventsForObject

Lists the events for an object.

Syntax

getEventsForObject objType=value objId=value

Description

This command lists the events for an object. To acknowledge an event, use the ackEvent command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>objType=value</td>
<td>The object type for which to list the events. The value of objType is an object such as FileServer, Repository, or Server. To get a list of all object types, use the showobjtypes command. The value for objType is case insensitive.</td>
</tr>
<tr>
<td>objId=value</td>
<td>The name or ID of the object.</td>
</tr>
</tbody>
</table>

Examples

Example A.148 Listing events for an Oracle VM Server

OVM> getEventsForObject objType=Server objId=MyServer

See Also

- Section A.175, “showobjtypes”
- Section A.113, “getEvents”
- Section A.112, “getEventListByQuery”
A.115 getJobs

Lists jobs.

Syntax

getJobs [startTime=value][ endTime=value][ amount=value]

Description

This command lists all jobs, or jobs within a date range. Although none of the options are mandatory, you must supply at least one option.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startTime</td>
<td>The start date and time from which to list jobs. The format to use is &quot;MM-dd-yyyy HH:mm&quot;.</td>
</tr>
<tr>
<td>endTime</td>
<td>The end date and time from which to list jobs. The format to use is &quot;MM-dd-yyyy HH:mm&quot;.</td>
</tr>
<tr>
<td>amount</td>
<td>The number of jobs to list. For example, if you specify amount=2, the command returns the latest two jobs.</td>
</tr>
</tbody>
</table>

Examples

Example A.149 Listing jobs in a date range

OVM> getJobs startTime="07-20-2012 12:00" endTime="07-22-2012 24:00"

See Also

• Section A.119, “getQueuedJobInfo”
• Section A.110, “getDebugTranscript”
• Section A.1, “abort Job”
• Section A.138, “list”
• Section A.171, “show”

A.116 getLatestStat

Lists the latest value of a statistic type for an object.

Syntax

getLatestStat objType= {Server | Vm | FileSystem} objId=value statType= {CPU_UTILIZATION | CPU_COUNT | MEMORY_USED | FREE_MEMORY | MEMORY_UTILIZATION |
**Description**

This command lists the latest value of a statistic type for an object. Statistics are held for a limited amount of time and there may be no statistics of the particular type that occurred.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>objType</code>={Server</td>
<td>Vm</td>
</tr>
<tr>
<td><code>objId=value</code></td>
<td>The name or ID of the object for which statistics should be listed. This must be the ID of a Server, Vm or FileSystem object.</td>
</tr>
</tbody>
</table>
| `statType`={CPU_UTILIZATION|CPU_COUNT|MEMORY_USED|FREE_MEMORY|MEMORY_UTILIZATION|FILE_SYSTEM_SPACE_FREE|FILE_SYSTEM_SPACE_UTILIZATION|FILE_SYSTEM_SPACE_TOTAL|FILE_SYSTEM_TOTAL_FILES_SIZE} | The type of statistic to list:  
  - `CPU_UTILIZATION`: The percentage of the total CPU power being used by the object  
  - `CPU_COUNT`: The number of CPUs.  
  - `MEMORY_USED`: The amount of memory, in MBs, used by the object.  
  - `FREE_MEMORY`: The amount of memory, in MBs, available on the server.  
  - `MEMORY_UTILIZATION`: The percentage of the total memory being used by the object.  
  - `FILE_SYSTEM_SPACE_FREE`: The amount of file system space, in GiB, that is not currently used.  
  - `FILE_SYSTEM_SPACE_UTILIZATION`: The percentage of the total file system space currently in use.  
  - `FILE_SYSTEM_SPACE_TOTAL`: The total size of the file system, in GiB.  
  - `FILE_SYSTEM_TOTAL_FILES_SIZE`: The total of all the maximum sizes of all virtual disks on the file system, in GiB. |

The `FILE_SYSTEM_SPACE_FREE`, `FILE_SYSTEM_SPACE_UTILIZATION`, `FILE_SYSTEM_SPACE_TOTAL`, and `FILE_SYSTEM_TOTAL_FILES_SIZE` parameters are supported for the FileSystem object type only.
Examples

Example A.150 Listing latest statistics for an Oracle VM Server

```
OVM> getLatestStat objType=Server objId=00:e0:81:4d:41:05:00:e0:81:4d:40:de:00:e0:81:4d \
statType=MEMORY_USED
```

See Also

- Section A.117, “getLatestStatForList”
- Section A.108, “getAverageStat”
- Section A.121, “getStatList”
- Section A.122, “getStatListByQuery”
- Section A.120, “getStatsConfig”
- Section A.169, “setStatsConfig”
- Section A.138, “list”
- Section A.171, “show”

A.117 getLatestStatForList

Lists the latest value of a statistic type for objects.

Syntax

```
getLatestStatForList objType={Server | Vm | FileSystem} ids=value statType= \
{CPU_UTILIZATION | CPU_COUNT | MEMORY_USED | FREE_MEMORY | MEMORY_UTILIZATION | 
FILE_SYSTEM_SPACE_FREE | FILE_SYSTEM_SPACE_UTILIZATION | FILE_SYSTEM_SPACE_TOTAL }
```

Description

This command lists the latest value of a statistic type for objects.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>objType={Server</td>
<td>Vm</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ids=value</td>
<td>A comma separated list of the IDs of the objects for which statistics should be listed. This must contain the IDs of Server, Vm or FileSystem objects.</td>
</tr>
<tr>
<td>statType={</td>
<td>The type of statistic to list:</td>
</tr>
<tr>
<td>CPU_UTILIZATION</td>
<td>• <strong>CPU_UTILIZATION</strong>: The percentage of the total CPU power being used by the object</td>
</tr>
<tr>
<td>CPU_COUNT</td>
<td>• <strong>CPU_COUNT</strong>: The number of CPUs.</td>
</tr>
<tr>
<td>MEMORY_USED</td>
<td>• <strong>MEMORY_USED</strong>: The amount of memory, in MBs, used by the object.</td>
</tr>
<tr>
<td>FREE_MEMORY</td>
<td>• <strong>FREE_MEMORY</strong>: The amount of memory, in MBs, available on the server.</td>
</tr>
<tr>
<td>MEMORY_UTILIZATION</td>
<td>• <strong>MEMORY_UTILIZATION</strong>: The percentage of the total memory being used by the object</td>
</tr>
<tr>
<td>FILE_SYSTEM_SPACE_FREE</td>
<td>• <strong>FILE_SYSTEM_SPACE_FREE</strong>: The amount of file system space, in GiB, that is not currently used.</td>
</tr>
<tr>
<td>FILE_SYSTEM_SPACE_UTILIZATION</td>
<td>The percentage of the total file system space currently in use.</td>
</tr>
<tr>
<td>FILE_SYSTEM_SPACE_TOTAL</td>
<td>The total size of the file system, in GiB.</td>
</tr>
</tbody>
</table>

The **FILE_SYSTEM_SPACE_FREE**, **FILE_SYSTEM_SPACE_UTILIZATION**, and **FILE_SYSTEM_SPACE_TOTAL** parameters are supported for the FileSystem object type only.

The **CPU_COUNT** parameter is not supported by the Server object type.

An error is returned when an incompatible object type and parameter are requested.

A value of 0 may be returned for any parameter if there are no statistics available for the object on Oracle VM Manager.

---

**Examples**

**Example A.151 Listing latest statistics for Oracle VM Servers**

```
OVM> getLatestStatForList objType=Server \
   ids=00:e0:81:4d:41:05:00:e0:81:4d:40:de:00:e0:81:4d,00:e0:81:4d:29:ee:00:e0:81:4d \
   statType=CPU_UTILIZATION
```

**See Also**

- Section A.116, "getLatestStat"
- Section A.108, "getAverageStat"


• Section A.121, “getStatList”
• Section A.122, “getStatListByQuery”
• Section A.120, “getStatsConfig”
• Section A.169, “setStatsConfig”
• Section A.138, “list”
• Section A.171, “show”

A.118 getManagerTime

Displays the time for Oracle VM Manager.

Syntax

getManagerTime

Description

This command displays the time for Oracle VM Manager.

Options

This command does not take any arguments or provide any options.

Examples

Example A.152 Display the Oracle VM Manager time

OVM> getManagerTime

A.119 getQueuedJobInfo

Lists information about a queued job.

Syntax

getQueuedJobInfo Job instance

Where instance is:

{ id=value | name=value } 

Description

This command lists information about a queued job.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>
Examples

Example A.153 Listing information about a queued job

```
OVM> getQueuedJobInfo Job id=1394647459371
```

See Also

- Section A.115, "getJobs"
- Section A.110, "getDebugTranscript"
- Section A.1, "abort Job"
- Section A.138, "list"
- Section A.171, "show"

A.120 getStatsConfig

Shows the configuration for the statistics displayed in Oracle VM Manager.

Syntax

```
getStatsConfig
```

Description

This command shows the configuration for the statistics displayed in Oracle VM Manager.

Options

This command does not take any arguments or provide any options.

Examples

Example A.154 Showing the Oracle VM Manager statistics configuration

```
OVM> getStatsConfig
```

See Also

- Section A.169, "setStatsConfig"
- Section A.108, "getAverageStat"
- Section A.116, "getLatestStat"
- Section A.117, "getLatestStatForList"
- Section A.108, "getAverageStat"

A.121 getStatList

Lists statistic types for an object.

Syntax

```
getStatList objType={ Server | Vm | FileSystem } objId=value statType={
    CPU_UTILIZATION | CPU_COUNT | MEMORY_USED | FREE_MEMORY | MEMORY_UTILIZATION |
```
FILE_SYSTEM_SPACE_FREE | FILE_SYSTEM_SPACE_UTILIZATION | FILE_SYSTEM_SPACE_TOTAL }  
startTime=value [ endTime=value ]

Description

This command lists statistics, that are stored within Oracle VM Manager, for an object.

Note that statistics are only stored for the length of time specified for the value of the statistics holdTime, as described in Section A.169, “setStatsConfig”. The default holdTime is 60 minutes. Every 60 minutes, an archive manager runs to trim the statistics to the value specified for the holdTime. Therefore, just prior to cleanup it is possible that statistics are available for the holdTime value in addition to 60 minutes. For example, if the holdTime is specified as the default 60 minutes, then it is possible that statistics of up to 120 minutes old may be returned if the command is run at exactly the same time that the archive manager runs.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>objType={ Server</td>
<td>Vm</td>
</tr>
<tr>
<td>objId=value</td>
<td>The name or ID of the object for which statistics should be listed. This must be the ID of either a Server, Vm or FileSystem object.</td>
</tr>
</tbody>
</table>
| statType={ CPU_UTILIZATION | CPU_COUNT | MEMORY_USED | FREE_MEMORY | MEMORY_UTILIZATION | FILE_SYSTEM_SPACE_FREE | FILE_SYSTEM_SPACE_UTILIZATION | FILE_SYSTEM_SPACE_TOTAL } | The type of statistic to list:
  - **CPU_UTILIZATION**: The percentage of the total CPU power being used by the object
  - **CPU_COUNT**: The number of CPUs.
  - **MEMORY_USED**: The amount of memory, in MBs, used by the object.
  - **FREE_MEMORY**: The amount of memory, in MBs, available on the server.
  - **MEMORY_UTILIZATION**: The percentage of the total memory being used by the object. This option applies only to objects associated with a server.
  - **FILE_SYSTEM_SPACE_FREE**: The amount of file system space, in GiB, that is not currently used.
  - **FILE_SYSTEM_SPACE_UTILIZATION**: The percentage of the total file system space currently in use.
  - **FILE_SYSTEM_SPACE_TOTAL**: The total size of the file system, in GiB. |
Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>FILE_SYSTEM_SPACE_TOTAL</strong> parameters are supported for the FileSystem object type only.</td>
</tr>
<tr>
<td></td>
<td>The <strong>CPU_COUNT</strong> parameter is not supported by the Server object type.</td>
</tr>
<tr>
<td></td>
<td>An error is returned when an incompatible object type and parameter are requested.</td>
</tr>
<tr>
<td></td>
<td>A value of <strong>0</strong> may be returned for any parameter if there are no statistics available for the object on Oracle VM Manager in the time frame when the statistics were requested.</td>
</tr>
<tr>
<td><strong>startTime=value</strong></td>
<td>The start date and time from which to list the average statistic. The format to use is &quot;MM-dd-yyyy HH:mm:ss&quot;.</td>
</tr>
<tr>
<td></td>
<td>The value must not be later than the current time or after the value of <strong>endTime</strong>.</td>
</tr>
<tr>
<td><strong>endTime=value</strong></td>
<td>The end date and time from which to list the average statistic. The format to use is &quot;MM-dd-yyyy HH:mm:ss&quot;.</td>
</tr>
<tr>
<td></td>
<td>This is an optional parameter.</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.155 Listing statistics for an Oracle VM Server**

```
OVM> getStatList objType=Server objId=00:e0:81:4d:41:05:00:e0:81:4d:40:de:00:e0:81:4d statType=MEMORY_USED startTime="08-20-2014 00:00"
```

**See Also**

- Section A.122, "getStatListByQuery"
- Section A.116, "getLatestStat"
- Section A.117, "getLatestStatForList"
- Section A.108, "getAverageStat"
- Section A.120, "getStatsConfig"
- Section A.169, "setStatsConfig"
- Section A.138, "list"
- Section A.171, "show"

**A.122 getStatListByQuery**

Lists statistic types for an object using a query to narrow results.

**Syntax**

```
getStatListByQuery objType={Server | Vm | FileSystem} objIds=value statType={CPU_UTILIZATION | CPU_COUNT | MEMORY_USED | FREE_MEMORY | MEMORY_UTILIZATION |}
```

189
**Description**

This command lists statistics for an object using a query to refine results.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>objType</strong></td>
<td>The object type for which to list the statistics. The value of <strong>objType</strong> may be Server, Vm, or FileSystem. The value for <strong>objType</strong> is case insensitive.</td>
</tr>
<tr>
<td><strong>objIds</strong></td>
<td>The name or ID of the objects for which statistics should be listed. This must be the ID of a Server, Vm, or FileSystem object. To enter one or more objects, use a comma separated list, for example: objIds=UUID1,UUID2,UUID3</td>
</tr>
<tr>
<td><strong>statType</strong></td>
<td>The type of statistic to list:</td>
</tr>
<tr>
<td></td>
<td>• CPU_UTILIZATION: The percentage of the total CPU power being used by the object</td>
</tr>
<tr>
<td></td>
<td>• CPU_COUNT: The number of CPUs.</td>
</tr>
<tr>
<td></td>
<td>• MEMORY_USED: The amount of memory, in MBs, used by the object.</td>
</tr>
<tr>
<td></td>
<td>• FREE_MEMORY: The amount of memory, in MBs, available on the server.</td>
</tr>
<tr>
<td></td>
<td>• MEMORY_UTILIZATION: The percentage of the total memory being used by the object.</td>
</tr>
<tr>
<td></td>
<td>• FILE_SYSTEM_SPACE_FREE: The amount of file system space, in GiB, that is not currently used.</td>
</tr>
<tr>
<td></td>
<td>• FILE_SYSTEM_SPACE_UTILIZATION: The percentage of the total file system space currently in use.</td>
</tr>
<tr>
<td></td>
<td>• FILE_SYSTEM_SPACE_TOTAL: The total size of the file system, in GiB.</td>
</tr>
</tbody>
</table>

The **FILE_SYSTEM_SPACE_FREE**, **FILE_SYSTEM_SPACE_UTILIZATION**, and **FILE_SYSTEM_SPACE_TOTAL** parameters are supported for the FileSystem object type only.

The **CPU_COUNT** parameter is not supported by the Server object type.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equalityType={LT</td>
<td>LTE</td>
</tr>
<tr>
<td>statValue=value</td>
<td>The value to use with the equalityType option. Use this option to search and filter the results. For example, when you want to find when a server is under 20% CPU utilization, use: equalityType=LT statValue=20</td>
</tr>
<tr>
<td>startTime=value</td>
<td>The start date and time from which to list statistics. The format to use is &quot;MM-dd-yyyy HH:mm:ss&quot;. The value must not be later than the current time or after the value of endTime.</td>
</tr>
<tr>
<td>endTime=value</td>
<td>The end date and time from which to list statistics. The format to use is &quot;MM-dd-yyyy HH:mm:ss&quot;. This is an optional parameter.</td>
</tr>
<tr>
<td>maxResults=value</td>
<td>The maximum number of statistics to list. This is an optional parameter.</td>
</tr>
</tbody>
</table>
Examples

Example A.156 Listing statistics for an Oracle VM Server using a query to filter results

```
OVM> getStatListByQuery objType=Server objIds=00:e0:81:4d:41:05:00:e0:81:4d:40:de:00:e0:81:4d \  
    statType=CPU_UTILIZATION equalityType=GT statValue=80
```

Example A.157 Listing statistics during a date range

```
OVM> getStatListByQuery objType=Server objIds=00:e0:81:4d:5f:2f:00:e0:81:4d:29:ee:00:e0:81:4d \  
    statType=CPU_UTILIZATION equalityType=GT statValue=80 startTime="02-01-2015 00:00:00" \  
    endTime="02-09-2015 23:59:00" maxResults=10
```

Example A.158 Listing statistics for a virtual machine using a query to filter results

```
OVM> getStatListByQuery objType=Vm objIds=0004fb000006000044f5ee8585b1d59c statType=MEMORY_USED \ 
    equalityType=GT statValue=2048
```

See Also

- Section A.121, “getStatList”
- Section A.116, “getLatestStat”
- Section A.117, “getLatestStatForList”
- Section A.108, “getAverageStat”
- Section A.120, “getStatsConfig”
- Section A.169, “setStatsConfig”
- Section A.138, “list”
- Section A.171, “show”

A.123 getTriageEvent

Lists the highest severity event for an object.

Syntax

```
getTriageEvent objType=value objId=value
```

Description

This command lists the highest severity event for an object. To acknowledge an event, use the `ackEvent` command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>objType=value</code></td>
<td>The object type for which to list the event. The value of <code>objType</code> is an object such as <code>FileServer</code>, <code>Repository</code>, or <code>Server</code>. To get a list of all object types, use the <code>showObjTypes</code> command. The value for <code>objType</code> is case insensitive.</td>
</tr>
</tbody>
</table>
Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>objId=value</td>
<td>The name or ID of the object.</td>
</tr>
</tbody>
</table>

Examples

Example A.159 Listing the highest severity event for an Oracle VM Server

```
OVM> getTriageEvent objType=Server objId=MyServer
```

See Also

- Section A.124, “getTriageEventSeverityList”
- Section A.175, “showobjtypes”
- Section A.113, “getEvents”
- Section A.114, “getEventsForObject”
- Section A.2, “ackEvent”

A.124 getTriageEventSeverityList

Lists the highest severity event for one or more objects.

Syntax

```
getTriageEventSeverityList objList=value
```

Description

This command lists the highest severity event for one or more objects. To acknowledge an event, use the `ackEvent` command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>objList=value</td>
<td>A comma delimited list of object IDs or names for which to list the event. The value of objList is a list of object IDs or names for objects such as FileServer, Repository, or Server. The value for objType is case insensitive.</td>
</tr>
</tbody>
</table>

Examples

Example A.160 Listing the highest severity event for a list of objects

```
OVM> getTriageEventSeverityList objList=0004fb00014000121afbf823ad987,\ 
    0004fb000060004df5e585b1d59c,MyRepository
```

See Also

- Section A.123, “getTriageEvent”
A.125 getVmCfgFileContent

Shows the contents of a virtual machine configuration file.

Syntax

getVmCfgFileContent Vm instance

Where instance is:

{ id=value | name=value }

Description

This command shows the contents of a virtual machine configuration file (vm.cfg).

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

Example A.161 Showing the contents of a virtual machine configuration file

OVM> getVmCfgFileContent Vm name=MyVM

See Also

- Section A.138, “list”
- Section A.171, “show”

A.126 getVmOsTypes

Shows the virtual machine operating system types.

Syntax

getVmOsTypes

Description

This command shows the virtual machine operating types used when creating or editing a virtual machine.
Options

This command does not take any arguments or provide any options.

Examples

Example A.162 Showing virtual machine operating system types

OVM> getVmOsTypes

See Also

• Section A.53, "create Vm"
• Section A.95, "edit Vm"

A.127 getVmReceivedMessages

Lists the key/value pair messages received from a running virtual machine.

Syntax

getVmReceivedMessages Vm instance

Where instance is:

{ id=value | name=value }

Description

This command lists the key/value pair messages received from a running virtual machine.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

Example A.163 Listing messages received from a virtual machine

OVM> getVmReceivedMessages Vm name=MyVm

See Also

• Section A.164, “sendVmMessage”
• Section A.128, “getVmSentMessages”
• Section A.22, “clearVmRcvdMessage”
• Section A.23, “clearVmSentMessage”
A.128 getVmSentMessages

Lists the key/value pair messages sent to a running virtual machine.

Syntax

```
getVmSentMessages Vm instance
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command lists the key/value pair messages sent to a running virtual machine.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

Examples

Example A.164 Listing messages sent to a virtual machine

```
OVM> getVmSentMessages Vm name=MyVm
```

See Also

- Section A.164, “sendVmMessage”
- Section A.127, “getVmReceivedMessages”
- Section A.22, “clearVmRcvdMessage”
- Section A.23, “clearVmSentMessage”
- Section A.21, “clearVmAllSentMessages”
- Section A.20, “clearVmAllRcvdMessages”

A.129 getVnicMacAddrRange

Displays the range of MAC addresses that are available to VNICS.

Syntax

```
getVnicMacAddrRange
```
**Description**

This command displays the range of MAC addresses that can be used when creating a VNIC. To set the MAC address range, use the `setVnicMacAddrRange` command.

**Options**

This command does not take any arguments or provide any options.

**Examples**

Example A.165 Displaying the VNIC MAC Address Range

```
OVM> getVnicMacAddrRange
```

**See Also**

- Section A.170, “setVnicMacAddrRange”
- Section A.58, “create Vnic”
- Section A.100, “edit Vnic”
- Section A.17, “add Vnic”
- Section A.160, “remove Vnic”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

---

**A.130 help**

Provides a list of commonly used commands and their syntax.

**Syntax**

```
help
```

**Description**

This command provides a list of commonly used commands and their syntax. It groups the commands according to how they are generally used.

**Options**

This command does not take any arguments or provide any options.

**Examples**

Example A.166 Using the Help Command

```
OVM> help
```
A.131 importAssembly (Deprecated)

Imports and adds an assembly to a storage repository.

**Important**

This command has been deprecated. Instead, use the importVirtualAppliance command.

### Syntax

```
importAssembly Repository instance url=value [proxy=value]
```

Where `instance` is:

```
{ id=value | name=value }
```

### Description

This command imports and adds an assembly file to a storage repository. The imported assembly is unpacked and each virtual machine is contained within an AssemblyVm object. The AssemblyVm objects are created in the same storage repository as the original assembly file. Use the `list AssemblyVm` command to find the name and ID of the new AssemblyVm objects, then use the `createVmFromAssembly (Deprecated)` command to create a virtual machine templates from each AssemblyVm object.

### Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository</td>
<td>The storage repository in which to import the assembly.</td>
</tr>
<tr>
<td>instance</td>
<td>The URL of the assembly. Note that if you quote this argument, the forward slashes in the URL should be escaped. This is illustrated in the example.</td>
</tr>
<tr>
<td>url=value</td>
<td>The IP address or hostname of a proxy server to use when importing the assembly.</td>
</tr>
<tr>
<td>proxy=value</td>
<td>The instance of the object using either the id or name option, for example name=MyServer.</td>
</tr>
</tbody>
</table>

### Examples

**Example A.167 Importing an assembly to a storage repository**

```
OVM> importAssembly Repository name=MyRepository url="http:///example.com//myassembly.ova"
```
See Also

- Section A.31, “createVmFromAssembly (Deprecated)"
- Section A.63, “edit Assembly (Deprecated)"
- Section A.64, “edit AssemblyVirtualDisk (Deprecated)"
- Section A.65, “edit AssemblyVm (Deprecated)"
- Section A.111, “getDescriptor”
- Section A.59, “delete”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”

A.132 importTemplate

Imports and adds a virtual machine template to a storage repository.

Syntax

importTemplate Repository instance url=value[ proxy=value]

Where instance is:

{ id=value | name=value }

Description

This command imports and adds a virtual machine template to a storage repository.

Virtual machines and virtual machine templates are treated the same in the CLI, so many of the commands you use to manage templates are handled by the same commands as managing virtual machines, for example, to list templates, use the list Vm command, and to delete a template, use the delete Vm command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository instance</td>
<td>The storage repository in which to import the virtual machine template.</td>
</tr>
<tr>
<td>url=value</td>
<td>The URL of the virtual machine template. Note that if you quote this argument, the forward slashes in the URL should be escaped. This is illustrated in the example.</td>
</tr>
<tr>
<td>proxy=value</td>
<td>The IP address or hostname of a proxy server to use when importing the template.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>
Examples

Example A.168 Importing a virtual machine template to a storage repository

```bash
OVMM> importTemplate Repository name=MyRepository url="http://example.com/mytemplate.tgz"
```

See Also

- Section A.136, “importVirtualMachine”
- Section A.24, “clone Vm”
- Section A.142, “refresh”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

A.133 importVirtualAppliance

Imports and adds a virtual appliance to a storage repository.

Syntax

```
importVirtualAppliance Repository instance url=value [proxy=value]
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command imports and adds a virtual appliance to a storage repository. The imported virtual appliance is unpacked and each virtual machine is contained within an VirtualApplianceVm object. The VirtualApplianceVm objects are created in the same storage repository as the original virtual appliance. Use the `list VirtualApplianceVm` command to find the name and ID of the new VirtualApplianceVm objects, then use the `createVmFromVirtualAppliance` command to create virtual machines from each VirtualApplianceVm object.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository instance</td>
<td>The storage repository in which to import the virtual appliance.</td>
</tr>
<tr>
<td>url=value</td>
<td>The URL of the virtual appliance. Note that if you quote this argument, the forward slashes in the URL should be escaped. This is illustrated in the example.</td>
</tr>
<tr>
<td>proxy=value</td>
<td>The IP address or hostname of a proxy server to use when importing the virtual appliance.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>
Examples

Example A.169 Importing a virtual appliance to a storage repository

```
OVM> importVirtualAppliance Repository name=MyRepository url="http://example.com/myvirtualappliance.ova"
```

See Also

- Section A.32, “createVmFromVirtualAppliance”
- Section A.33, “createVmFromVirtualApplianceVm”
- Section A.106, “exportVirtualAppliance”
- Section A.89, “edit VirtualAppliance”
- Section A.90, “edit VirtualApplianceVirtualDisk”
- Section A.91, “edit VirtualApplianceVm”
- Section A.111, “getDescriptor”
- Section A.59, “delete”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”

A.134 importVirtualCdrom

Imports and adds a virtual CDROM/ISO file to a storage repository.

Syntax

```
importVirtualCdrom Repository instance url=value [ proxy=value ]
```

Where `instance` is:

{ id=value | name=value }

Description

This command imports and adds a virtual CDROM/ISO file to a storage repository.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Repository instance</code></td>
<td>The storage repository in which to import the virtual CDROM/ISO file.</td>
</tr>
<tr>
<td><code>url=value</code></td>
<td>The URL of the virtual CDROM/ISO file. Note that if you quote this argument, the forward slashes in the URL should be escaped. This is illustrated in the example.</td>
</tr>
</tbody>
</table>
### Option and Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>proxy=value</code></td>
<td>The IP address or hostname of a proxy server to use when importing the virtual CDROM/ISO file.</td>
</tr>
<tr>
<td>{ <code>id=value</code></td>
<td><code>name=value</code> }</td>
</tr>
</tbody>
</table>

### Examples

#### Example A.170 Importing a virtual CDROM/ISO file to a storage repository

```shell
OVM> importVirtualCdrom Repository name=MyRepository url="http://example.com//myiso.iso"
```

**Tip**

Note that, in the above example, forward slashes are used to escape the forward slashes that appear in the URL. Hence the doubling up of forward slashes.

### See Also

- Section A.92, “edit VirtualCdrom”
- Section A.25, “cloneCdToRepo”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

### A.135 importVirtualDisk

Imports and adds a virtual disk file to a storage repository.

#### Syntax

```
importVirtualDisk Repository instance url=value [proxy=value]
```

Where `instance` is:

{ `id=value` | `name=value` }

#### Description

This command imports and adds a virtual disk file to a storage repository.

#### Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Repository instance</code></td>
<td>The storage repository in which to import the virtual disk file.</td>
</tr>
<tr>
<td><code>url=value</code></td>
<td>The URL of the virtual disk file. Note that if you quote this argument, the forward slashes in the URL should be escaped. This is illustrated in the example.</td>
</tr>
</tbody>
</table>
### Examples

**Example A.171 Importing a virtual disk file to a storage repository**

```bash
OVM> importVirtualDisk Repository name=MyRepository url="http:///example.com//myvdisk.img"
```

### See Also

- Section A.51, “create VirtualDisk”
- Section A.93, “edit VirtualDisk”
- Section A.59, “delete”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

### A.136 importVirtualMachine

Imports and adds a virtual machine to a storage repository.

**Syntax**

```
importVirtualMachine Repository instance url=value[ proxy=value]
```

**Where instance is:**

```
{ id=value | name=value }
```

**Description**

This command imports and adds a virtual machine to a storage repository. The virtual machine is placed in the **Unassigned Virtual Machines** folder in Oracle VM Manager. To deploy the virtual machine to an Oracle VM Server, use the `add Vm` command.

### Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Repository instance</em></td>
<td>The storage repository in which to import the virtual machine.</td>
</tr>
</tbody>
</table>
Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url=value</td>
<td>The URL of the virtual machine. To import a multi-file virtual machine, enter each URL in a comma separated list, for example: url=<a href="http://example.com/Sys.img,http://example.com/vm.cfg">http://example.com/Sys.img,http://example.com/vm.cfg</a></td>
</tr>
<tr>
<td>proxy=value</td>
<td>The IP address or hostname of a proxy server to use when importing the virtual machine.</td>
</tr>
<tr>
<td>{id=value</td>
<td>The instance of the object using either the id or name option, for example name=MyRepository.</td>
</tr>
</tbody>
</table>

Examples

Example A.172 Importing a virtual machine to a storage repository

OVM> importVirtualMachine Repository name=MyRepository url="http://example.com/mytemplate.tgz"

See Also

- Section A.16, “add Vm”
- Section A.132, “importTemplate”
- Section A.24, “clone Vm”
- Section A.142, “refresh”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

A.137 kill

Kills an Oracle VM Server or virtual machine.

Syntax

\[kill\{Server|Vm\} instance\]

Where \textit{instance} is:

\{id=value | name=value\}

Description

This command kills an Oracle VM Server or virtual machine.

Options

The following table shows the available options for this command.
Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Server</td>
<td>Vm}</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.173 Killing an Oracle VM Server

OVM> kill Server name=MyServer

Example A.174 Killing a virtual machine

OVM> kill Vm name=MyVM

See Also

- Section A.177, “start”
- Section A.162, “restart”
- Section A.178, “stop”

A.138 list

Lists all instances of an object.

Syntax

```
list {AccessGroup | AntiAffinityGroup | Assembly | AssemblyVirtualDisk | AssemblyVm | BondPort | ControlDomain | Cpu | CpuCompatibilityGroup | FileServer | FileServerPlugin | FileSystem | Job | Manager | Network | PeriodicTask | PhysicalDisk | Port | Repository | RepositoryExport | Server | ServerController | ServerPool | ServerPoolNetworkPolicy | ServerUpdateGroup | ServerUpdateRepository | StorageArray | StorageArrayPlugin | StorageInitiator | Tag | VirtualAppliance | VirtualApplianceVirtualDisk | VirtualApplianceVm | VirtualCdrom | VirtualDisk | VlanInterface | Vm | VmCloneCustomizer | VmCloneNetworkMapping | VmCloneStorageMapping | VmDiskMapping | Vnic | VolumeGroup }
```

Description

This command lists all instances of an object.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{AccessGroup</td>
<td>AntiAffinityGroup</td>
</tr>
<tr>
<td>Vm }</td>
<td>The Vm option lists both virtual machines and virtual machine templates. You can use the show command with either the</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>ControlDomain</td>
<td>Cpu</td>
</tr>
</tbody>
</table>

The VmDiskMapping option lists the disk mapping objects for both virtual machines and virtual machine templates.

**Important**

The Assembly, AssemblyVirtualDisk and AssemblyVm options have been deprecated. You should instead use VirtualAppliance, VirtualApplianceVirtualDisk, and VirtualApplianceVm, respectively.

### Examples

**Example A.175 Listing Oracle VM Servers**

```bash
OVMM> list Server
```

**Example A.176 Listing virtual machines and virtual machine templates**

```bash
OVMM> list Vm
```

**Example A.177 Listing networks**

```bash
OVMM> list Network
```

**Example A.178 Listing virtual machine and virtual machine templates disk mapping**

```bash
OVMM> list VmDiskMapping
```

### See Also

- Section A.171, “show”

### A.139 migrate Vm

Migrates a virtual machine.

#### Syntax

```
migrate Vm instance [destServer=value | destServerPool=value]
```
**Description**

This command migrates a virtual machine to an Oracle VM Server or server pool.

You can migrate a running virtual machine to an Oracle VM Server within the same server pool. To migrate the virtual machine to the **Unassigned Virtual Machines** folder (undeploy it), do not supply a destination.

It is not possible to migrate a stopped virtual machine using this command. To do this, you should use the remove Vm and add Vm commands instead.

To migrate a virtual machine, including the local storage, use the migrateWithLocalStorage Vm command.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![destServer=</td>
<td>destServerPool=] value</td>
</tr>
<tr>
<td>![id=] value</td>
<td>The instance of the object using either the id or name option, for example name=MyServer.</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.179 Migrating a virtual machine to an Oracle VM Server**

```
OVM> migrate Vm name=MyVM destServer=MyServer
```

**Example A.180 Migrating a virtual machine to a server pool**

```
OVM> migrate Vm name=MyVM destServerPool=MyServerPool
```

**Example A.181 Migrating a virtual machine to the Unassigned Virtual Machines folder**

```
OVM> migrate Vm name=MyVM
```

**See Also**

- Section A.140, “migrateWithLocalStorage Vm”
- Section A.16, “add Vm”
- Section A.159, “remove Vm”
- Section A.177, “start”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”
migrateWithLocalStorage Vm

A.140 migrateWithLocalStorage Vm

Migrates a virtual machine to another Oracle VM Server within the same server pool, and moves its local storage to another storage repository.

Syntax

```plaintext
migrateWithLocalStorage Vm instance destServer=value repository=value
```

Where `instance` is:

```plaintext
{id=value | name=value}
```

Description

This command migrates a virtual machine to another Oracle VM Server within the same server pool, and moves its local storage to another storage repository.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>destServer=value</td>
<td>The name or ID of the Oracle VM Server on which to migrate the virtual machine. The Oracle VM Server must be in the same server pool as the source Oracle VM Server.</td>
</tr>
<tr>
<td>repository=value</td>
<td>The name or ID of the storage repository on which to migrate the virtual machine local storage. Only repositories hosted on an OCFS2 file system may be used if you are moving the virtual machine storage of a running virtual machine. This command moves the virtual machine configuration and virtual disks to the target repository. However, the virtual machine configuration is moved only if it resides in a local repository, not in a shared repository.</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.182 Migrating a virtual machine to an Oracle VM Server, including local storage

```plaintext
OVM> migrateWithLocalStorage Vm name=MyVM destServer=MyServer repository=MyRepository
```

See Also

- Section A.139, “migrate Vm”
- Section A.16, “add Vm”
- Section A.159, “remove Vm”
- Section A.177, “start”
A.141 moveVmToRepository

Moves a virtual machine to a storage repository that has been defined within a clone customizer.

Syntax

```
moveVmToRepository Vm instance cloneCustomizer=value targetRepository=value
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command moves a virtual machine to a target storage repository. The target storage repository is set using a predefined clone customizer. Use the `create VmCloneCustomizer` command to create a clone customizer.

Any VmDiskMapping objects are renamed during the move operation. The move job copies any virtual disks to the target storage repository, and, consequently, the new virtual disks have new UUIDs. New VmDiskMapping objects are created to map the newly created virtual disks and added to the virtual machine. The old VmDiskMapping objects are then deleted. Deleting the VmDiskMapping object also causes any associated VmCloneStorageMapping objects to be deleted.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloneCustomizer=value</td>
<td>The name or ID of the clone customizer to use to move the virtual machine to a storage repository.</td>
</tr>
<tr>
<td>targetRepository=value</td>
<td>The name or ID of the storage repository to which the virtual machine is to be moved.</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

Example A.183 Moving a virtual machine to a storage repository

```
OVM> moveVmToRepository Vm name=MyVm cloneCustomizer=MyVMCloneCustomizer \
    targetRepository=MyRepository
```

See Also

- Section A.54, "create VmCloneCustomizer"
**A.142 refresh**

Refreshes configuration information about an object in Oracle VM Manager.

**Syntax**

```
refresh {AccessGroup|Assembly|FileServer|FileSystem|PhysicalDisk|Repository|Server|StorageArray|VirtualAppliance} instance
```

Where `instance` is:

```
{ id=value | name=value }
```

**Description**

This command reads the configuration information about the object and updates the Oracle VM Manager database repository.

When refreshing the file systems or a file server with non-uniform exports, all of the refresh servers for the file server must be available. If a refresh server is out of commission and you need to perform a refresh, you can remove that refresh server from the file server and add an alternate that has access to the same set of exports on the file server. For more information on uniform and non-uniform exports, please refer to [What are Uniform and Non-uniform Exports?](#) in the *Oracle VM Concepts Guide*.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{ AccessGroup</td>
<td>Assembly</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>
Examples

Example A.184 Refreshing a file server

```
OVM> refresh FileServer name=MyNFSServer
```

Example A.185 Refreshing a storage array

```
OVM> refresh StorageArray name=MyISCSIServer
```

Example A.186 Refreshing a physical disk

```
OVM> refresh PhysicalDisk id=0004fb0400000035c16ee4d58dc4d
```

Example A.187 Refreshing a file system

```
OVM> refresh FileSystem name="nfs on 10.172.76.125://mnt//vol1//repo01"
```

Example A.188 Refreshing a storage repository

```
OVM> refresh Repository name=MyRepository
```

See Also

- Section A.143, “refreshAll"
- Section A.144, “refreshStorageLayer”

A.143 refreshAll

RedisCOVERs all Oracle VM Server instances, file servers, and storage arrays.

Syntax

```
refreshAll
```

Description

This command redisCOVERs all Oracle VM Server instances, file servers, and storage arrays.

Important

The Refresh All function does not pick up the contents of file systems that have never been refreshed before. Furthermore, it does not refresh repositories that are not already presented on at least one server. It is important to keep this in mind if you have restored a configuration from a backup, since some items may not have been refreshed before at the time that the backup was created.

Options

This command does not take any arguments or provide any options.

Examples

Example A.189 Rediscovering Oracle VM Servers, file servers and storage arrays

```
OVM> refreshAll
```
See Also

- Section A.142, "refresh"
- Section A.144, "refreshStorageLayer"

### A.144 refreshStorageLayer

Refreshes the storage visible to an Oracle VM Server.

#### Syntax

```
refreshStorageLayer Server instance
```

Where `instance` is:

```
{ id=value | name=value }
```

#### Description

This command refreshes the storage visible to an Oracle VM Server. This command discovers:

- Exported NFS shares residing on any NFS file servers. The Oracle VM Server must be an admin server or refresh server.
- Presented physical disks on any iSCSI or unmanaged Fibre Channel storage arrays. The Oracle VM Server must be an admin server.

#### Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

#### Examples

**Example A.190 Refreshing Oracle VM Server storage**

```
OVM> refreshStorageLayer Server name=MyServer
```

See Also

- Section A.142, "refresh"
- Section A.143, "refreshAll"
- Section A.138, "list"
- Section A.171, “show”

### A.145 releaseOwnership

Releases ownership of an Oracle VM Server or storage repository.
Syntax

```
releaseOwnership {Repository | Server} instance
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command releases ownership of an Oracle VM Server or storage repository.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{Repository</td>
<td>Server}`</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Examples

**Example A.191 Releasing ownership of an Oracle VM Server**

```
OVM> releaseOwnership Server name=MyServer
```

**Example A.192 Releasing ownership of a storage repository**

```
OVM> releaseOwnership Repository name=MyRepository
```

See Also

- Section A.180, “takeOwnership”

A.146 removeAccessHost

Removes an access host from an ISCSI server.

Syntax

```
removeAccessHost StorageArray instance accessHost=value
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command removes an access host from an ISCSI server when using a storage array with `multipath` capability. At least one access host must be set. Multipath is not supported with the generic ISCSI storage array plug-in. This is not applicable to fibre channel storage.
Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accessHost=value</td>
<td>The hostname or IP address for the access host. To find the hostname or IP address, use the <code>show</code> command to display information about the storage array.</td>
</tr>
<tr>
<td>{id=value | name=value}</td>
<td>The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyISCSIServer</code>.</td>
</tr>
</tbody>
</table>

Examples

**Example A.193 Removing a storage array access host**

```
OVM> removeAccessHost StorageArray name=MyISCSIServer accessHost=10.172.76.131
```

See Also

- Section A.3, “addAccessHost”
- Section A.49, “create StorageArray”
- Section A.86, “edit StorageArray”
- Section A.138, “list”
- Section A.171, “show”

A.147 removeAdminServer

Removes an administrative Oracle VM Server from a file server or storage array.

**Syntax**

```
removeAdminServer {FileServer | StorageArray } instance server=value
```

Where `instance` is:

```
\{id=value \| name=value\}
```

**Description**

This command removes an administrative Oracle VM Server from a file server or storage array.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server=value</td>
<td>The name or ID of the administrative Oracle VM Server.</td>
</tr>
</tbody>
</table>
Option | Description
--- | ---
{id=value | name=value} | The instance of the object using either the id or name option, for example name=MyFileServer.

Examples

Example A.194 Removing an admin server from a file server

OVM> removeAdminServer FileServer name=MyNFSServer server=MyServer

See Also

- Section A.4, “addAdminServer”
- Section A.38, “create FileServer”
- Section A.49, “create StorageArray”
- Section A.70, “edit FileServer”
- Section A.86, “edit StorageArray”
- Section A.154, “remove Server”
- Section A.11, “add Server”
- Section A.6, “addRefreshServer”
- Section A.149, “removeRefreshServer”
- Section A.86, “edit StorageArray”

A.148 removePolicyServer

Removes a server pool policy from a server.

Syntax

```
removePolicyServer ServerPool instance server=value
```

Where instance is:

```
{id=value | name=value}
```

Description

This command removes a server pool policy from a server.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>


Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server=value</td>
<td>The name or ID of an Oracle VM Server from which the policy is removed.</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.195 Removing a policy from a server**

```bash
OVM> removePolicyServer ServerPool name=MyServerPool server=MyServer
```

**See Also**

- Section A.5, “addPolicyServer”
- Section A.46, “create ServerPoolNetworkPolicy”
- Section A.83, “edit ServerPoolNetworkPolicy”

**A.149 removeRefreshServer**

Removes an Oracle VM Server that is able to refresh a file server.

**Syntax**

```bash
removeRefreshServer FileServer instance server=value
```

Where `instance` is:

{ `id=value` | `name=value` }

**Description**

This command removes a refresh server from a file server. The refresh server is an Oracle VM Server that is used to refresh the file systems on an NFS file server. A file server must have at least one refresh server assigned to it.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileServer instance</td>
<td>The name or ID of the file server.</td>
</tr>
<tr>
<td>server=value</td>
<td>The name or ID of the Oracle VM Server to be removed as a refresh server.</td>
</tr>
<tr>
<td><code>id=value</code></td>
<td>The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code>.</td>
</tr>
<tr>
<td><code>name=value</code></td>
<td></td>
</tr>
</tbody>
</table>

**Examples**

**Example A.196 Removing a refresh server from a file server**

```bash
OVM> removeRefreshServer FileServer name=MyNFSServer server=MyServer1
```
A.150 remove BondPort

Removes a bond port from a network object.

Syntax

```
remove BondPort instance from Network instance
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command removes a bond port from a network object.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

Examples

**Example A.197 Removing a bonded port from a network**

```
OVMB> remove BondPort id=0004fb00002000065822cb7bb9ec296 from Network name=MyVMNetwork
```

See Also

- Section A.36, "create BondPort"
- Section A.66, "edit BondPort"
A.151 remove FileSystem

Removes a file system from an access group.

Syntax

```
remove FileSystem instance from AccessGroup instance
```

Where `instance` is:

```
{id=value | name=value}
```

Note that if you need to quote the instance name and it contains forward slashes, you need to escape those forward slashes with additional forward slashes. This is illustrated in the example for this command.

Description

This command removes a file system from an access group.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

Examples

**Example A.198 Removing a file system from an access group**

```
OVMD> remove FileSystem name="nfs on 10.172.76.125://mnt//vol2//repo03" from AccessGroup \ name=MyAccessGroup
```

See Also

- Section A.39, “create FileSystem”
A.152 remove PhysicalDisk

Removes a physical disk from a SAN storage access group.

Syntax

```
remove PhysicalDisk instance from AccessGroup instance
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command removes a physical disk from a SAN storage access group. Local storage and generic storage plug-ins are not supported with this command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

Examples

**Example A.199 Removing physical disk from a SAN storage access group**

```
OVM> remove PhysicalDisk id=0004fb00001800007ee6dbda7b4461cb from AccessGroup \ 
    name='Default access group @ MyISCSIServer'
```

See Also

- Section A.41, “create PhysicalDisk”
- Section A.9, “add PhysicalDisk”
- Section A.76, “edit PhysicalDisk”
- Section A.59, “delete”
A.153 remove Port

Removes an Ethernet port from a network object.

Syntax

```
remove Port instance from { BondPort | Network } instance
```

Where `instance` is:

```
{id=value | name=value }
```

Description

This command removes an Ethernet port from a network object.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{ BondPort</td>
<td>Network }`</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Examples

```
Example A.200 Removing an Ethernet port from a network

OVM> remove Port id=0004fb0000200000be8fa354cb7d98ae from Network name=MyVMNetwork
```

See Also

- Section A.36, “create BondPort”
- Section A.66, “edit BondPort”
- Section A.7, “add BondPort”
- Section A.10, “add Port”
- Section A.102, “embeddedCreate”
- Section A.103, “embeddedDelete”
- Section A.104, “embeddedEdit”
A.154 remove Server

Removes an Oracle VM Server from an object.

Syntax

```
remove Server instance from {AccessGroup | CpuCompatibilityGroup | Repository | ServerPool} instance
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command removes an Oracle VM Server from either a CPU compatibility group, server pool, storage repository or access group.

To remove admin servers from a file server or storage array, use the `removeAdminServer` command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{ AccessGroup</td>
<td>CpuCompatibilityGroup</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Examples

**Example A.201 Removing an Oracle VM Server from a CPU compatibility group**

```
OVM> remove Server name=MyServer from CpuCompatibilityGroup name=MyCPUGroup
```

**Example A.202 Removing an Oracle VM Server from a server pool**

```
OVM> remove Server name=MyServer from ServerPool name=MyServerPool
```

**Example A.203 Removing an Oracle VM Server from an access group**

```
OVM> remove Server name=MyServer from AccessGroup name=MyAccessGroup
```

**Example A.204 Removing (unpresenting) an Oracle VM Server from a storage repository**

```
OVM> remove Server name=MyServer from Repository name=MyRepository
```
See Also

- Section A.60, “discoverServer”
- Section A.11, “add Server”
- Section A.80, “edit Server”
- Section A.142, “refresh”
- Section A.177, “start”
- Section A.178, “stop”
- Section A.162, “restart”
- Section A.137, “kill”
- Section A.181, “upgrade”
- Section A.154, “remove Server”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

A.155 remove ServerPool

Removes a server pool from a storage repository or file system access group.

**Syntax**

```
remove ServerPool instance from {AccessGroup | Repository} instance
```

Where `instance` is:

```
{id=value | name=value}
```

**Description**

This command unpresents a storage repository from the Oracle VM Servers in a server pool. To unpresent a storage repository to an individual Oracle VM Server, use the `remove Server` command.

This command also removes a server pool from an access group.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{AccessGroup</td>
<td>Repository</td>
</tr>
</tbody>
</table>
### A.156 remove StorageInitiator

Removes a storage initiator from an access group for a SAN storage server.

**Syntax**

```
remove StorageInitiator instance from AccessGroup instance
```

Where `instance` is:

```
{id=value | name=value}
```

**Description**

This command removes a storage initiator from an access group for a SAN storage server.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.207 Removing a storage initiator**

```
OVM> remove StorageInitiator name=iqn.1988-12.com.oracle:d72d82d0817f from AccessGroup name='Default access group @ MyISCSIServer'
```
A.157 remove Tag

Removes a tag from an object.

Syntax

```
remove Tag instance from { Server | ServerPool | Vm } instance
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command removes a tag used to identify and group objects from an object.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{ Server</td>
<td>ServerPool</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Examples

Example A.208 Removing a tag from a server pool

```
OVM> remove Tag name=MyTag from ServerPool name=MyServerPool
```

See Also

- Section A.50, “create Tag”
- Section A.88, “edit Tag”
- Section A.14, “add Tag”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”
Syntax

Removes a VLAN interface from a network.

Syntax

\texttt{remove VlanInterface instance from Network instance}

Where \textit{instance} is:

\{ \texttt{id=value} \mid \texttt{name=value} \}

Description

This command removes a VLAN interface from a network.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ \texttt{id=value} \mid \texttt{name=value} }</td>
<td>The instance of the object using either the \texttt{id} or \texttt{name} option, for example \texttt{name=MyVlanInterface}.</td>
</tr>
</tbody>
</table>

Examples

Example A.209 Removing a VLAN interface from a network

\texttt{OVM> remove VlanInterface name=MyVLANInterface from Network name=MyNetwork}

See Also

- Section A.52, “create VlanInterface”
- Section A.15, “add VlanInterface”
- Section A.94, “edit VlanInterface”
- Section A.102, “embeddedCreate”
- Section A.103, “embeddedDelete”
- Section A.104, “embeddedEdit”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

A.159 remove Vm

Removes a virtual machine from an Oracle VM Server, server pool, or anti affinity group.

Syntax

\texttt{remove Vm instance from \{AntiAffinityGroup} | Server | ServerPool} \texttt{instance}
Where instance is:

{ id=value | name=value }

Description

This command removes a virtual machine from an Oracle VM Server, server pool, or anti affinity group. The virtual machine cannot be running, and must be stopped before using this command.

In the Oracle VM Manager Web Interface:

• When you remove a virtual machine from an Oracle VM Server, it is moved to the server pool and is available by displaying the Virtual Machines perspective at the server pool level in the Servers and VMs tab.

• When you remove a virtual machine from a server pool, it is moved to the Unassigned Virtual Machines folder in the Servers and VMs tab.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ AntiAffinityGroup</td>
<td>Server</td>
</tr>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

Example A.210 Removing a virtual machine from a server pool

OVM> remove Vm name=MyVM from ServerPool name=MyServerPool

Example A.211 Removing a virtual machine from an Oracle VM Server

OVM> remove Vm name=MyVM from Server name=MyServer

Example A.212 Removing a virtual machine from an anti affinity group

OVM> remove Vm name=MyVM from AntiAffinityGroup name=MyAAGroup

See Also

• Section A.16, “add Vm”
• Section A.138, “list”
• Section A.171, “show”

A.160 remove Vnic

Removes a VNIC from a network.
Syntax

```
remove Vnic instance from Network instance
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command removes a VNIC from a network.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

Examples

**Example A.213 Removing a VNIC from a network**

```
OVM> remove Vnic name=00:21:f6:00:00:00 from Network name=MyNetwork
```

See Also

- Section A.58, “create Vnic”
- Section A.100, “edit Vnic”
- Section A.17, “add Vnic”
- Section A.170, “setVnicMacAddrRange”
- Section A.129, “getVnicMacAddrRange”
- Section A.59, “delete”
- Section A.138, “list”
- Section A.171, “show”

**A.161 resize**

Resizes a physical or virtual disk.

**Syntax**

```
resize {PhysicalDisk | VirtualDisk} instance size=value sparse= {Yes | No}
```

Where `instance` is:

```
{id=value | name=value}
```
Description

This command resizes a physical or virtual disk. The `sparse` option is only available when resizing a virtual disk.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>size=value</code></td>
<td>The size of the physical or virtual disk in GiB.</td>
</tr>
<tr>
<td>`sparse= { Yes</td>
<td>No }`</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Examples

**Example A.214 Resizing a virtual disk**

```
OVM> resize VirtualDisk name=MyVMDisk size=200 sparse=Yes
```

**Example A.215 Resizing a physical disk**

```
OVM> resize PhysicalDisk name=MyVMDisk size=200
```

See Also

- Section A.41, “create PhysicalDisk”
- Section A.9, “add PhysicalDisk”
- Section A.76, “edit PhysicalDisk”
- Section A.152, “remove PhysicalDisk”
- Section A.135, “importVirtualDisk”
- Section A.51, “create VirtualDisk”
- Section A.93, “edit VirtualDisk”
- Section A.59, “delete”
- Section A.142, “refresh”
- Section A.138, “list”
- Section A.171, “show”
- Section A.113, “getEvents”

A.162 restart

Restarting an Oracle VM Server or virtual machine.
Syntax

```
restart { Server | Vm } instance
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command restarts an Oracle VM Server or virtual machine.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{ Server</td>
<td>Vm }`</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Examples

**Example A.216 Restarting an Oracle VM Server**

```
OVM> restart Server name=MyServer
```

**Example A.217 Restarting a virtual machine**

```
OVM> restart Vm name=MyVM
```

See Also

- Section A.177, "start"
- Section A.178, "stop"
- Section A.137, "kill"

A.163 resume

Resumes a suspended virtual machine.

Syntax

```
resume Vm instance
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command resumes a suspended virtual machine.
Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.218 Resuming a virtual machine

OVM> resume Vm name=MyVM

See Also

- Section A.179, “suspend”
- Section A.138, “list”
- Section A.171, “show”

A.164 sendVmMessage

Sends a key/value pair message to a running virtual machine.

Syntax

```
sendVmMessage Vm instance key=value message=value log={Yes|No}
```

Where instance is:

{id=value | name=value}

Description

This command sends a key/value pair message to a running virtual machine.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key=value</td>
<td>The message key.</td>
</tr>
<tr>
<td>message=value</td>
<td>The message content.</td>
</tr>
<tr>
<td>log={Yes</td>
<td>No}</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

Examples

Example A.219 Sending a message to a virtual machine

OVM> sendVmMessage Vm name=MyVM key=com.oracle.linux.network.device.0 message=eth0 log=No
See Also

- Section A.128, "getVmSentMessages"
- Section A.127, "getVmReceivedMessages"
- Section A.22, "clearVmRcvdMessage"
- Section A.23, "clearVmSentMessage"
- Section A.21, "clearVmAllSentMessages"
- Section A.20, "clearVmAllRcvdMessages"

A.165 set

Sets CLI session configuration options.

Syntax

```plaintext
set { AlphabetizeAttributes= \{ Yes | No \} | CommandMode= \{ Asynchronous | Synchronous \} | CommandTimeout= value | EndlineChars= \{ CRLF | CR | LF \} | OutputMode= \{ Verbose | Sparse | Xml \} }
```

Description

This command sets the CLI session configuration options. To show the values for the CLI session options, use the `showclisession` command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`AlphabetizeAttributes= { Yes</td>
<td>No }`</td>
</tr>
<tr>
<td>`CommandMode= { Asynchronous</td>
<td>Synchronous }`</td>
</tr>
</tbody>
</table>
Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommandTimeout= value</td>
<td>Sets when the CLI will timeout. value is seconds, and can be an integer between 1 and 43200. The default is 7200 seconds (two hours).</td>
</tr>
<tr>
<td>EndlineChars={ CRLF</td>
<td>CR</td>
</tr>
<tr>
<td>OutputMode= { Verbose</td>
<td>Sparse</td>
</tr>
</tbody>
</table>

Examples

Example A.220 Setting end of line characters

OVM> set EndlineChars=LF

Example A.221 Setting output mode to XML

OVM> set OutputMode=Xml

See Also

• Section A.173, “showclisession”

A.166 setArchiveConfig

Sets the configuration for managing archived statistics.
**Syntax**

```plaintext
setArchiveConfig interval=value
```

**Description**

This command configures how Oracle VM Manager manages archived statistics. To show the configuration, use the `getArchiveConfig` command.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>interval=value</code></td>
<td>The interval, in minutes, at which Oracle VM Manager deletes archived statistics. Oracle VM Manager uses the statistics hold time to calculate how many archived statistics to delete. For example, you set the hold time to 15 minutes and the archive interval to 2 days. In this case, every 2 days Oracle VM Manager deletes archived statistics that are older than 15 minutes from the current time. The value for this field can be between 30 minutes and 525600 minutes (1 year). The default value is 60 minutes.</td>
</tr>
</tbody>
</table>

**Examples**

Example A.222 Setting the interval to delete archived statistics

```
OVM> setArchiveConfig interval=1200
```

**See Also**

- Section A.107, "getArchiveConfig"
- Section A.169, "setStatsConfig"

**A.167 setDbBackupConfig**

Sets the configuration for the automated database backup facility within Oracle VM Manager.

**Syntax**

```plaintext
setDbBackupConfig interval=value numberToKeep=value
```

**Description**

This command sets the configuration for the automated database backup facility within Oracle VM Manager. Further information on the automated database backup facility is provided in Backing up the MySQL Database Repository in the Oracle VM Administrator's Guide.

**Options**

The following table shows the available options for this command.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interval=value</td>
<td>Sets how frequently the automated backups of the Oracle VM Manager database are performed. The value is measured in minutes and can be any integer between 360 minutes (6 hours) and 43200 minutes (30 days).</td>
</tr>
<tr>
<td>numberToKeep=value</td>
<td>Sets how many automated backups of the Oracle VM Manager database are retained before they are rotated. The value can be between 1 and 500.</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.223 Setting the Oracle VM Manager database backup configuration**

```
OVM> setDbBackupConfig interval=1440 numberToKeep=14
```

**See Also**

- Section A.109, "getDbBackupConfig"

**A.168 setMaintenanceMode**

Sets maintenance mode on an Oracle VM Server.

**Syntax**

```
setMaintenanceMode Server instance mode={on | off}
```

Where `instance` is:

```
{id=value | name=value}
```

**Description**

This command sets whether an Oracle VM Server is in maintenance mode in order to perform software updates.

**Options**

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode={on</td>
<td>off}</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

**Examples**

**Example A.224 Placing an Oracle VM Server into maintenance mode**

```
OVM> setMaintenanceMode Server name=MyServer mode=on
```
A.169 setStatsConfig

Sets the configuration for the statistics displayed in Oracle VM Manager.

Syntax

```
setStatsConfig statisticsEnabled={Yes|No} samplingInterval=value holdTime=value
fsSamplingInterval=value
```

Description

This command sets the configuration for the statistics displayed in Oracle VM Manager.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`statisticsEnabled={Yes</td>
<td>No}`</td>
</tr>
<tr>
<td><code>samplingInterval=value</code></td>
<td>The number of seconds between which statistics should be recorded. <code>value</code> may be an integer between 20 and 31536000.</td>
</tr>
<tr>
<td><code>holdTime=value</code></td>
<td>The number of minutes to retain the statistics. <code>value</code> may be an integer between 15 and 4320.</td>
</tr>
<tr>
<td><code>fsSamplingInterval=value</code></td>
<td>The number of seconds between which file system statistics should be recorded. <code>value</code> may be an integer between 60 and 31536000.</td>
</tr>
</tbody>
</table>

Examples

**Example A.225 Setting the Oracle VM Manager statistics configuration**

```
OVM> setStatsConfig samplingInterval=60 holdTime=15
```

See Also

- Section A.120, "getStatsConfig"
- Section A.108, "getAverageStat"
- Section A.116, "getLatestStat"
- Section A.117, "getLatestStatForList"
A.170 setVnicMacAddrRange

Sets the range of MAC addresses that are available to VNICs.

Syntax

```
setVnicMacAddrRange [ oui=value ] [ start=value ] [ end=value ]
```

Description

This command sets the range of MAC addresses that can be used when creating a VNIC. To display the MAC address range, use the `getVnicMacAddrRange` command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oui=value</td>
<td>The OUI (Organizationally Unique Identifier) is used as the first three octets of the MAC address. The MAC address is created by combining the OUI as the first three octets with the randomly selected second three octets in the range of start to end values specified, inclusive. Therefore, the value specified here should represent the first three octets that make up the MAC address. The default OUI is 00:21:f6 and is owned by Oracle. Changing the OUI can result in an overlap of MAC addresses on your network causing MAC address spooking, network conflicts and unexpected network behavior. It is recommended that you do not change this value.</td>
</tr>
<tr>
<td>start=value</td>
<td>The parameter used to specify the first possible value for the second three octets that form the MAC address. The default start value is 00:00:00.</td>
</tr>
<tr>
<td>end=value</td>
<td>The parameter used to specify the final possible value for the second three octets that form the MAC address. The default start value is FF:FF:FF.</td>
</tr>
</tbody>
</table>

Examples

**Example A.226 Setting the VNIC MAC Address Range**

```
OVM> setVnicMacAddrRange oui=00:21:f6 start=00:00:00 end=FF:FF:FF
```

See Also

- Section A.129, "getVnicMacAddrRange"
- Section A.58, “create Vnic”
- Section A.100, “edit Vnic”
- Section A.17, “add Vnic”
A.171 show

Shows information about an object.

Syntax

```
show { AccessGroup | AntiAffinityGroup | Assembly | AssemblyVirtualDisk | AssemblyVm |
BondPort | ControlDomain | Cpu | CpuCompatibilityGroup | FileServer | FileServerPlugin |
FileSystem | Job | Manager | Network | PeriodicTask | PhysicalDisk | Port | Repository |
RepositoryExport | Server | ServerController | ServerPool | ServerPoolNetworkPolicy |
ServerUpdateGroup | ServerUpdateRepository | StorageArray | StorageArrayPlugin |
StorageInitiator | Tag | VirtualAppliance | VirtualApplianceVirtualDisk |
VirtualApplianceVm | VirtualCdrom | VirtualDisk | VlanInterface | Vm |
VmCloneCustomizer | VmCloneNetworkMapping | VmCloneStorageMapping | VmDiskMapping |
Vnic | VolumeGroup } instance
```

Where `instance` is:

```
{ id=value | name=value }
```

Description

This command shows information about an object. Use the `list` command to find all instances of an object type, then use the `show` command to show more detailed information about the object.

To display the output in alphabetical order, use the `set AlphabetizeAttributes=Yes` command. To show CLI session options set using the `set` command, use the `showclisession` command.

A Job object does not have a `name` attribute, only an `id` attribute. The `show Job name=value` command is the same as entering `show Job id=value`. You can use these two options interchangeably. Any `name` attributes are automatically converted to `ids`.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ AccessGroup</td>
<td>AntiAffinityGroup</td>
</tr>
</tbody>
</table>

The `Assembly`, `AssemblyVirtualDisk` and `AssemblyVm` options have been deprecated. You should instead use `VirtualAppliance`, `VirtualApplianceVirtualDisk`, `VirtualApplianceVm`, etc.
### Examples

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServerController</td>
<td>ServerPool</td>
</tr>
<tr>
<td>{id=value</td>
<td>name=value}</td>
</tr>
</tbody>
</table>

#### Examples

**Example A.227 Showing details about an Oracle VM Server**

```
OVM> show Server name=MyServer
```

**Example A.228 Showing details about a virtual machine**

```
OVM> show Vm name=MyVM
```

**Example A.229 Showing details about a network**

```
OVM> show Network id=0004fb0010ff705
```

**Example A.230 Showing details about a VNIC**

```
OVM> show Vnic name=00:21:f6:00:00:0b
```

Using the `show Vnic` command can provide you with information such as the virtual machine that is using a particular VNIC, as well as the IP addresses that are configured for that VNIC.

---

**Note**

IP addresses are only displayed for VNICs attached to virtual machines that have been properly set up with the Oracle VM Guest Additions packages. See the *Oracle VM Administrator's Guide* for information on using the Oracle VM Guest Additions packages.

---

**See Also**

- Section A.138, “list”
- Section A.165, “set”
- Section A.173, “showclisession”
A.172 showallcustomcmds

Provides a list of all commands and the objects that they relate to.

Syntax

    showallcustomcmds

Description

This command provides a list of all commands along with the objects that the commands relate to.

Options

This command does not take any arguments or provide any options.

Examples

Example A.231 Showing all custom commands

    OVM> showallcustomcmds

See Also

- Section A.130, “help”
- Section A.174, “showcustomcmds”

A.173 showclisession

Provides a list of CLI session options and their settings.

Syntax

    showclisession

Description

This command provides a list of the CLI session options and their settings. The CLI session options are set using the set command.

Options

This command does not take any arguments or provide any options.

Examples

Example A.232 Showing CLI session options and settings

    OVM> showclisession

See Also

- Section A.165, “set”
A.174 showcustomcmds

Shows available custom commands for an object type.

Syntax

```
```

Description

This command shows the available custom commands specific to an object. Use the `showobjtypes` command to find all object types, then use the `showcustomcmds` command to show associated commands.

**Note**

Not all object types have custom commands associated. For example, the `YumConfig` object type does not have any custom commands.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>VmCloneNetworkMapping</td>
<td></td>
</tr>
<tr>
<td>VmCloneStorageMapping</td>
<td></td>
</tr>
<tr>
<td>VmDiskMapping Vnic VolumeGroup</td>
<td></td>
</tr>
</tbody>
</table>

### Examples

**Example A.233 Showing custom commands for an Oracle VM Server**

```
OVM> showcustomcmds Server
```

**Example A.234 Showing custom commands for a virtual machine**

```
OVM> showcustomcmds VM
```

**Example A.235 Showing custom commands for a repository**

```
OVM> showcustomcmds Repository
```

### See Also

- Section A.130, “help”
- Section A.172, “showallcustomcmds”
- Section A.175, “showobjtypes”

#### A.175 showobjtypes

Provides a list of all object types.

**Syntax**

```
showobjtypes
```

**Description**

This command provides a list of all object types. This command is useful to assist in determining which object types can be acted upon.

**Options**

This command does not take any arguments or provide any options.

**Examples**

**Example A.236 Showing all object types**

```
OVM> showobjtypes
```

### See Also

- Section A.113, “getEvents”
- Section A.114, “getEventsForObject”
• Section A.130, “help”
• Section A.174, “showcustomcmds”

A.176 showversion

Shows the version number of the CLI/Oracle VM Manager.

Syntax

showversion

Description

This command shows the version number of the CLI/Oracle VM Manager.

Options

This command does not take any arguments or provide any options.

Examples

Example A.237 Showing the CLI/Oracle VM Manager version number

OVM> showversion

See Also

• Section A.130, “help”

A.177 start

Starts an Oracle VM Server or virtual machine.

Syntax

start {Server | Vm} instance

Where instance is:

{ id=value | name=value }

Description

This command starts an Oracle VM Server or virtual machine.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Server</td>
<td>Vm}</td>
</tr>
</tbody>
</table>
Examples

Example A.238 Starting an Oracle VM Server

```bash
OVM> start Server name=MyServer
```

Example A.239 Starting a virtual machine

```bash
OVM> start Vm name=MyVM
```

See Also

- Section A.162, “restart”
- Section A.178, “stop”
- Section A.137, “kill”

A.178 stop

Stops an Oracle VM Server or virtual machine.

Syntax

```
stop {Server | Vm} instance
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command stops an Oracle VM Server or virtual machine.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{Server</td>
<td>Vm}`</td>
</tr>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

Examples

Example A.240 Stopping an Oracle VM Server

```bash
OVM> stop Server name=MyServer
```
Example A.241 Stopping a virtual machine

```bash
OVM> stop Vm name=MyVM
```

See Also

- Section A.177, "start"
- Section A.162, "restart"
- Section A.137, "kill"

A.179 suspend

Suspends a running virtual machine.

Syntax

```bash
suspend Vm instance
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command suspends a running virtual machine.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

Examples

Example A.242 Suspending a virtual machine

```bash
OVM> suspend Vm name=MyVM
```

See Also

- Section A.163, "resume"
- Section A.138, "list"
- Section A.171, "show"

A.180 takeOwnership

Take ownership of an Oracle VM Server or storage repository.
Syntax

```
takeOwnership{Repository instance[serverpool=value]|Server instance password=value}
```

Where `instance` is:

```
{ id=value|name=value }
```

Description

This command takes ownership of an Oracle VM Server or storage repository. After taking ownership of a repository, you should refresh it using the `refresh` command. If a server is only partially discovered, in the sense that it is already under the ownership of another Oracle VM Manager instance, and ownership is subsequently released, you may need to rediscover the server before you are able to take ownership of it in the current Oracle VM Manager instance.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{ Repository instance [serverpool=value]</td>
<td>Server instance password=value }`</td>
</tr>
<tr>
<td>`{ id=value</td>
<td>name=value }`</td>
</tr>
</tbody>
</table>

Examples

**Example A.243 Taking ownership of an Oracle VM Server**

```
OVM> takeOwnership Server name=MyServer password=********
```

**Example A.244 Taking ownership of a storage repository**

```
OVM> takeOwnership Repository name=MyRepository serverpool=MyServerPool
```

See Also

- Section A.145, "releaseOwnership"
- Section A.142, "refresh"

A.181 upgrade
Syntax

Upgrades an Oracle VM Server.

Syntax

upgrade Server instance

Where instance is:

{ id=value | name=value }

Description

This command updates or upgrades an Oracle VM Server using a server update repository. The repository used for the upgrade is set using the create ServerUpdateRepository command. This command places the Oracle VM Server into maintenance mode, checks for any updates in the server update repository, installs any updates, restarts the Oracle VM Server, then takes it out of maintenance mode and returns it to the server pool as a fully functioning member of the pool.

You can check if an Oracle VM Server has an update available using the checkUpToDate command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ id=value</td>
<td>name=value }</td>
</tr>
</tbody>
</table>

Examples

Example A.245 Upgrading an Oracle VM Server

OVM> upgrade Server name=MyServer

See Also

- Section A.47, “create ServerUpdateGroup”
- Section A.84, “edit ServerUpdateGroup”
- Section A.48, “create ServerUpdateRepository”
- Section A.85, “edit ServerUpdateRepository”
- Section A.19, “checkUpToDate”
- Section A.168, “setMaintenanceMode”
- Section A.138, “list”
- Section A.171, “show”

A.182 validate

Validates a storage array.
Syntax

```
validate StorageArray instance
```

Where `instance` is:

```
{id=value | name=value}
```

Description

This command validates a storage array using the storage array plug-in. Validation is required after the storage array is discovered and after modification of storage array attributes. At least one administrative Oracle VM Server must be configured before using this command.

Options

The following table shows the available options for this command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`{id=value</td>
<td>name=value}`</td>
</tr>
</tbody>
</table>

Examples

Example A.246 Validating a storage array

```
OVM> validate StorageArray name=MyISCSIServer
```

See Also

- Section A.142, “refresh”
- Section A.143, “refreshAll”
- Section A.138, “list”
- Section A.171, “show”
Glossary

A
admin server
An Oracle VM Server dedicated to performing administrative functions on storage servers such as creating a new LUN or extending a file system. The server must be capable of logging into a storage array or file server as an admin user. The administrative functions available to the server are defined by the Oracle VM Storage Connect plug-in.

assembly
An infrastructure template containing a configuration of multiple virtual machines with their virtual disks, and the inter-connectivity between them. Assemblies can be created as a set of .ovf (Open Virtualization Format) and .img (disk image) files, or may all be contained in a single .ova (Open Virtualization Format Archive) file.

As of Release 3.4.1, assemblies are known as virtual appliances.

B
bonding
Bonding is a Linux OS feature that provides a method for aggregating several ports into a single bonded interface, to provide load balancing or redundancy. When you discover an Oracle VM Server, the bonded interface is shown as containing a single port.

Network bonding refers to the combination of network interfaces on one host for redundancy and/or increased throughput. Redundancy is the key factor: You want to protect your virtualized environment from loss of service due to failure of a single physical link.

In Oracle VM, there are three modes of network bonding:

- Active - Backup: One Network Interface Card (NIC) is active while the other NIC is standby. If the active NIC goes down, the other NIC becomes active.
- Dynamic Link Aggregation: All NICs act as one NIC and the network traffic flows through all interfaces concurrently, which results in a higher throughput. With this mode, your network administrator must create LACP (Link Aggregation Control Protocol) bonding on the network switch(es).
- Load Balanced: The network traffic is equally balanced over the NICs of the bond. This mode does not require any special configuration on connected network switch(es), However, this mode does not support using VLAN with bridges. If using this mode for your bonded interfaces in any network, you cannot use VLANs if this network is configured with the Virtual Machine network channel.

C
clone
The action or result of making an exact copy of an object. The object may be a virtual machine, virtual machine template, ISO file, or virtual disk. Cloning is similar to copying and maintains the integrity of the original object, while creating a new object based on the original. A clone customizer may be used to define cloning options to specify details of where the object components may reside when cloned, such as in a different storage repository.

D
discover
The process of adding systems as objects within Oracle VM Manager is known as discovery. When you add Oracle VM Servers and storage to your Oracle VM environment, Oracle VM Manager uses the information
provided to connect to the resource and perform verification. During this process, information is usually exchanged between the server and the manager. In the case of an Oracle VM Server, Oracle VM Manager obtains information about the server, its network connectivity and any storage that is already attached to the server. Depending on your hardware and networking configuration, external storage may be automatically detected during discovery of Oracle VM Servers. This is always the case with local OCFS2 storage on an Oracle VM Server.

While storage can be automatically discovered during the process of discovering Oracle VM Servers, you may need to perform storage discovery for resources that are not already attached to any of your Oracle VM Servers. It is important that storage is configured outside of the Oracle VM environment prior to discovery. Depending on the storage type, you can perform different storage discovery operations from within Oracle VM Manager.

dom0
An abbreviation for domain zero. The management domain with privileged access to the hardware and device drivers. Dom0 is the first domain started at boot time. Dom0 has more privileges than domU. It can access the hardware directly and can manage the device drivers for other domains. It can also start new domains.

See Also: control domain

domU
An unprivileged domain with no direct access to the hardware or device drivers. Each domU is started by dom0.

E

events
Events are used to register status information of "objects" within Oracle VM Manager for future reference or to make problems easier to trace back. Events are often, though not always, related to jobs that are initiated within Oracle VM Manager. For instance, when a job fails, an event is generated. Events can also be triggered through changes in the environment such as server crashes or storage disconnects. Therefore, events are used to alert you to potential problems that may need your attention.

Events are categorized by severity. Most events will be informational, but they can also be warnings or errors. If an event has an error level severity, you need to acknowledge the error event to clear the error and to perform further operations on the object that generated the error.

See Also: jobs

G

guest
A guest operating system that runs within a domain in Oracle VM Server. A guest may be paravirtualized or hardware virtualized. Multiple guests can run on the same Oracle VM Server.

H

host computer
The physical computer on which the software is installed. Typically used to refer to either the computer on which Oracle VM Server or Oracle VM Manager is running.

J

jobs
Jobs consist of discrete operations that take place through Oracle VM Manager, such as server discovery, presenting a repository and creating a VM. Jobs are assigned a status that is refreshed according to their progress. A history of all jobs in the environment is stored within Oracle VM Manager.
Since jobs are often performed sequentially and sometimes take time to complete, tracking the status of a job allows you to understand what actions the system is currently performing, and which actions are queued to run in sequence after the current job has completed. Jobs also allow you to access system messages that may be useful to debug the failure of an operation.

Most jobs tend to generate events that each have a different severity level.

See Also: events

L

local storage
Local storage consists of hard disks installed locally in an Oracle VM Server. Local storage is often not appropriate for enterprise production environments, because it sharply constrains the ability of a virtual machine to run anywhere in the server pool in the event of the failure of the Oracle VM server, which owns the local storage, and because the management overhead of this storage is often significant.

M

migrate
The act of moving a virtual machine from one Oracle VM Server to another, or to the Unassigned Virtual Machines folder. Technically, a migration can only be performed on a running virtual machine, however the Oracle VM Manager Web Interface and Oracle VM Manager Command Line Interface may combine multiple operations to make it appear that you can perform a migration on either a running or a stopped virtual machine.

move
The act of moving an object from one location to another. This may be moving a stopped virtual machine from one Oracle VM Server to another, moving a virtual machine template from one storage repository to another, or moving an ISO file or virtual disk to another storage location. In the case where a live migration is performed on a virtual machine located on a local repository, the migration operation may move the virtual machine across repositories even though the virtual machine is running as part of the migration process.

multipath
The technique of creating more than one physical path between the server CPU and its storage devices. It results in better fault tolerance and performance enhancement. Oracle VM supports multipath I/O out of the box. Oracle VM Servers are installed with multipathing enabled because it is a requirement for SAN disks to be discovered by Oracle VM Manager.

O

Oracle VM Server
A self-contained virtualization environment designed to provide a lightweight, secure, server-based platform for running virtual machines. The Oracle VM Server comprises a hypervisor and a privileged domain (called dom0) that allow multiple domains or guest operation systems (such as Linux, Solaris, and Windows) to run on one physical machine. Includes Oracle VM Agent to enable communication with Oracle VM Manager.

The Oracle VM Server for x86 incorporates an open source Xen hypervisor component, which has been customized and optimized to integrate into the larger, Oracle-developed virtualization server. The Oracle VM Server for x86 is also responsible for access and security management and generally acts as the server administrative entity, because the hypervisor's role is limited.

On Oracle VM Server for SPARC systems, the SPARC hypervisor is built into the SPARC firmware and is generally referred to as the Logical Domains Manager. As with the Xen hypervisor, each virtual machine is securely executed on a single computer and runs its own guest Oracle Solaris operating system.
refresh server
An Oracle VM Server dedicated to handling file system refreshes on behalf of a server pool. A refresh server temporarily mounts file systems on an NFS file server during the refresh operation. The server must be granted full data access in order to perform the refresh. For each NFS file server, at least one Oracle VM Server from each server pool accessing the file server must be assigned as a refresh server.

server processor compatibility group
A server processor compatibility group is a group of Oracle VM Servers with compatible processors, or CPUs sharing the same processor family and model number. These groups are created to ensure that a virtual machine running on one Oracle VM Server can safely be migrated and continue to run on another Oracle VM Server. Oracle VM Manager automatically creates processor compatibility groups as it discovers servers that have different processor types.

Using Oracle VM Manager you can create custom compatibility groups to improve your ability to do smooth migrations and to group servers according to your own requirements.
Index

Symbols
?
command, 9

A
abort
job, 41
access group
add file system, 47
add Oracle VM Server, 49
add physical disk, 48
add server pool, 51, 51
create, 73
delete, 111, 111
edit, 114
list, 205, 205, 205
refresh, 210
remove file system, 218
remove physical disk, 219
remove server pool, 222
show, 237, 237, 237
access host
add, 42
remove, 213
AccessGroup
showcustomcmds, 240
acknowledge
event, 41
add
access group, 47
access host, 42
admin server, 43
network bond, 46, 76
NFS refresh server, 45
Oracle VM Server to access group, 49
Oracle VM Server to CPU compatibility group, 49
Oracle VM Server to repository, 49
Oracle VM Server to server pool, 49
physical disk to access group, 48
policy server, 44
port, 48
refresh server, 45
server pool to access group, 51
server pool to repository, 51
storage initiator, 52
tag, 53
virtual machine to anti affinity group, 55
virtual machine to Oracle VM Server, 55
virtual machine to server pool, 55
VLAN interface to network, 54
VNIC, 56
admin server
add, 43
remove, 214
anti affinity group
add virtual machine, 55
create, 75
delete, 111
edit, 115
list, 205
remove virtual machine, 225
show, 237
AntiAffinityGroup
showcustomcmds, 240
archive
get configuration, 168
set configuration, 232
assembly
create virtual machine template from, 70, 70
edit, 116, 117
get descriptor, 171
import, 198
refresh, 210
assembly virtual machine
edit, 118

B
bond port
add Ethernet port, 49
create IP address, 162
delete, 111
delete IP address, 164
edit IP address, 165
list, 205
show, 237
BondPort
showcustomcmds, 240

C
case sensitivity, 16
CDROM
adding to a virtual machine, 109
clone to repository, 63
delete, 111
edit, 149
import, 201
list, 205
mapping to a virtual machine, 109
show, 237
checkUpToDate, 58
CLI
connect to, 6
restart, 5
start, 5
status, 5
stop, 5
clonecustomizer, 62
disk, 63, 64, 65, 67, 68, 69
physical disk to physical disk, 64
physical disk to repository, 65
physical disk to storage array, 67
template, 62
virtual cdrom to repository, 63
virtual disk to physical disk, 68
virtual disk to repository, 69
virtualmachine, 62
clonemodal
adding to a virtualmachine, 105
clone, 62
create, 105
delete, 111edit, 156
list, 205
mapnetworks, 157
mappingnetworks, 106
mappingstorage, 107, 158
movervirtualmachine, 209
show, 237
shownetworkmapping, 237
showstoragemapping, 237
clonenetworkmapping
delete, 111
list, 205
clonestoragemapping
delete, 111
list, 205
commandmode
set, 231
commandtimeout
set, 231
configurationfile
virtualmachine, 194
connect
to CLI, 6
controldomain
edit, 121
list, 205
show, 237
ControlDomain
showcustomcmds, 240
CPU
edit, 121
list, 205
show, 237
Cpu
showcustomcmds, 240
CPUcompatibilitygroup
add Oracle VM Server, 49
create, 78
delete, 111edit, 122
list, 205
remove Oracle VM Server, 221
show, 237
CpuCompatibilityGroup
showcustomcmds, 240
create
accessgroup, 73, 73
anti affinity group, 75
clonemodal, 105
CPUcompatibilitygroup, 77
disk, 83, 97
Ethernet network, 82
fileserver, 78
IP address, 162
OCFS2 file system, 80
physical disk, 83
repository, 84
repositoryexport, 86
servercontroller, 87
serverpool, 88
serverpoolnetworkpolicy, 90
serverupdategroup, 91
serverupdaterepository, 92
storagearray, 94
storagearrayaccessgroup, 73
storagerpository, 84
tag, 97
virtual disk, 97
virtualmachine, 100
virtualmachineclonemodal, 105
virtualmachinefromvirtualappliance, 71, 72
virtualmachinetemplatefromassembly, 70
VLAN interface, 99
VNIC, 110
D
databasebackup
configuration, 170, 233
debug
job, 171
delete
command, 111
IP address, 164
discover
fileserver, 79
Oracle VM Server, 113
storagerarray, 94
disk
clone, 63, 64, 65, 67, 68, 69
create, 83, 97
delete, 111
edit, 130, 150
import, 202
list, 205
show, 237
disk mapping
delete, 111
edit, 159
list, 205
show, 237
disk mapping
delete, 111
edit, 159
list, 205
show, 237

e
edit
access group, 114
anti affinity group, 115
assembly, 116, 117
assembly virtual machine, 118
CDROM, 149
customizer, 156
control domain, 121
CPU, 121
CPU compatibility group, 122
disk, 130, 150
disk mapping, 159
Ethernet network, 127
Ethernet port, 131
file server, 123
file server plug-in, 124
IP address, 165
ISO, 149
Manager, 127
network bond, 119
OCFS2 file system, 125
Oracle VM Manager, 126
Oracle VM Server, 134
periodic task, 129
physical disk, 129
repository, 132
repository export, 133
server controller, 136
server pool, 136
server pool network policy, 139
server update group, 140, 141
storage array, 142
storage array plug-in, 144
tag, 145
virtual appliance, 146, 147
virtual appliance virtual machine, 148
virtual disk, 150
virtual machine, 152
virtual machine clone customizer, 156
virtual machine disk mapping, 159
VLAN interface, 151
VNIC, 160
volume group, 162
end of line character
set, 9, 231
escape
special characters, 9
Ethernet network
create, 82
edit, 127
list, 237
Ethernet port
add bond, 76
add to network object, 49
edit, 131
edit bond, 119
list, 237
event
acknowledge, 41
events
get, 179, 181
generate by query, 172
generate highest severity, 192
generate highest severity for object(s), 193
exit, 166
export
virtual appliance, 166

F
file server
add admin server, 43
add refresh server, 45
create access group, 73
delete, 111
discover, 79
edit, 123
list, 205
refresh, 210, 211
remove admin server, 214
remove refresh server, 216
show, 237
file server access group
delete, 111
list, 205
file server plug-in
edit, 125
list, 205
show, 237
file system
add to access group, 47
create, 80
delete, 111
edit, 125
list, 205
refresh, 210
remove from access group, 218
show, 237
FileServer
  showcustomcmds, 240
FileServerPlugin
  showcustomcmds, 240
FileSystem
  showcustomcmds, 240

H
help, 197
  command, 9

I
import
  assembly, 198
  CDROM, 201
  disk, 202
  ISO, 201
  template, 199
  virtual appliance, 200
  virtual disk, 202
  virtual machine, 203
inbound migration lock
  Oracle VM Server, 134
IP address
  create, 162
  delete, 164
  edit, 165
IPMI
  configure, 87, 136
  delete, 111
  list, 205
  show, 237
IPS
  create repository, 93
  edit repository, 141
  show, 237
upgrade Oracle VM Server and Oracle VM Agent, 245
ISCSI
  multipath configuration, 42, 213
ISO
  delete, 111
  edit, 149
  import, 201
  list, 205
  show, 237
ISO file
  mapping to a virtual machine, 109

J
job
  abort, 41
  debug transcript, 171
  get info about queued job, 186
  list, 205
  list in date range, 182
  show, 237
Job
  showcustomcmds, 240

K
kill
  Oracle VM Server, 204
  virtual machine, 204

L
list
  command, 205
local network
  create, 82
lock
  timeout, 16
log file, 18

M
maintenance mode
  Oracle VM Server, 234
Manager
  edit, 127
  showcustomcmds, 240
master Oracle VM Server
  change, 137
message
  clear all received by virtual machine, 59
  clear all sent to virtual machine, 59
  clear received from virtual machine, 60
  clear sent to virtual machine, 61
  list received from virtual machine, 195
  list sent to virtual machine, 196
  send to virtual machine, 230
migrate
  virtual machine, 206, 208
move
  virtual machine, 209, 225
  virtual machine to Oracle VM Server, 55
  virtual machine to repository, 209
  virtual machine to server pool, 55
  virtual machine using a clone customizer, 209
multipath
  add access hosts, 42
  remove access hosts, 213
N
network
  add port, 49
  add port bond, 46, 76
  add VLAN interface, 54
  add VNIC, 56
  create, 82
  delete, 111
  edit, 127
  edit Ethernet port, 131
  edit port bond, 119
  list, 205
  remove bond port, 217
  remove Ethernet port, 220
  remove VLAN interface, 224
  remove VNIC, 226
  show, 237
Network
  showcustomcmds, 240
network bond
  add Ethernet port, 49
network port
  add to network, 49
  bond, 46, 76
  edit bond, 119
  remove bond port, 217
  remove Ethernet port, 220
NFS
  add refresh server, 45
  remove refresh server, 216
NFS file system
  add to access group, 47
  remove from access group, 218
NTP
  set on Oracle VM Server, 134

O
object types
  show, 241
operating system types
  get, 194
Oracle VM Agent
  change password, 57
Oracle VM Manager
  backup configuration, 170, 233
  edit, 127
  get time, 186
  list, 205
  show, 237
  statistics, 187, 235
Oracle VM Server
  add to access group, 49
  add to CPU compatibility group, 49
  add to repository, 49
  add to server pool, 49
  add virtual machine, 55
  anti affinity group, 75, 115
  change password, 57
  check for update, 58
  create port bond, 76
  delete, 111
  discover, 113
  edit, 134
  edit Ethernet port, 131
  edit port bond, 119
  inbound migration lock, 134
  IPMI, 87, 136
  kill, 204
  list, 205
  maintenance mode, 234
  NTP, 134
  refresh, 210, 211
  refresh storage, 212
  release ownership, 212
  remove from CPU compatibility group, 221
  remove from repository, 221
  remove from server pool, 221
  remove virtual machine, 225
  restart, 228
  role, 134
  show, 237
  start, 242
  stop, 243
  take ownership, 244
  upgrade, 245
  output mode
    set, 9, 9, 231
P
password
  change, 57
periodic task
  edit, 129
  list, 205
  show, 237
PeriodicTask
  showcustomcmds, 240
physical disk
  add to access group, 48
  clone to physical disk, 64
  clone to repository, 65
  clone to storage array, 67
  create, 83
  delete, 111
  edit, 130
  list, 205
refresh, 210
remove from access group, 219
resize, 227
show, 237

PhysicalDisk
showcustomcmds, 240

policy server
add, 44
remove, 215

Port
add bond, 46, 76
add to network, 49
create IP address, 162
delete IP address, 164
edit, 119, 131
edit IP address, 165
list, 205
remove, 220
remove bond, 217
SSH, 16

Repository
add Oracle VM Server, 49
create, 84
create server update, 93
delete, 111
edit, 132
list, 205
refresh, 210
remove Oracle VM Server, 221
show, 237

RepositoryExport
showcustomcmds, 240

RepositoryExport
add physical disk to access group, 48
remove physical disk from access group, 219

role
Oracle VM Server, 225, 225
virtual machine from server pool, 225

SAN storage
add physical disk to access group, 48
remove physical disk from access group, 219
script
samples, 13
server
refresh, 210
Server
showcustomcmds, 240
server controller
create, 87
delete, 111
edit, 136
list, 205
show, 237
server pool
  add Oracle VM Server, 49
  add policy server, 44
  add to access group, 51
  add to repository, 51
  add virtual machine, 55
  anti affinity group, 75, 115
  create, 88
  create network policy, 90
  create update group, 91
  create update repository, 93
  delete, 111
  edit, 137
  edit network policy, 139
  edit update group, 140, 141
  list, 205
  policy server, 44, 215
  remove from access group, 222
  remove from repository, 222
  remove Oracle VM Server, 221
  remove policy server, 215
  remove virtual machine, 225
  show, 237
server pool network policy
  delete, 111
  list, 205
  show, 237
server update group
  create, 91
  edit, 140, 141
server update repository
  create, 93
server upgrade group
  delete, 111
  list, 205
  show, 237
server upgrade repository
  delete, 111
  list, 205
  show, 237
ServerController
  showcustomcmds, 240
ServerPool
  showcustomcmds, 240
ServerPoolNetworkPolicy
  showcustomcmds, 240
ServerUpdateGroup
  showcustomcmds, 240
ServerUpdateRepository
  showcustomcmds, 240
session
  show CLI session information, 239
set
  command, 231
show
  CLI session information, 239
  command, 237
showallcustomcmds, 239
showcustomcmds, 240
showversion, 242
special characters
  escape, 9
SSH
  port, 16
  timeout, 6, 16
ssh
  background, 9
  script, 9
start
  Oracle VM Server, 242
  virtual machine, 242
statistics
  get, 187
  get average, 168
  get latest, 182
  get latest for list, 184
  get list for object, 187
  get list for object using query, 189
  set, 235
stop
  Oracle VM Server, 243
  virtual machine, 243
storage
  mapping storage, 107
  refresh, 212
storage array
  add admin server, 43
  create access group, 73
  delete, 111
  discover, 94
  edit, 142
  list, 205
  refresh, 210, 211
  remove admin server, 214
  show, 237
  validate, 246
storage array access group
  list, 205
storage array plug-in
  edit, 144
  list, 205
  show, 237
storage initiator
add, 52
list, 205
remove, 223
show, 237
storage repository
create, 84
present server pool, 51
release ownership, 212
take ownership, 244
unpresent server pool, 222
StorageArray
showcustomcmds, 240
StorageArrayPlugin
showcustomcmds, 240
StoragInitiator
showcustomcmds, 240
suspend
virtual machine, 244

T

tag
add, 53
create, 97
delete, 111
edit, 145
list, 205
remove, 224
show, 237
Tag
showcustomcmds, 240
take ownership
Oracle VM Server, 244
storage repository, 244
template
clonet, 62
import, 199
list, 205
time
gentrfor Oracle VM Manager, 186
set on Oracle VM Server, 134
timeout
locks, 16
set, 231
SSH, 6, 16
SSH public key, 16

U
unpresent
server pool from repository, 222
update
check for, 58
update group
create, 91
dd, 140, 141
update repository
create, 93
upgrade
check for update, 58
Oracle VM Server, 245

V
validate
storage array, 246
version
show, 242
virtual appliance
create virtual machine from, 71, 71, 72, 72
delete, 111
edit, 146, 147
export, 167
get descriptor, 171
import, 200
list, 205
refresh, 210
show, 237
virtual appliance virtual disk
list, 205
show, 237
virtual appliance virtual machine
edit, 148
list, 205
show, 237
virtual CDROM
clone to repository, 63
list, 205
virtual disk
adding to a virtual machine, 109, 109
clone to physical disk, 68
clone to repository, 69
create, 97
delete, 111
edit, 150
edit from assembly, 117
edit from virtual appliance, 147
import, 202
list, 205
mapping to a virtual machine, 109
resize, 227
show, 237
virtual machine
add to anti affinity group, 55
add to Oracle VM Server, 55
add to server pool, 55
adding a CDROM, 109
adding a clone customizer, 105
adding a virtual disk, 109
adding an ISO file, 109
clear all received messages, 58
clear all sent messages, 59
clear received message, 60
clear sent message, 61
close, 62
configuration file, 194
create, 100
create a clone customizer, 105
create from virtual appliance, 71, 72
create VNIC, 110
delete, 111
delete disk mapping, 111
disk mapping, 109
edit, 152
edit a clone customizer, 156
edit from assembly, 118
edit from virtual appliance, 148
import, 203
kill, 204
list, 205
list disk mapping, 205
list received messages, 195
list sent messages, 196
migrate, 206, 208
move, 209, 225
move to Oracle VM Server, 55
move to repository, 209
move to server pool, 55
operating system types, 194
remove from anti affinity group, 225
remove from Oracle VM Server, 225, 225
remove from server pool, 225
restart, 228
resume, 229
send message, 230
show, 237
show disk mapping, 237
start, 242
stop, 243
suspend, 244
virtual machine clone customizer
create, 105
delete, 111
edit, 156
list, 205
show, 237
virtual machine clone network mapping
delete, 111
list, 205
virtual machine clone storage mapping
delete, 111
list, 205
virtual machine disk mapping
edit, 159
virtual machine template
create from assembly, 70
list, 205
VirtualAppliance
showcustomcmds, 240
VirtualApplianceVirtualDisk
showcustomcmds, 240
VirtualApplianceVm
showcustomcmds, 240
VirtualCDROM
delete, 111
VirtualCDRom
show, 237
VirtualCdmrom
showcustomcmds, 240
VirtualDisk
showcustomcmds, 240
VLAN interface
add to network, 54
create, 99
create IP address, 162
delete, 111
delete IP address, 164
edit, 151
edit IP address, 165
list, 205
remove from network, 224
show, 237
VlanInterface
showcustomcmds, 240
vm
clear all received messages, 59
clear all sent messages, 59
clear received message, 60
clear sent message, 61
import, 203
list received messages, 195
list sent messages, 196
send message, 230
VM
configuration file, 194
show IP address, 237
Vm
showcustomcmds, 240
VmCloneCustomizer
showcustomcmds, 240
VmCloneNetworkMapping
showcustomcmds, 240
VmCloneStorageMapping
showcustomcmds, 240
VmDiskMapping
showcustomcmds, 240
VNIC
add to network, 56
create, 110
delete, 111
edit, 161
get address range, 196
list, 205
remove from network, 226
set address range, 236
show, 237
Vnic
  showcustomcmds, 240
volume group
  edit, 162
  list, 205
  show, 237
VolumeGroup
  showcustomcmds, 240

X
XML
  set output, 231

Y
YUM
  create repository, 93
  edit repository, 141
  show, 237
  upgrade Oracle VM Server, 245