

**Oracle® VM**  
**Command Line Interface User's Guide for Release 3.2**

**ORACLE®**

E35336-06  
March 2014

---

# Oracle® VM: Command Line Interface User's Guide for Release 3.2

Copyright © 2012, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

## Abstract

Document generated on: 2014-03-05 (revision: 3776)

---

---

# Table of Contents

Preface .....	xi
1 Audience .....	xi
2 Documentation Accessibility .....	xi
3 Related Documents .....	xi
4 Command Syntax .....	xi
5 Conventions .....	xii
1 Introduction to the Oracle VM Command Line Interface (CLI) .....	1
1.1 Connecting to the Oracle VM CLI .....	1
1.2 SSH Host Keys .....	3
1.3 Key-Based Authentication .....	4
1.4 Using the Oracle VM CLI and Getting Help .....	5
1.5 Passing in a CLI Command at the Command Line .....	8
1.6 Using Expect Scripts .....	9
1.6.1 Expect Script Prerequisites .....	9
1.6.2 Writing Expect Scripts .....	10
1.7 Sample Scripts .....	11
1.8 Integrating the CLI into Your Applications .....	14
1.9 Configuring the Oracle VM CLI .....	14
1.9.1 SSH Port .....	15
1.9.2 Timeout .....	15
1.9.3 Case Sensitivity .....	15
1.9.4 Public Key Authentication Expiry .....	15
1.9.5 Lock Exceptions .....	15
1.10 CLI Logs .....	15
I Using the CLI .....	17
2 Discovering Oracle VM Servers .....	21
2.1 Discovering an Oracle VM Server .....	21
3 Discovering Storage .....	23
3.1 Discovering a File Server .....	23
3.2 Discovering a SAN Server .....	24
4 Networking .....	27
4.1 To create a virtual machine network .....	27
4.2 To create VNICs .....	27
5 Managing Server Pools .....	29
5.1 To create a server pool .....	29
5.2 To add Oracle VM Servers to a server pool .....	29
6 Managing Storage Repositories .....	31
6.1 To create a storage repository on a file server .....	31
6.2 To create a storage repository on a SAN server .....	32
6.3 To add resources to a storage repository .....	33
6.3.1 Adding an assembly .....	33
6.3.2 Adding a virtual machine template .....	33
6.3.3 Adding an ISO file .....	33
7 Managing Virtual Machines .....	35
7.1 To create a virtual machine from a template: .....	35
7.2 To create a virtual machine from an assembly: .....	35
7.3 To create a virtual machine from an ISO: .....	35
7.4 To start a virtual machine: .....	37
II CLI Command Reference .....	39
A CLI Command Reference .....	43
A.1 abort Job .....	47

---

A.2	ackEvent .....	48
A.3	addAccessHost .....	49
A.4	addAvailableVnic Vm .....	50
A.5	addNfsRefreshServer .....	51
A.6	add BondPort .....	51
A.7	add FileSystem .....	52
A.8	add PhysicalDisk .....	53
A.9	add Port .....	54
A.10	add Server .....	55
A.11	add ServerPool .....	56
A.12	add StorageInitiator .....	57
A.13	add Tag .....	58
A.14	add Vm .....	59
A.15	add Vnic .....	60
A.16	clone VirtualDisk .....	60
A.17	clone Vm .....	61
A.18	create AccessGroup .....	62
A.19	create BondPort .....	63
A.20	create FileServer .....	65
A.21	create Network .....	66
A.22	create NfsAccessGroup .....	68
A.23	create PhysicalDisk .....	69
A.24	create Repository .....	70
A.25	create SanServer .....	71
A.26	create ServerPool .....	73
A.27	create Tag .....	74
A.28	create VirtualDisk .....	75
A.29	create VlanGroup .....	76
A.30	create VlanSegment .....	77
A.31	create Vm .....	78
A.32	create VmDiskMapping .....	80
A.33	create Vnic .....	82
A.34	createTemplatesFromAssembly .....	83
A.35	delete .....	83
A.36	discoverServer .....	85
A.37	edit AccessGroup .....	86
A.38	edit Assembly .....	87
A.39	edit BondPort .....	88
A.40	edit FileServer .....	89
A.41	edit FileSystem .....	90
A.42	edit Network .....	91
A.43	edit NfsAccessGroup .....	92
A.44	edit PhysicalDisk .....	93
A.45	edit Port .....	94
A.46	edit Repository .....	95
A.47	edit SanServer .....	96
A.48	edit Server .....	97
A.49	edit ServerPool .....	99
A.50	edit Tag .....	100
A.51	edit VirtualCdrom .....	101
A.52	edit VirtualDisk .....	102
A.53	edit VlanGroup .....	103
A.54	edit VlanInterface .....	104
A.55	edit Vm .....	104

---

A.56 edit VmDiskMapping .....	106
A.57 edit Vnic .....	107
A.58 edit VolumeGroup .....	108
A.59 edit YumConfig .....	109
A.60 exit .....	110
A.61 getEvents .....	110
A.62 getIncompatibleReasons .....	111
A.63 getJobsUsingRange .....	112
A.64 getLatestNumberOfJobs .....	113
A.65 help .....	113
A.66 importAssembly .....	114
A.67 importTemplate .....	115
A.68 importVirtualCdrom .....	116
A.69 importVirtualDisk .....	117
A.70 importVirtualMachine .....	118
A.71 initiateYumUpgrade .....	119
A.72 kill .....	119
A.73 list .....	120
A.74 migrate Vm .....	121
A.75 refresh .....	122
A.76 refreshStorageLayer .....	123
A.77 remove BondPort .....	124
A.78 remove FileSystem .....	125
A.79 remove PhysicalDisk .....	126
A.80 remove Port .....	126
A.81 remove Server .....	127
A.82 remove ServerPool .....	129
A.83 remove StorageInitiator .....	129
A.84 remove Tag .....	130
A.85 remove Vm .....	131
A.86 remove Vnic .....	132
A.87 removeAccessHost .....	133
A.88 removeNfsRefreshServer .....	134
A.89 restart .....	134
A.90 resume Vm .....	135
A.91 sendVmMessage .....	136
A.92 set .....	136
A.93 setNtp .....	137
A.94 show .....	138
A.95 showallcustomcmds .....	139
A.96 showcustomcmds .....	140
A.97 showNtp .....	141
A.98 showobjtypes .....	141
A.99 showversion .....	142
A.100 start .....	142
A.101 stop .....	143
A.102 suspend Vm .....	144
A.103 syncNtp .....	145
A.104 vnicCreate .....	145
Index .....	147



---

## List of Examples

1.1 list_server.exp Expect script .....	10
A.1 Aborting a job .....	48
A.2 Acknowledging an event .....	49
A.3 Adding a SAN server access host .....	49
A.4 Adding the next available VNIC to a virtual machine .....	50
A.5 Adding a refresh server to file server .....	51
A.6 Adding a bonded port to a network .....	52
A.7 Adding a file system to a file server access group .....	53
A.8 Adding a physical disk to a SAN storage access group .....	54
A.9 Adding an Ethernet port to a network .....	54
A.10 Adding an Oracle VM Server to a server pool .....	56
A.11 Adding an admin server to file server .....	56
A.12 Adding an admin server to a SAN server .....	56
A.13 Adding an Oracle VM Server to a file server access group .....	56
A.14 Adding (presenting) an Oracle VM Server to a storage repository .....	56
A.15 Presenting a storage repository to a server pool .....	57
A.16 Adding a storage initiator .....	58
A.17 Adding a tag to a server pool .....	58
A.18 Adding a virtual machine to an Oracle VM Server .....	59
A.19 Adding a VNIC to a virtual machine .....	60
A.20 Cloning a virtual disk .....	61
A.21 Cloning a virtual machine to a virtual machine .....	62
A.22 Cloning a virtual machine to a template .....	62
A.23 Cloning a template to a virtual machine .....	62
A.24 Cloning a template to a template .....	62
A.25 Creating an access group on a SAN server .....	63
A.26 Creating a bond port .....	64
A.27 Discovering a file server .....	66
A.28 Creating a network .....	67
A.29 Creating an access group on a file server .....	68
A.30 Creating a physical disk on a volume group .....	69
A.31 Creating a storage repository on a file server .....	71
A.32 Creating a storage repository on a physical disk .....	71
A.33 Discovering a SAN server .....	72
A.34 Creating a clustered server pool .....	74
A.35 Creating an unclustered server pool .....	74
A.36 Creating a tag .....	75
A.37 Creating a virtual disk in a storage repository .....	76
A.38 Creating a VLAN group .....	77
A.39 Creating a VLAN segment .....	78
A.40 Creating a virtual machine .....	80
A.41 Mapping a virtual disk to a virtual machine .....	81
A.42 Mapping an ISO file to a virtual machine .....	81
A.43 Mapping an empty CDROM drive .....	81
A.44 Creating a VNIC .....	82
A.45 Creating a virtual machine template from an assembly .....	83
A.46 Deleting an Oracle VM Server .....	84
A.47 Deleting a virtual machine or virtual machine template .....	84
A.48 Deleting a network .....	84
A.49 Deleting a virtual machine disk mapping .....	84
A.50 Discovering an Oracle VM Server .....	85

---

A.51 Editing an access group on a SAN server .....	86
A.52 Editing an assembly .....	87
A.53 Editing a bond port .....	88
A.54 Editing a file server .....	89
A.55 Editing a file system .....	90
A.56 Editing a network .....	91
A.57 Editing an access group on a file server .....	92
A.58 Editing a physical disk .....	93
A.59 Editing a port .....	94
A.60 Editing a storage repository .....	95
A.61 Editing a SAN server .....	97
A.62 Editing an Oracle VM Server .....	98
A.63 Taking ownership of an Oracle VM Server .....	98
A.64 Placing an Oracle VM Server into maintenance mode .....	98
A.65 Editing a server pool .....	99
A.66 Changing the master Oracle VM Server .....	99
A.67 Changing the virtual machine start policy .....	100
A.68 Editing a tag .....	100
A.69 Editing a ISO file/CDROM .....	101
A.70 Editing a virtual disk .....	102
A.71 Resizing a virtual disk .....	102
A.72 Editing a VLAN group .....	103
A.73 Editing a VLAN interface .....	104
A.74 Edit the disk boot order a virtual machine .....	106
A.75 Editing a virtual disk mapping object .....	107
A.76 Editing a VNIC .....	108
A.77 Editing a volume group .....	109
A.78 Editing a Yum repository .....	110
A.79 Exiting the CLI .....	110
A.80 Listing Oracle VM Server events .....	111
A.81 Listing virtual machine placement incompatibility reasons .....	112
A.82 Listing jobs in a date range .....	112
A.83 Listing recent jobs .....	113
A.84 Using the Help Command .....	114
A.85 Importing an assembly to a storage repository .....	114
A.86 Importing a virtual machine template to a storage repository .....	115
A.87 Importing a virtual CDROM/ISO file to a storage repository .....	116
A.88 Importing a virtual disk file to a storage repository .....	117
A.89 Importing a virtual machine to a storage repository .....	118
A.90 Upgrading an Oracle VM Server .....	119
A.91 Killing an Oracle VM Server .....	120
A.92 Killing a virtual machine .....	120
A.93 Listing Oracle VM Servers .....	121
A.94 Listing virtual machines and virtual machine templates .....	121
A.95 Listing networks .....	121
A.96 Listing virtual machine and virtual machine templates disk mapping .....	121
A.97 Migrating a virtual machine to an Oracle VM Server .....	122
A.98 Migrating a virtual machine to another server pool .....	122
A.99 Migrating a virtual machine to the Unassigned Virtual Machines folder .....	122
A.100 Refreshing a file server .....	123
A.101 Refreshing a SAN server .....	123
A.102 Refreshing a physical disk .....	123
A.103 Refresh a file system .....	123
A.104 Refresh a storage repository .....	123



---

A.105 Refreshing Oracle VM Server storage .....	124
A.106 Removing a bonded port from a network .....	124
A.107 Removing a file system from a file server access group .....	125
A.108 Removing physical disk from a SAN storage access group .....	126
A.109 Removing a port from a network .....	127
A.110 Removing an Oracle VM Server from a server pool .....	128
A.111 Removing an admin server from file server .....	128
A.112 Removing an admin server from a SAN server .....	128
A.113 Removing an Oracle VM Server from a file server access group .....	128
A.114 Removing (unpresenting) an Oracle VM Server from a storage repository .....	128
A.115 Unpresenting a storage repository from a server pool .....	129
A.116 Removing a storage initiator .....	130
A.117 Removing a tag from a server pool .....	131
A.118 Removing a virtual machine from an Oracle VM Server .....	132
A.119 Removing a VNIC from a virtual machine .....	132
A.120 Removing a SAN server access host .....	133
A.121 Removing a refresh server from a file server .....	134
A.122 Restarting an Oracle VM Server .....	135
A.123 Restarting a virtual machine .....	135
A.124 Resuming a virtual machine .....	136
A.125 Sending a message to a virtual machine .....	136
A.126 Setting end of line characters .....	137
A.127 Setting output mode to XML .....	137
A.128 Setting the list of NTP servers .....	138
A.129 Show details about an Oracle VM Server .....	139
A.130 Show details about a virtual machine .....	139
A.131 Show details about a network .....	139
A.132 Show Yum repository configuration .....	139
A.133 Show details about a VNIC .....	139
A.134 Showing all custom commands .....	140
A.135 Show Custom Commands for an Oracle VM Server .....	140
A.136 Show Custom Commands for a virtual machine .....	141
A.137 Show Custom Commands for a Repository .....	141
A.138 Showing the list of NTP servers .....	141
A.139 Showing all object types .....	142
A.140 Showing the CLI/Oracle VM Manager version number .....	142
A.141 Starting an Oracle VM Server .....	143
A.142 Starting a virtual machine .....	143
A.143 Stopping an Oracle VM Server .....	144
A.144 Stopping a virtual machine .....	144
A.145 Suspending a virtual machine .....	144
A.146 Pushing the list of NTP servers to Oracle VM Servers .....	145
A.147 Generating multiple VNICs .....	146



---

# Preface

The *Oracle VM Command Line Interface User's Guide* is your reference to the Oracle VM Command Line Interface (CLI). The CLI is intended to offer the same features as the Oracle VM Manager user interface, so you can manage your Oracle VM environment without using the user interface. 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.

## 1 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 User's Guide](#)), and that you are familiar with Oracle Linux system administration and Linux command line operation.

## 2 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## 3 Related Documents

For more information, see the following documents in the Oracle VM Release 3 documentation.

- [Oracle VM Release Notes](#)
- [Oracle VM Installation and Upgrade Guide](#)
- [Oracle VM Getting Started Guide](#)
- [Oracle VM User's Guide](#)
- [Oracle VM Windows Paravirtual Drivers Installation Guide](#)
- [Oracle VM Utilities Guide](#)
- [Oracle VM Security Guide](#)

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

<http://www.oracle.com/virtualization>

## 4 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:

Convention	Description
backslash \	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:  <pre>dd if=/dev/rdskc0t1d0s6 of=/dev/rst0 bs=10b \ count=10000</pre>
braces { }	Braces indicate required items:  <pre>.DEFINE {macro1}</pre>
brackets [ ]	Brackets indicate optional items:  <pre>cvtcrt <i>termname</i> [<i>outfile</i>]</pre>
ellipses ...	Ellipses indicate an arbitrary number of similar items:  <pre>CHKVAL fieldname value1 value2 ... valueN</pre>
<i>italics</i>	Italic type indicates a variable. Substitute a value for the variable:  <pre><i>library_name</i></pre>
vertical line	A vertical line indicates a choice within braces or brackets:  <pre>FILE <i>filesize</i> [K M]</pre>
forward slash /	A forward slash is used as an escape character in the CLI to escape the special characters ", ', ?, \, /, <, >:  <pre>create Tag name=MyTag description="HR/'s VMs"</pre>

## 5 Conventions

The following text conventions are used in this document:

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

---

# Chapter 1 Introduction to the Oracle VM Command Line Interface (CLI)

## Table of Contents

1.1 Connecting to the Oracle VM CLI .....	1
1.2 SSH Host Keys .....	3
1.3 Key-Based Authentication .....	4
1.4 Using the Oracle VM CLI and Getting Help .....	5
1.5 Passing in a CLI Command at the Command Line .....	8
1.6 Using Expect Scripts .....	9
1.6.1 Expect Script Prerequisites .....	9
1.6.2 Writing Expect Scripts .....	10
1.7 Sample Scripts .....	11
1.8 Integrating the CLI into Your Applications .....	14
1.9 Configuring the Oracle VM CLI .....	14
1.9.1 SSH Port .....	15
1.9.2 Timeout .....	15
1.9.3 Case Sensitivity .....	15
1.9.4 Public Key Authentication Expiry .....	15
1.9.5 Lock Exceptions .....	15
1.10 CLI Logs .....	15

## List of Examples

1.1 list_server.exp Expect script .....	10
---	----

The Oracle VM Command Line Interface (CLI) provides a command line interface to communicate with Oracle VM Manager. You can use the CLI to perform 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 CLI automatically starts and stops when you start or stop Oracle VM Manager. When you make changes to the Oracle VM environment using the CLI, these changes are reflected in real time in the Oracle VM Manager user interface.

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 Utilities Guide](#).

## 1.1 Connecting to the Oracle VM 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

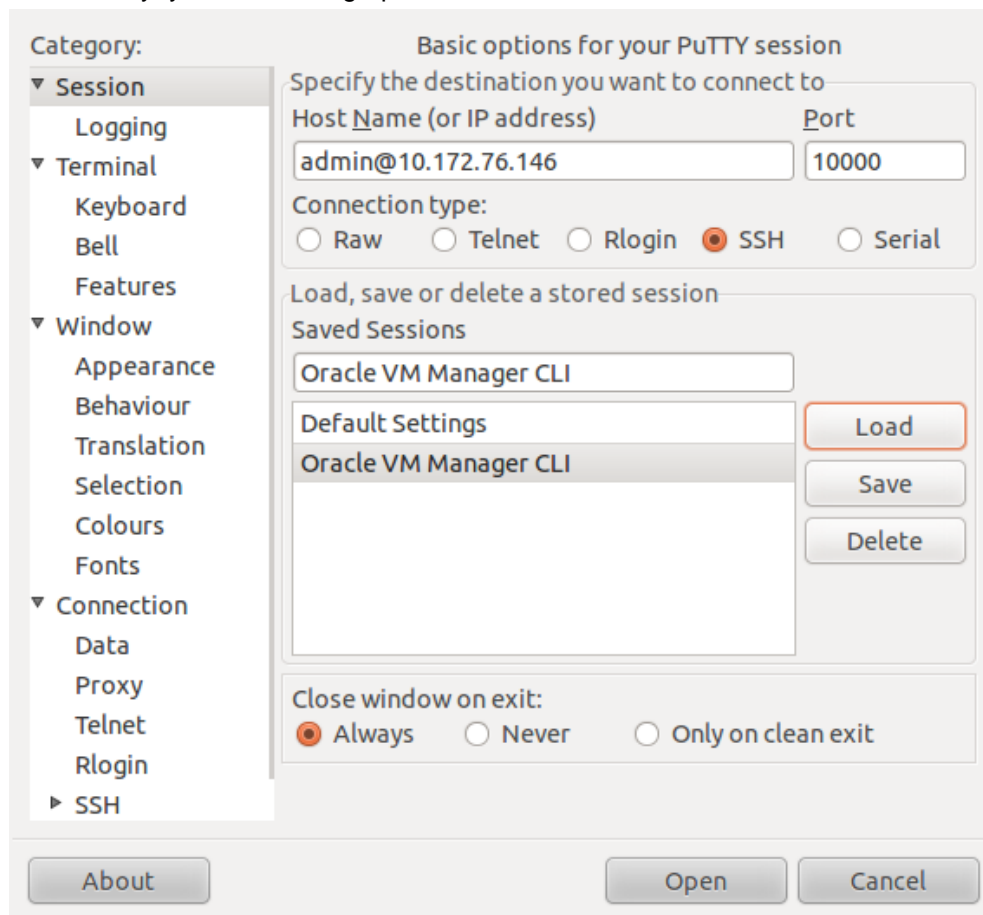
To keep your ssh session from disconnecting due to inactivity, you can use the `ssh ServerAliveInterval` option to send a null packet to the CLI to keep the connection alive. You can use this option, either on the command line when you enter the ssh command, or in the `~/.ssh/config` file, for example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

Alternatively, you can use a graphical SSH client like PuTTY:



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
OVM>
```

To exit the CLI, enter `exit`, or end the SSH session.

## 1.2 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.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '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
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@      WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!      @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the DSA host key has just been changed.
The fingerprint for the DSA key sent by the remote host is
fingerprint.
Please contact your system administrator.
Add correct host key in /root/.ssh/known_hosts to get rid of this message.
Offending key in ~/.ssh/known_hosts:1
DSA host key for localhost has changed and you have requested strict checking.
Host key verification failed.
```

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
```

You may need to add braces around the hostname if the previous command does not remove the entry, for example:

```
$ ssh-keygen -R [localhost]: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.

## 1.3 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.

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:

```
$ 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.

```
$ 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:

```
$ 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:

```
$ 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`).

```
$ 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.

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

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

You are requested to enter the admin user's password. Enter it.

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

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 1.9, “Configuring the Oracle VM CLI”](#) for information on how to change the public key authentication expiry time.

## 1.4 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.



### 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.92, “set”](#) for more information.

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

```
OVM> ?
add
create
delete
edit
exit
help
list
remove
set
show
showallcustomcmds
showcustomcmds
showobjtypes
showversion
OVM> list ?
    AccessGroup
    Assembly
    BondPort
    FileServer
    FileSystem
    Job
    Network
```

*Perhaps this is the command? Let's drill down further.*

```

PhysicalDisk
Port
Repository
SanServer
Server This looks like the command to use to list
Oracle VM Servers
ServerPool
StorageInitiator
Tag
VirtualCdrom
VirtualDisk
VlanGroup
VlanInterface
VlanSegment
Vm
VmDiskMapping
Vnic
VolumeGroup
OVM> list Server No more options can be entered so the results are automatically displayed
id:00:e0:81:4d:40:c6:00:e0:81:4d:40:c7:ff:ff:ff:ff name:MyServer1
id:00:e0:81:4d:5e:82:00:e0:81:4d:5e:83:ff:ff:ff:ff name:MyServer2
id:00:e0:81:4d:40:f5:00:e0:81:4d:40:be:00:e0:81:4d 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
exit
help
list
remove
set
show This looks like the command to use to show information
commands available for all objects
showallcustomcmds
showcustomcmds commands specific to an object (requires object as argument)
showobjtypes
showversion
OVM> show ?
AccessGroup
Assembly
BondPort
FileServer
FileSystem
Job
Network
PhysicalDisk
Port
Repository
SanServer
Server This looks like the command to use to show information
about Oracle VM Servers
ServerPool
StorageInitiator
Tag
VirtualCdrom
VirtualDisk
VlanGroup
VlanInterface
VlanSegment
Vm
VmDiskMapping
Vnic
VolumeGroup

```

```

YumConfig
OVM> show Server ?
           id=<object identifier> OR
           name=<object name>

```

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.

```

OVM> show Server name=MyServer1
Name = MyServer1
Id = 00:e0:81:4d:40:c6:00:e0:81:4d:40:c7:ff:ff:ff:ff
Status = Running
Mgmt MAC Address = 00:e0:81:4d:40:c6
Processors = 4
Ethernet Ports = 2
iSCSI Ports = 1
Maintenance Mode = Off
TakeOwnership = Yes
Role 1 = Utility Server
Role 2 = Virtual Machine Server
IP Address = 10.172.76.73
Processor Speed (GHz) = 1.995107
Memory (GiB) = 24.0
Dom0 Memory (GiB) = 0.96
FiberChannel Ports = 0
Network Failover Groups = 1
CPU Compatibility Group = Default_AMD_Family:15_Model:65
Server Architecture Type = x86-64b
Hypervisor Type = OVM/Xen
Manufacturer = empty
Serial Number = empty
Product Name = empty
Processor Family = AuthenticAMD
Processor Model = Dual-Core AMD Opteron(tm) Processor 2212
L1 Cache Size = 0
L2 Cache Size = 1024
L3 Cache Size = 0
BIOS Version = 'V1.02.B10 '
BIOS Release Date = 11/30/2007
Sockets Filled = 2
Manager UUID = 0004fb00000100008e2c477634f634c9
Ethernet Port 1 = 0004fb0000200000182e64895a3318ba [eth0 on MyServer1]
Ethernet Port 2 = 0004fb0000200000939f6e9ealae5562 [eth1 on MyServer1]
Bond Port 1 = 0004fb00002000002c9078c717247e87 [bond0 on MyServer1]
Physical Disk 1 = 0004fb0000180000707410381f2bfca5 [FreeBSD (2)]
Physical Disk 2 = 0004fb0000180000a0286fc8bc74ab94 [FreeBSD (10)]
Physical Disk 3 = 0004fb00001800008feee8ef5d311c1c [FreeBSD (5)]
Physical Disk 4 = 0004fb0000180000f33d1cd098bc0561 [FreeBSD (9)]
Physical Disk 5 = 0004fb0000180000df407c862d9924bb [FreeBSD (6)]
Physical Disk 6 = 0004fb000018000064cddacad73641a6 [FreeBSD (8)]
Physical Disk 7 = 0004fb0000180000d9b142fa8a4eb3cb [FreeBSD (1)]
Physical Disk 8 = 0004fb0000180000f195b73d4c15f64f [FreeBSD (7)]
Physical Disk 9 = 0004fb000018000043b42ad2924b9b48 [FreeBSD (3)]
Physical Disk 10 = 0004fb0000180000b9b3201691841100 [SATA_WDC_WD5001ABYS-_WD-WCAS86575890]
Storage Initiator 1 = iqn.1988-12.com.oracle:eaale2b65d3
Storage Initiator 2 = storage.LocalStorageInitiator in 00:e0:81:4d:40:c6:00:e0:81:4d:40:c7
:ff:ff:ff:ff
Version = 3.2.1-422
Vm 1 = 0004fb0000060000a0050840c71be743 [MyOL5VM]
Server Pool = 0004fb0000020000e2c7f63c784ba711 [MyServerPool]
OVM>

```

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 1.1 Helpful CLI commands**

Command/Feature	Description
?	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.
<code>help</code>	Displays the syntax to use for the top level commands.
<code>showallcustomcmds</code>	Displays a list of the all custom commands for all object types.
<code>showcustomcmds [object type]</code>	Displays a list of the custom commands for a specific object type provided as a parameter.
<code>showobjtypes</code>	Displays a list of the object types.
tab completion	Press the <b>Tab</b> key to auto-complete the command.
history	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.

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.

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

## 1.5 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 1.3, “Key-Based Authentication”](#) for information on setting up key-based authentication. Alternatively, you can use the sample `eovmcli` or `eovmclixml` Expect scripts on the Oracle VM Manager host to pass in the password without being prompted. These scripts return the results in either plain text, or XML. For example, to display a list of the Oracle VM Servers in plain text using the `eovmcli` Expect script, enter:

```
# cd /u01/app/oracle/ovm-manager-3/ovm_cli/expectscripts/
```

```
# ./eovmcli admin password "list Server"
Command: list Server
Status: Success
Time: date
Data:
  id:00:e0:81:4d:41:01:00:e0:81:4d:40:d6:00:e0:81:4d  name:MyServer1
  id:00:e0:81:4d:40:c6:00:e0:81:4d:40:c7:ff:ff:ff:ff  name:MyServer2
  id:00:e0:81:4d:40:f5:00:e0:81:4d:40:be:00:e0:81:4d  name:MyServer3
```

To display the output in XML, use the `eovmclixml` Expect script, for example:

```
# cd /u01/app/oracle/ovm-manager-3/ovm_cli/expectscripts/
# ./eovmclixml admin password "list Server"
<response>
  <command>
    list Server
  </command>
  <status>
    Success
  </status>
  <time>
    date
  </time>
  <data>
    <object value="id:00:e0:81:4d:41:01:00:e0:81:4d:40:d6:00:e0:81:4d  name:MyServer1" />
    <object value="id:00:e0:81:4d:40:c6:00:e0:81:4d:40:c7:ff:ff:ff:ff  name:MyServer2" />
    <object value="id:00:e0:81:4d:40:f5:00:e0:81:4d:40:be:00:e0:81:4d  name:MyServer3" />
  </data>
```

You can only use the semicolon to separate commands when using the `ssh` command. The `eovmcli` or `eovmclixml` Expect scripts do not accept the semicolon as a command delimiter.

If you want to pass in more complex commands to the CLI, you can write your own Expect scripts, as discussed in [Section 1.6.2, "Writing Expect Scripts"](#). You can also use any other programming language to write scripts using the CLI as discussed in [Section 1.8, "Integrating the CLI into Your Applications"](#).

## 1.6 Using Expect Scripts

Expect scripts can be used to automate multiple CLI commands in batch mode. Expect is a UNIX scripting and testing utility which can be used with SSH-based applications, like the Oracle VM CLI. Expect scripts can have any file name suffix you like, though they generally have an `.exp` extension. This guide and the sample scripts use either the `.sh` or no extension on Expect script file names. To run an Expect script, use the syntax:

```
expect script
```

Alternatively, make sure you have execute permission set on the Expect script, and run it as would any shell script; without explicitly specifying the `expect` command.

### 1.6.1 Expect Script Prerequisites

To use an Expect script, make sure you have the Expect utility installed. To install Expect using a Yum server on Oracle Linux, enter:

```
# yum install expect
```

To install Expect on other operating systems, see:

<http://expect.sourceforge.net/>

## 1.6.2 Writing Expect Scripts

You can write your own Expect scripts to perform batch jobs with the CLI. This section shows you how to create a very simple Expect script and run it with the CLI, but does not go into details about the programming languages used with Expect. For more information on writing Expect scripts, see:

<http://expect.sourceforge.net/>

A very simple Expect script that lists the Oracle VM Servers is listed in [Example 1.1, "list\\_server.exp Expect script"](#):

### Example 1.1 list\_server.exp Expect script

```
#!/usr/bin/expect

## Access CLI
set loginUser "admin"
set loginPassword "password"
set mgmtServerAddress manager_host

## Expect Parameters
set timeout 20
set successMsg "Status: Success"
set failureMsg "Status: Failure"

spawn ssh -l $loginUser $mgmtServerAddress -p 10000
expect_after eof {exit 0}

set timeout 10

##interact with SSH
##expect "yes/no" {send "yes\r"}
expect "password:" {send "$loginPassword\r"}
puts "\n## Starting Generated OVMCLI Script... ##\n"
set timeout 600

expect "OVM> "
send "set OutputMode=Verbose\r"
expect $successMsg {} \
    timeout { puts "\n\nTest Failure: \n\r"; exit}

expect "OVM> "
send "list Server\r"
expect $successMsg {} \
    timeout { puts "\n\nScript Failure: \n\r"; exit}
```

Edit the `loginUser`, `loginPassword` and `mgmtServerAddress` variables for your Oracle VM Manager environment.

To execute this Expect script file, use the `expect` command followed by the location of the Expect script, for example, enter:

```
# expect /myscripts/list_server.exp
spawn ssh -l admin localhost -p 10000
admin@localhost's password:
## Starting Generated OVMCLI Script... ##

OVM> set OutputMode=Verbose
Command: set OutputMode=Verbose
Status: Success
Time: date
```

```
OVM> list Server
Command: list Server
Status: Success
Time: date
Data:
  id:00:e0:81:4d:41:01:00:e0:81:4d:40:d6:00:e0:81:4d  name:MyServer1
  id:00:e0:81:4d:40:c6:00:e0:81:4d:40:c7:ff:ff:ff:ff  name:MyServer2
  id:00:e0:81:4d:40:f5:00:e0:81:4d:40:be:00:e0:81:4d  name:MyServer3
OVM>
```

## 1.7 Sample Scripts

There are a number of shell and Expect scripts provided with Oracle VM to help you use the CLI. These scripts are located in:

[/u01/app/oracle/ovm-manager-3/ovm\\_cli/expectscripts](/u01/app/oracle/ovm-manager-3/ovm_cli/expectscripts)



### Warning

Some of the sample scripts store the Oracle VM Manager login credentials in plain text. These scripts should not be used in a production environment, and especially should not be used on a machine with low security settings. You should implement your own security methods for the login credentials.

**Table 1.2 Sample scripts**

Script Name	Purpose
<code>eovmcli</code>	<p>An Expect script that passes a command to the CLI and returns the results in plain text. This script takes three parameters, <code>username</code>, <code>password</code> and <code>command</code>, for example:</p> <pre># ./eovmcli admin <i>password</i> "list Server"</pre> <p>This example returns a list of the Oracle VM Servers.</p> <p>See <a href="#">Section 1.8, "Integrating the CLI into Your Applications"</a> for information on using this script to integrate the CLI in other applications.</p>
<code>eovmclixml</code>	<p>An Expect script that passes a command to the CLI and returns the results in XML. This script takes three parameters, <code>username</code>, <code>password</code> and <code>command</code>, for example:</p> <pre># ./eovmclixml admin <i>password</i> "list Server"</pre> <p>This example returns a list of the Oracle VM Servers in XML.</p> <p>See <a href="#">Section 1.8, "Integrating the CLI into Your Applications"</a> for information on using this script to integrate the CLI in other applications.</p>
<code>ackAllEvents</code>	<p>An Expect script that acknowledges all events.</p> <p>To use this script, edit it and change the following lines to include the login credentials for Oracle VM Manager:</p> <pre>set username <i>username</i> set password <i>password</i></pre> <p>To run the script, enter:</p>

Script Name	Purpose
	# ./ackAllEvents
<code>inventory</code>	<p>An Expect script that displays an inventory of all objects managed by Oracle VM Manager.</p> <p>To use this script, edit it and change the following lines to include the login credentials for Oracle VM Manager:</p> <pre>set username <i>username</i> set password <i>password</i></pre> <p>To run the script, enter:</p> <pre># ./inventory</pre>

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

`/u01/app/oracle/ovm-manager-3/ovm_cli/expectscripts/createdeletescripts`

A text file named [README](#) in the directory provides information on using these scripts and may have additional information not included in this section.



**Warning**

Some of the sample scripts store the Oracle VM Manager login credentials in plain text. These scripts should not be used in a production environment, and especially should not be used on a machine with low security settings. You should implement your own security methods for the login credentials.

**Table 1.3 Sample create/delete scripts**

Script Name	Purpose
<code>create-fc-based-VM.cli</code>	<p>An Expect 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.</p> <p>To use this script, edit the <code>fc-based-VM.properties</code> file in the same directory and change the parameters to suit your environment. To run the script, enter:</p> <pre># ./runOVMCLITest -test=create-fc-based-VM.cli \ -arguments=fc-based-VM.properties</pre>
<code>delete-fc-based-VM.cli</code>	<p>An Expect script that deletes the set up created using the <code>create-fc-based-VM.cli</code> script.</p> <p>To use this script, edit the <code>fc-based-VM.properties</code> file in the same directory and change the parameters to suit your environment. To run the script, enter:</p> <pre># ./runOVMCLITest -test=delete-fc-based-VM.cli \ -arguments=fc-based-VM.properties</pre>
<code>create-iscsi-based-VM.cli</code>	<p>An Expect 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.</p>



## Sample Scripts

Script Name	Purpose
	<p>To use this script, edit the <code>iscsi-based-VM.properties</code> file in the same directory and change the parameters to suit your environment. To run the script, enter:</p> <pre># ./runOVMCLITest -test=create-iscsi-based-VM.cli \ -arguments=iscsi-based-VM.properties</pre>
<code>delete-iscsi-based-VM.cli</code>	<p>An Expect script that deletes the set up created using the <code>create-iscsi-based-VM.cli</code> script.</p> <p>To use this script, edit the <code>iscsi-based-VM.properties</code> file in the same directory and change the parameters to suit your environment. To run the script, enter:</p> <pre># ./runOVMCLITest -test=delete-iscsi-based-VM.cli \ -arguments=iscsi-based-VM.properties</pre>
<code>create-nfs-based-VM.cli</code>	<p>An Expect 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.</p> <p>To use this script, edit the <code>nfs-based-VM.properties</code> file in the same directory and change the parameters to suit your environment. To run the script, enter:</p> <pre># ./runOVMCLITest -test=create-nfs-based-VM.cli \ -arguments=nfs-based-VM.properties</pre>
<code>delete-nfs-based-VM.cli</code>	<p>An Expect script that deletes the set up created using the <code>create-nfs-based-VM.cli</code> script.</p> <p>To use this script, edit the <code>nfs-based-VM.properties</code> file in the same directory and change the parameters to suit your environment. To run the script, enter:</p> <pre># ./runOVMCLITest -test=delete-nfs-based-VM.cli \ -arguments=nfs-based-VM.properties</pre>
<code>runAllCreateDeleteTests.sh</code>	<p>An Expect script that runs all the create/delete scripts provided in the directory.</p> <p>To use this script, edit the <code>*-based-VM.properties</code> files in the same directory and change the parameters to suit your environment. To run the script, enter:</p> <pre># ./runAllCreateDeleteTests.sh</pre>
<code>runOVMCLITest</code>	<p>A shell script to run the <code>*-based-VM.cli</code> Expect scripts which create/delete the Oracle VM environment. To run the script, enter:</p> <pre># ./runOVMCLITest -test=<i>cli_script</i> -arguments=<i>properties_file</i></pre>
<code>commonExpectDef.cli</code>	<p>An Expect script that contains the common functions used by other scripts in this directory. Do not run this script directly.</p>
<code>runCLI.py</code>	<p>A Python script used by <code>runOVMCLITest</code> that reads the arguments defined in the property files and generates Expect scripts. Do not run this script directly.</p>

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 an Expect script, the output is displayed to the screen; spool it to a file or other process or script as required.

## 1.8 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.

There are two Expect scripts, `eovmcli` and `eovmcli.xml` that enable you to easily execute CLI commands from the operating system shell. All you need to do is install the Expect RPM (free) on your Oracle VM Manager host and you are ready to execute scripts against the CLI.

The `eovmcli.xml` script take three command line arguments (`username`, `password` and `command`, for example:

```
# eovmcli.xml "admin" "password" "list Server"
```

The results are returned in XML format at the shell prompt. XML is useful for parsing the results in another application.

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.

The `eovmcli` script does the same thing but *pretty prints* the results to the screen, just like when you run the CLI using an SSH connection.

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.

## 1.9 Configuring the Oracle VM CLI

When Oracle VM Manager starts, it reads the CLI configuration file. The configuration file is located at:

```
/u01/app/oracle/ovm-manager-3/ovm_cli/config/CLIConfigParams.xml
```

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 Oracle VM Manager for the changes to take effect. To stop Oracle VM Manager, enter:

```
# service ovmm stop
```

To start Oracle VM Manager, enter:

```
# service ovmm start
```

## 1.9.1 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.

## 1.9.2 Timeout

By default, connections to the CLI time out after 15 minutes. To change the timeout period for connections to the CLI client, edit the `clientInactivityTimeout="15"` line in the configuration file. Set the time out to be between 1-59 minutes. If you do not want the CLI to time out, set this option to `-1`.

In addition to this timeout option, you can use the `ssh ServerAliveInterval` option to stop an ssh client from timing out. See the [Tip](#) in [Section 1.1, "Connecting to the Oracle VM CLI"](#) for more information on using this option.

## 1.9.3 Case Sensitivity

CLI commands are not case sensitive, so you can enter `list vm`, `List VM`, or any other variation in case. By default, 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`. You can change the case sensitivity to make data values case insensitive, using the `dataCaseSensitive="yes"` line in the configuration file, and changing the option to `"no"`.

## 1.9.4 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.

## 1.9.5 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.

## 1.10 CLI Logs

A log of the CLI application is available in:

`/u01/app/oracle/ovm-manager-3/machine1/base_adf_domain/servers/AdminServer/  
logs/CLI.log`

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

`/u01/app/oracle/ovm-manager-3/machine1/base_adf_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 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 Getting Started Guide](#) for overview information. If you need more detailed information on a step, see the [Oracle VM User's Guide](#).

---

---

---

---

## Table of Contents

2	Discovering Oracle VM Servers .....	21
2.1	Discovering an Oracle VM Server .....	21
3	Discovering Storage .....	23
3.1	Discovering a File Server .....	23
3.2	Discovering a SAN Server .....	24
4	Networking .....	27
4.1	To create a virtual machine network .....	27
4.2	To create VNICs .....	27
5	Managing Server Pools .....	29
5.1	To create a server pool .....	29
5.2	To add Oracle VM Servers to a server pool .....	29
6	Managing Storage Repositories .....	31
6.1	To create a storage repository on a file server .....	31
6.2	To create a storage repository on a SAN server .....	32
6.3	To add resources to a storage repository .....	33
6.3.1	Adding an assembly .....	33
6.3.2	Adding a virtual machine template .....	33
6.3.3	Adding an ISO file .....	33
7	Managing Virtual Machines .....	35
7.1	To create a virtual machine from a template: .....	35
7.2	To create a virtual machine from an assembly: .....	35
7.3	To create a virtual machine from an ISO: .....	35
7.4	To start a virtual machine: .....	37





---

## Chapter 2 Discovering Oracle VM Servers

### Table of Contents

2.1 Discovering an Oracle VM Server .....	21
---	----

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

### 2.1 Discovering an Oracle VM Server

To discover an Oracle VM Server use the following syntax:

```
discoverServer ipAddress= { ip_address | hostname } port=port_number  
username=username password=password takeOwnership= { yes | no }
```

For example:

```
OVM> discoverServer ipAddress=10.172.76.73 port=8899 username=oracle password=password \  
takeOwnership=yes
```

For more information on the syntax and usage, see [Section A.36, “discoverServer”](#).



---

## Chapter 3 Discovering Storage

### Table of Contents

3.1 Discovering a File Server .....	23
3.2 Discovering a SAN Server .....	24

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

### 3.1 Discovering a File Server

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

```
create FileServer plugin=name accessHost=value [ adminHost=value
adminUserName=value adminPassword=value ][ uniformedExports= { yes | no } ][
name=value ][ description=value ]
```

For example:

```
OVM> create FileServer plugin="Oracle Generic Network File System" \
accessHost=10.172.76.125 name=MyNFSServer
```

For more information on the syntax and usage, see [Section A.20, “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:

```
add Server instance to { ServerPool | FileServer | SanServer | Repository |
NfsAccessGroup } instance
```

For example:

```
OVM> add Server name=MyServer to FileServer name=MyNFSServer
```

For more information on the syntax and usage, see [Section A.10, “add Server”](#) .

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:

```
addNfsRefreshServer FileServer instance nfsRefreshServer=value
```

For example:

```
OVM> addNfsRefreshServer FileServer name=MyNFSServer nfsRefreshServer=MyServer1
```

For more information on the syntax and usage, see [Section A.5, “addNfsRefreshServer”](#) .

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 { Assembly | FileServer | FileSystem | PhysicalDisk | Repository | SanServer
| Server } instance
```

For example:

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

For more information on the syntax and usage, see [Section A.75, “refresh”](#).

## 3.2 Discovering a SAN Server

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

```
create SanServer plugin=name storageType= { FibreChannelStorageArray |
iSCSIStorageArray } [accessHost=value] [accessPort=value] [ accessUsername=value
accessPassword=value ] [useChap= { yes | no } ] [ adminHost=value adminUserName=value
adminPassword=value ] name=value [description=value]
```

For example:

```
OVM> create SanServer plugin="Oracle Generic SCSI Plugin(1.2.1)" name=MyISCSIServer \
storageType=iSCSIStorageArray accessHost=10.172.76.130 accessport=3260
```

For more information on the syntax and usage, see [Section A.25, “create SanServer”](#).

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

```
add Server instance to { ServerPool | FileServer | SanServer | Repository |
NfsAccessGroup } instance
```

For example:

```
OVM> add Server name=MyServer to SanServer name=MyISCSIServer
```

For more information on the syntax and usage, see [Section A.10, “add Server”](#).

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 SAN server is discovered. First, find the name of the access group by listing the access groups for the server using the `show SanServer` command, for example:

```
OVM> show SanServer name=MyISCSIServer
Name = MyISCSIServer
Id = 0004fb00000900001426b77f079c83ea
Storage Plug-in = Oracle Generic SCSI Plugin(1.2.1)
Use Chap = No
Access Host 1 = 10.172.76.130
Access Port 1 = 3260
Storage Type = iSCSI Storage Server
Allocated (GiB) = 0.0
Free (GiB) = 0.0
Used (GiB) = 0.0
Total (GiB) = 0.0
Status = online
description = Generic iSCSI Storage
Admin Server 1 = 00:e0:81:4d:5f:2f:00:e0:81:4d:29:ee:00:e0:81:4d [MyServer6]
Admin Server 2 = 00:e0:81:4d:41:01:00:e0:81:4d:40:d6:00:e0:81:4d [MyServer5]
Admin Server 3 = 00:e0:81:4d:41:05:00:e0:81:4d:40:de:00:e0:81:4d [MyServer4]
Admin Server 4 = 00:e0:81:4d:5e:16:00:e0:81:4d:5e:17:ff:ff:ff:ff [MyServer9]
Admin Server 5 = 00:e0:81:4d:40:c6:00:e0:81:4d:40:c7:ff:ff:ff:ff [MyServer1]
Volume Group 1 = Generic_iSCSI_Volume_Group @ 0004fb00000900001426b77f079c83ea
[Generic_iSCSI_Volume_Group]
Access Group 1 = Default access group @ MyISCSIServer @ 0004fb00000900001426b77f079c83ea
[Default access group @ MyISCSIServer] Access group name
```

OVM>

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

```
OVM> show Server name=MyServer1
Name = MyServer1
Id = 00:e0:81:4d:40:c6:00:e0:81:4d:40:c7:ff:ff:ff:ff
Status = Running
Mgmt MAC Address = 00:e0:81:4d:40:c6
Processors = 4
Ethernet Ports = 2
iSCSI Ports = 1
Maintenance Mode = Off
TakeOwnership = Yes
Role 1 = Utility Server
Role 2 = Virtual Machine Server
IP Address = 10.172.76.73
Processor Speed (GHz) = 1.995106
Memory (GB) = 23
Dom0 Memory (MB) = 987
FiberChannel Ports = 0
Network Failover Groups = 1
CPU Compatibility Group = Default_AMD_Family:15_Model:65
Server Architecture Type = x86-64b
Hypervisor Type = OVM/Xen
Manufacturer = empty
Serial Number = empty
Product Name = empty
Processor Family = AuthenticAMD
Processor Model = Dual-Core AMD Opteron(tm) Processor 2212
L1 Cache Size = 0
L2 Cache Size = 1024
L3 Cache Size = 0
BIOS Version = 'V1.02.B10 '
BIOS Release Date = 11/30/2007
Sockets Filled = 2
Manager UUID = 0004fb00000100008e2c477634f634c9
Ethernet Port 1 = 0004fb00002000002e8cb0ebad4efcb3 [eth0 on MyServer1]
Ethernet Port 2 = 0004fb0000200000681d3335a81e771a [eth1 on MyServer1]
Bond Port 1 = 0004fb0000200000afbcf70ade70d69d [bond0 on MyServer1]
Physical Disk 1 = 0004fb0000180000f9029d0c23534229 [SATA_WDC_WD5001ABYS-_WD-WCAS86575890]
Storage Initiator 1 = iqn.1988-12.com.oracle:d72d82d0817f Storage initiator name
Storage Initiator 2 = storage.LocalStorageInitiator in
00:e0:81:4d:40:c6:00:e0:81:4d:40:c7:ff:ff:ff:ff
OVM>
```

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 \
name='Default access group @ MyISCSIServer'
```

For more information on the syntax and usage, see [Section A.12, “add StorageInitiator”](#) .

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

`refresh { Assembly | FileServer | FileSystem | PhysicalDisk | Repository | SanServer | Server } instance`

For example:

```
OVM> refresh SanServer name=MyISCServer
```

For more information on the syntax and usage, see [Section A.75, "refresh"](#) .

---

## Chapter 4 Networking

### Table of Contents

4.1 To create a virtual machine network .....	27
4.2 To create VNICs .....	27

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

### 4.1 To create 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 }] [vlanSegment=value] [server=value] name=value  
[description=value]
```

For example:

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

For more information on the syntax and usage, see [Section A.21, “create Network”](#).

2. Add a port from each Oracle VM Server to the network using the syntax:

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

For example:

```
OVM> add Port id=0004fb0000200000ed75de88c4dcb296 to Network name=MyVMNetwork
```

For more information on the syntax and usage, see [Section A.9, “add Port”](#).

### 4.2 To create VNICs

Create a VNIC on a network using the syntax:

```
create Vnic network=value name=value [description=value]
```

For example:

```
OVM> create Vnic name=00:21:f6:00:00:18 network=MyVMNetwork
```

For more information on the syntax and usage, see [Section A.33, “create Vnic”](#).





---

# Chapter 5 Managing Server Pools

## Table of Contents

5.1 To create a server pool .....	29
5.2 To add Oracle VM Servers to a server pool .....	29

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

## 5.1 To create 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. 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, use the syntax:

```
refresh { Assembly | FileServer | FileSystem | PhysicalDisk | Repository | SanServer | Server } instance
```

For example, to refresh a physical disk:

```
OVM> refresh PhysicalDisk id=0004fb0000180000efa8fd003a5f1613
```

And to refresh a file system:

```
OVM> refresh FileSystem name=nfs:/mnt/voll/poolfs01
```

For more information on the syntax and usage, see [Section A.75, "refresh"](#).

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

```
create ServerPool virtualIP=value clusterEnable= { yes | no } filesystem=value physicalDisk=value keymapName=value migrateUsingSsl= { yes | no } [startPolicy= { BEST_SERVER | CURRENT_SERVER }] name=value [description=value]
```

For example to create a clustered server pool:

```
OVM> create ServerPool virtualIP=10.172.77.195 clusterEnable=yes \ filesystem=nfs:/mnt/voll/poolfs01 name=MyServerPool description='Clustered server pool'
```

And to create an unclustered server pool:

```
OVM> create ServerPool virtualIP=10.172.77.195 clusterEnable=no name=MyServerPool \ description='Unclustered server pool'
```

For more information on the syntax and usage, see [Section A.26, "create ServerPool"](#).

## 5.2 To add Oracle VM Servers to a server pool

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

```
add Server instance to { ServerPool | FileServer | SanServer | Repository | NfsAccessGroup } instance
```

For example:

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

For more information on the syntax and usage, see [Section A.10, “add Server”](#).

---

# Chapter 6 Managing Storage Repositories

## Table of Contents

6.1 To create a storage repository on a file server .....	31
6.2 To create a storage repository on a SAN server .....	32
6.3 To add resources to a storage repository .....	33
6.3.1 Adding an assembly .....	33
6.3.2 Adding a virtual machine template .....	33
6.3.3 Adding an ISO file .....	33

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

## 6.1 To create a storage repository on a file server

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

```
OVM> list FileSystem
id:0004fb0000050000002618dec56ee0e8 name:0004fb0000050000002618dec56ee0e8
id:0004fb00000500007d0ab143ddfe4467 name:0004fb00000500007d0ab143ddfe4467
id:0004fb0000050000d426fb8e37a84bbc name:0004fb0000050000d426fb8e37a84bbc
id:0004fb0000050000014a27bcf721326e name:0004fb0000050000014a27bcf721326e
id:bcfca7ee-e3e5-4e17-b89f-330924f4b8e0 name:nfs:/mnt/vol1/poolfs01
id:8040d5f1-810f-4f44-8c46-35639113385b name:nfs:/mnt/vol1/poolfs02
id:cd3df8d5-ee75-4c7d-ala1-4bb40bf2eadb name:nfs:/mnt/vol1/poolfs03
id:29cc47bd-cf4a-426e-a89e-771d44b473cc name:nfs:/mnt/vol1/poolfs04
id:9f06d8ce-0059-4767-b9d9-65075a7762cc name:nfs:/mnt/vol1/repo01
id:cdf33132-d692-4abd-9e12-79ae078bb6d2 name:nfs:/mnt/vol2/repo02
id:de6289b7-168f-4d5a-827f-e4e44ddb45 name:nfs:/mnt/vol2/repo03
OVM>
```

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

```
refresh { Assembly | FileServer | FileSystem | PhysicalDisk | Repository | SanServer
| Server } instance
```

For example:

```
OVM> refresh FileSystem name=nfs:/mnt/vol1/poolfs01
```

For more information on the syntax and usage, see [Section A.75, “refresh”](#).

3. Create the storage repository. Use the syntax:

```
create Repository { fileSystem=value | serverPool=value physicalDisk=value } [
sharePath=value ] name=value [ description=value ]
```

For example:

```
OVM> create Repository name=MyRepository fileSystem=nfs:/mnt/vol1/repo01
```

For more information on the syntax and usage, see [Section A.24, “create Repository”](#).

4. To grant access to the storage repository to any Oracle VM Servers, you must *present* the Oracle VM Server to the storage repository. To present Oracle VM Servers or all the members of a server pool to a storage repository, use the syntax:

```
add Server instance to { ServerPool | FileServer | SanServer | Repository |
NfsAccessGroup } instance
```

For example:

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

For more information on the syntax and usage, see [Section A.10, “add Server”](#) .

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

```
refresh { Assembly | FileServer | FileSystem | PhysicalDisk | Repository | SanServer
| Server } instance
```

For example:

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

For more information on the syntax and usage, see [Section A.75, “refresh”](#) .

## 6.2 To create a storage repository on a SAN server

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

```
OVM> list PhysicalDisk
id:0004fb000018000056ae6a85d4474461 name:SATA_WDC_WD5001ABYS-_WD-WCAS86257005
id:0004fb00001800009896ed1ca8e3ee5c name:SATA_WDC_WD5001ABYS-_WD-WCAS86571931
id:0004fb000018000051a357cffe313c2 name:FreeBSD (1)
id:0004fb0000180000647fb9054783d1c4 name:FreeBSD (2)
id:0004fb00001800009b72e3fb0c468142 name:FreeBSD (3)
id:0004fb0000180000ba63ec88aa330640 name:FreeBSD (4)
id:0004fb0000180000a042d897ecbf8b2c name:FreeBSD (5)
id:0004fb000018000091194eed2490e09d name:FreeBSD (6)
id:0004fb000018000041883f8012028079 name:FreeBSD (7)
id:0004fb000018000060759fe97b407452 name:FreeBSD (8)
id:0004fb0000180000aaad232b0daa0f97 name:FreeBSD (9)
id:0004fb00001800009ebad6fd7ad4e087 name:SATA_WDC_WD5001ABYS-_WD-WCAS86288968
id:0004fb00001800006abba7ddb7373a3 name:SATA_WDC_WD5001ABYS-_WD-WCAS86287217
id:0004fb00001800003e85e8167af61c97 name:SATA_WDC_WD5001ABYS-_WD-WCAS86575561
id:0004fb00001800005ce71884046579a4 name:SATA_WDC_WD5001ABYS-_WD-WCAS86578492
id:0004fb0000180000d566f3335e919303 name:SATA_WDC_WD5001ABYS-_WD-WCAS86575890
OVM>
```

2. Create the storage repository. Use the syntax:

```
create Repository { fileSystem=value | serverPool=value physicalDisk=value } [
sharePath=value ] name=value [ description=value ]
```

For example:

```
OVM> create Repository name=MySANRepository serverPool=MyServerPool \
physicalDisk=0004fb0000180000ba63ec88aa330640
```

For more information on the syntax and usage, see [Section A.24, “create Repository”](#) .

3. To grant access to the storage repository to any Oracle VM Servers, you must *present* the Oracle VM Server to the storage repository. To present Oracle VM Servers or all the members of a server pool to a storage repository, use the syntax:

```
add Server instance to { ServerPool | FileServer | SanServer | Repository |  
NfsAccessGroup } instance
```

For example:

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

For more information on the syntax and usage, see [Section A.10, “add Server”](#).

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

```
refresh { Assembly | FileServer | FileSystem | PhysicalDisk | Repository | SanServer  
| Server } instance
```

For example:

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

For more information on the syntax and usage, see [Section A.75, “refresh”](#).

## 6.3 To add resources to a storage repository

### 6.3.1 Adding an assembly

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

```
importAssembly Repository instance server= { id | name } url=value
```

For example:

```
OVM> importAssembly Repository name=MyRepository server=MyServer \  
url=http://example.com/assemblies/myassembly.ova
```

For more information on the syntax and usage, see [Section A.66, “importAssembly”](#).

### 6.3.2 Adding a virtual machine template

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

```
importTemplate Repository=instance server= { id | name } url=value
```

For example:

```
OVM> importTemplate Repository name=MyRepository server=MyServer \  
url=http://example.com/OVM_OL5U6_X86_64_PVM_10GB.tgz
```

For more information on the syntax and usage, see [Section A.67, “importTemplate”](#).

### 6.3.3 Adding an ISO file

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

```
importVirtualCdrom Repository=instance server= { id | name } url=value
```

For example:

```
OVM> importVirtualCdrom Repository name=MyRepository server=MyServer \  
url=http://example.com/iso/MyIso.iso
```

```
url=http://example.com/isos/myiso.iso
```

For more information on the syntax and usage, see [Section A.68](#), “importVirtualCdrom”.

---

# Chapter 7 Managing Virtual Machines

## Table of Contents

7.1 To create a virtual machine from a template: .....	35
7.2 To create a virtual machine from an assembly: .....	35
7.3 To create a virtual machine from an ISO: .....	35
7.4 To start a virtual machine: .....	37

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

### 7.1 To create 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
```

For example:

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

For more information on the syntax and usage, see [Section A.17, “clone Vm”](#).

### 7.2 To create a virtual machine from an assembly:

1. Create a one or more templates from an assembly file using the syntax:

```
createTemplatesFromAssembly assembly= { id | name }
```

For example:

```
OVM> createTemplatesFromAssembly assembly=myassembly.ova
```

For more information on the syntax and usage, see [Section A.34, “createTemplatesFromAssembly”](#).

2. Clone the template to a virtual machine as shown in [Section 7.1, “To create a virtual machine from a template.”](#)

### 7.3 To create a virtual machine from an ISO:

1. Create a virtual machine using the syntax:

```
create Vm memory=value memoryLimit=value cpuCountLimit=value cpuCount=value  
cpuPriority=value cpuUtilizationCap=value highAvailability= { yes | no }  
osType= { WIN_2003 | WIN_2008 | WIN_7 | WIN_VISTA | OTHER_WIN | OL_4 | OL_5 | OL_6  
| RHL_4 | RHL_5 | RHL_6 | LINUX_RECOVERY | OTHER_LINUX | SOLARIS_10 | SOLARIS_11 |  
OTHER_SOLARIS | NONE }  
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 } networkBootPath=value repository=value  
name=value [description=value] { on Server instance | ServerPool instance }
```

For example:

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

For more information on the syntax and usage, see [Section A.31, “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.28, “create VirtualDisk”](#).

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

```
create VmDiskMapping slot=value [storageDevice=value] name=value  
[description=value] on Vm instance
```

For example:

```
OVM> create VmDiskMapping slot=0 storageDevice=MyVMDisk name=BootDisk on Vm name=MyVM
```

For more information on the syntax and usage, see [Section A.32, “create VmDiskMapping”](#).

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

```
create VmDiskMapping slot=value [storageDevice=value] name=value  
[description=value] on Vm instance
```

For example:

```
OVM> create VmDiskMapping slot=1 storageDevice=OracleLinux-R6-U2-Server-x86_64-dvd.iso \  
name=CDROM on Vm name=MyVM
```

For more information on the syntax and usage, see [Section A.32, “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:

```
edit Vm instance [memory=value] [memoryLimit=value] [cpuCountLimit=value]  
[cpuCount=value] [cpuPriority=value] [cpuUtilizationCap=value] [highAvailability=  
{yes | no }]  
[osType= {WIN_2003 | WIN_2008 | WIN_7 | WIN_VISTA | OTHER_WIN | OL_4 | OL_5  
| OL_6 | RHL_4 | RHL_5 | RHL_6 | LINUX_RECOVERY | OTHER_LINUX | SOLARIS_10 |  
SOLARIS_11 | OTHER_SOLARIS | NONE }]  
[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 }][networkBootPath=value][name=value]  
[description=value]
```



For example:

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

For more information on the syntax and usage, see [Section A.55, “edit Vm”](#).

6. Add a VNIC to the virtual machine using the syntax:

```
add Vnic instance to Vm instance
```

For example:

```
OVM> add Vnic name=00:21:f6:00:00:00 to Vm name=MyVM
```

For more information on the syntax and usage, see [Section A.15, “add Vnic”](#).

## 7.4 To start 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.100, “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.

---

---

---

---

## Table of Contents

A CLI Command Reference .....	43
A.1 abort Job .....	47
A.2 ackEvent .....	48
A.3 addAccessHost .....	49
A.4 addAvailableVnic Vm .....	50
A.5 addNfsRefreshServer .....	51
A.6 add BondPort .....	51
A.7 add FileSystem .....	52
A.8 add PhysicalDisk .....	53
A.9 add Port .....	54
A.10 add Server .....	55
A.11 add ServerPool .....	56
A.12 add StorageInitiator .....	57
A.13 add Tag .....	58
A.14 add Vm .....	59
A.15 add Vnic .....	60
A.16 clone VirtualDisk .....	60
A.17 clone Vm .....	61
A.18 create AccessGroup .....	62
A.19 create BondPort .....	63
A.20 create FileServer .....	65
A.21 create Network .....	66
A.22 create NfsAccessGroup .....	68
A.23 create PhysicalDisk .....	69
A.24 create Repository .....	70
A.25 create SanServer .....	71
A.26 create ServerPool .....	73
A.27 create Tag .....	74
A.28 create VirtualDisk .....	75
A.29 create VlanGroup .....	76
A.30 create VlanSegment .....	77
A.31 create Vm .....	78
A.32 create VmDiskMapping .....	80
A.33 create Vnic .....	82
A.34 createTemplatesFromAssembly .....	83
A.35 delete .....	83
A.36 discoverServer .....	85
A.37 edit AccessGroup .....	86
A.38 edit Assembly .....	87
A.39 edit BondPort .....	88
A.40 edit FileServer .....	89
A.41 edit FileSystem .....	90
A.42 edit Network .....	91
A.43 edit NfsAccessGroup .....	92
A.44 edit PhysicalDisk .....	93
A.45 edit Port .....	94
A.46 edit Repository .....	95
A.47 edit SanServer .....	96
A.48 edit Server .....	97
A.49 edit ServerPool .....	99
A.50 edit Tag .....	100

---

A.51 edit VirtualCdrom .....	101
A.52 edit VirtualDisk .....	102
A.53 edit VlanGroup .....	103
A.54 edit VlanInterface .....	104
A.55 edit Vm .....	104
A.56 edit VmDiskMapping .....	106
A.57 edit Vnic .....	107
A.58 edit VolumeGroup .....	108
A.59 edit YumConfig .....	109
A.60 exit .....	110
A.61 getEvents .....	110
A.62 getIncompatibleReasons .....	111
A.63 getJobsUsingRange .....	112
A.64 getLatestNumberOfJobs .....	113
A.65 help .....	113
A.66 importAssembly .....	114
A.67 importTemplate .....	115
A.68 importVirtualCdrom .....	116
A.69 importVirtualDisk .....	117
A.70 importVirtualMachine .....	118
A.71 initiateYumUpgrade .....	119
A.72 kill .....	119
A.73 list .....	120
A.74 migrate Vm .....	121
A.75 refresh .....	122
A.76 refreshStorageLayer .....	123
A.77 remove BondPort .....	124
A.78 remove FileSystem .....	125
A.79 remove PhysicalDisk .....	126
A.80 remove Port .....	126
A.81 remove Server .....	127
A.82 remove ServerPool .....	129
A.83 remove StorageInitiator .....	129
A.84 remove Tag .....	130
A.85 remove Vm .....	131
A.86 remove Vnic .....	132
A.87 removeAccessHost .....	133
A.88 removeNfsRefreshServer .....	134
A.89 restart .....	134
A.90 resume Vm .....	135
A.91 sendVmMessage .....	136
A.92 set .....	136
A.93 setNtp .....	137
A.94 show .....	138
A.95 showallcustomcmds .....	139
A.96 showcustomcmds .....	140
A.97 showNtp .....	141
A.98 showobjtypes .....	141
A.99 showversion .....	142
A.100 start .....	142
A.101 stop .....	143
A.102 suspend Vm .....	144
A.103 syncNtp .....	145
A.104 vnicCreate .....	145

---

## Appendix A CLI Command Reference

### Table of Contents

A.1 abort Job .....	47
A.2 ackEvent .....	48
A.3 addAccessHost .....	49
A.4 addAvailableVnic Vm .....	50
A.5 addNfsRefreshServer .....	51
A.6 add BondPort .....	51
A.7 add FileSystem .....	52
A.8 add PhysicalDisk .....	53
A.9 add Port .....	54
A.10 add Server .....	55
A.11 add ServerPool .....	56
A.12 add StorageInitiator .....	57
A.13 add Tag .....	58
A.14 add Vm .....	59
A.15 add Vnic .....	60
A.16 clone VirtualDisk .....	60
A.17 clone Vm .....	61
A.18 create AccessGroup .....	62
A.19 create BondPort .....	63
A.20 create FileServer .....	65
A.21 create Network .....	66
A.22 create NfsAccessGroup .....	68
A.23 create PhysicalDisk .....	69
A.24 create Repository .....	70
A.25 create SanServer .....	71
A.26 create ServerPool .....	73
A.27 create Tag .....	74
A.28 create VirtualDisk .....	75
A.29 create VlanGroup .....	76
A.30 create VlanSegment .....	77
A.31 create Vm .....	78
A.32 create VmDiskMapping .....	80
A.33 create Vnic .....	82
A.34 createTemplatesFromAssembly .....	83
A.35 delete .....	83
A.36 discoverServer .....	85
A.37 edit AccessGroup .....	86
A.38 edit Assembly .....	87
A.39 edit BondPort .....	88
A.40 edit FileServer .....	89
A.41 edit FileSystem .....	90
A.42 edit Network .....	91
A.43 edit NfsAccessGroup .....	92
A.44 edit PhysicalDisk .....	93
A.45 edit Port .....	94
A.46 edit Repository .....	95
A.47 edit SanServer .....	96
A.48 edit Server .....	97

---

A.49 edit ServerPool .....	99
A.50 edit Tag .....	100
A.51 edit VirtualCdrom .....	101
A.52 edit VirtualDisk .....	102
A.53 edit VlanGroup .....	103
A.54 edit VlanInterface .....	104
A.55 edit Vm .....	104
A.56 edit VmDiskMapping .....	106
A.57 edit Vnic .....	107
A.58 edit VolumeGroup .....	108
A.59 edit YumConfig .....	109
A.60 exit .....	110
A.61 getEvents .....	110
A.62 getIncompatibleReasons .....	111
A.63 getJobsUsingRange .....	112
A.64 getLatestNumberOfJobs .....	113
A.65 help .....	113
A.66 importAssembly .....	114
A.67 importTemplate .....	115
A.68 importVirtualCdrom .....	116
A.69 importVirtualDisk .....	117
A.70 importVirtualMachine .....	118
A.71 initiateYumUpgrade .....	119
A.72 kill .....	119
A.73 list .....	120
A.74 migrate Vm .....	121
A.75 refresh .....	122
A.76 refreshStorageLayer .....	123
A.77 remove BondPort .....	124
A.78 remove FileSystem .....	125
A.79 remove PhysicalDisk .....	126
A.80 remove Port .....	126
A.81 remove Server .....	127
A.82 remove ServerPool .....	129
A.83 remove StorageInitiator .....	129
A.84 remove Tag .....	130
A.85 remove Vm .....	131
A.86 remove Vnic .....	132
A.87 removeAccessHost .....	133
A.88 removeNfsRefreshServer .....	134
A.89 restart .....	134
A.90 resume Vm .....	135
A.91 sendVmMessage .....	136
A.92 set .....	136
A.93 setNtp .....	137
A.94 show .....	138
A.95 showallcustomcmds .....	139
A.96 showcustomcmds .....	140
A.97 showNtp .....	141
A.98 showobjtypes .....	141
A.99 showversion .....	142
A.100 start .....	142
A.101 stop .....	143
A.102 suspend Vm .....	144



A.103 syncNtp .....	145
A.104 vnicCreate .....	145

## List of Examples

A.1 Aborting a job .....	48
A.2 Acknowledging an event .....	49
A.3 Adding a SAN server access host .....	49
A.4 Adding the next available VNIC to a virtual machine .....	50
A.5 Adding a refresh server to file server .....	51
A.6 Adding a bonded port to a network .....	52
A.7 Adding a file system to a file server access group .....	53
A.8 Adding a physical disk to a SAN storage access group .....	54
A.9 Adding an Ethernet port to a network .....	54
A.10 Adding an Oracle VM Server to a server pool .....	56
A.11 Adding an admin server to file server .....	56
A.12 Adding an admin server to a SAN server .....	56
A.13 Adding an Oracle VM Server to a file server access group .....	56
A.14 Adding (presenting) an Oracle VM Server to a storage repository .....	56
A.15 Presenting a storage repository to a server pool .....	57
A.16 Adding a storage initiator .....	58
A.17 Adding a tag to a server pool .....	58
A.18 Adding a virtual machine to an Oracle VM Server .....	59
A.19 Adding a VNIC to a virtual machine .....	60
A.20 Cloning a virtual disk .....	61
A.21 Cloning a virtual machine to a virtual machine .....	62
A.22 Cloning a virtual machine to a template .....	62
A.23 Cloning a template to a virtual machine .....	62
A.24 Cloning a template to a template .....	62
A.25 Creating an access group on a SAN server .....	63
A.26 Creating a bond port .....	64
A.27 Discovering a file server .....	66
A.28 Creating a network .....	67
A.29 Creating an access group on a file server .....	68
A.30 Creating a physical disk on a volume group .....	69
A.31 Creating a storage repository on a file server .....	71
A.32 Creating a storage repository on a physical disk .....	71
A.33 Discovering a SAN server .....	72
A.34 Creating a clustered server pool .....	74
A.35 Creating an unclustered server pool .....	74
A.36 Creating a tag .....	75
A.37 Creating a virtual disk in a storage repository .....	76
A.38 Creating a VLAN group .....	77
A.39 Creating a VLAN segment .....	78
A.40 Creating a virtual machine .....	80
A.41 Mapping a virtual disk to a virtual machine .....	81
A.42 Mapping an ISO file to a virtual machine .....	81
A.43 Mapping an empty CDROM drive .....	81
A.44 Creating a VNIC .....	82
A.45 Creating a virtual machine template from an assembly .....	83
A.46 Deleting an Oracle VM Server .....	84
A.47 Deleting a virtual machine or virtual machine template .....	84
A.48 Deleting a network .....	84
A.49 Deleting a virtual machine disk mapping .....	84

---

A.50 Discovering an Oracle VM Server .....	85
A.51 Editing an access group on a SAN server .....	86
A.52 Editing an assembly .....	87
A.53 Editing a bond port .....	88
A.54 Editing a file server .....	89
A.55 Editing a file system .....	90
A.56 Editing a network .....	91
A.57 Editing an access group on a file server .....	92
A.58 Editing a physical disk .....	93
A.59 Editing a port .....	94
A.60 Editing a storage repository .....	95
A.61 Editing a SAN server .....	97
A.62 Editing an Oracle VM Server .....	98
A.63 Taking ownership of an Oracle VM Server .....	98
A.64 Placing an Oracle VM Server into maintenance mode .....	98
A.65 Editing a server pool .....	99
A.66 Changing the master Oracle VM Server .....	99
A.67 Changing the virtual machine start policy .....	100
A.68 Editing a tag .....	100
A.69 Editing a ISO file/CDROM .....	101
A.70 Editing a virtual disk .....	102
A.71 Resizing a virtual disk .....	102
A.72 Editing a VLAN group .....	103
A.73 Editing a VLAN interface .....	104
A.74 Edit the disk boot order a virtual machine .....	106
A.75 Editing a virtual disk mapping object .....	107
A.76 Editing a VNIC .....	108
A.77 Editing a volume group .....	109
A.78 Editing a Yum repository .....	110
A.79 Exiting the CLI .....	110
A.80 Listing Oracle VM Server events .....	111
A.81 Listing virtual machine placement incompatibility reasons .....	112
A.82 Listing jobs in a date range .....	112
A.83 Listing recent jobs .....	113
A.84 Using the Help Command .....	114
A.85 Importing an assembly to a storage repository .....	114
A.86 Importing a virtual machine template to a storage repository .....	115
A.87 Importing a virtual CDROM/ISO file to a storage repository .....	116
A.88 Importing a virtual disk file to a storage repository .....	117
A.89 Importing a virtual machine to a storage repository .....	118
A.90 Upgrading an Oracle VM Server .....	119
A.91 Killing an Oracle VM Server .....	120
A.92 Killing a virtual machine .....	120
A.93 Listing Oracle VM Servers .....	121
A.94 Listing virtual machines and virtual machine templates .....	121
A.95 Listing networks .....	121
A.96 Listing virtual machine and virtual machine templates disk mapping .....	121
A.97 Migrating a virtual machine to an Oracle VM Server .....	122
A.98 Migrating a virtual machine to another server pool .....	122
A.99 Migrating a virtual machine to the Unassigned Virtual Machines folder .....	122
A.100 Refreshing a file server .....	123
A.101 Refreshing a SAN server .....	123
A.102 Refreshing a physical disk .....	123
A.103 Refresh a file system .....	123

---

A.104 Refresh a storage repository .....	123
A.105 Refreshing Oracle VM Server storage .....	124
A.106 Removing a bonded port from a network .....	124
A.107 Removing a file system from a file server access group .....	125
A.108 Removing physical disk from a SAN storage access group .....	126
A.109 Removing a port from a network .....	127
A.110 Removing an Oracle VM Server from a server pool .....	128
A.111 Removing an admin server from file server .....	128
A.112 Removing an admin server from a SAN server .....	128
A.113 Removing an Oracle VM Server from a file server access group .....	128
A.114 Removing (unpresenting) an Oracle VM Server from a storage repository .....	128
A.115 Unpresenting a storage repository from a server pool .....	129
A.116 Removing a storage initiator .....	130
A.117 Removing a tag from a server pool .....	131
A.118 Removing a virtual machine from an Oracle VM Server .....	132
A.119 Removing a VNIC from a virtual machine .....	132
A.120 Removing a SAN server access host .....	133
A.121 Removing a refresh server from a file server .....	134
A.122 Restarting an Oracle VM Server .....	135
A.123 Restarting a virtual machine .....	135
A.124 Resuming a virtual machine .....	136
A.125 Sending a message to a virtual machine .....	136
A.126 Setting end of line characters .....	137
A.127 Setting output mode to XML .....	137
A.128 Setting the list of NTP servers .....	138
A.129 Show details about an Oracle VM Server .....	139
A.130 Show details about a virtual machine .....	139
A.131 Show details about a network .....	139
A.132 Show Yum repository configuration .....	139
A.133 Show details about a VNIC .....	139
A.134 Showing all custom commands .....	140
A.135 Show Custom Commands for an Oracle VM Server .....	140
A.136 Show Custom Commands for a virtual machine .....	141
A.137 Show Custom Commands for a Repository .....	141
A.138 Showing the list of NTP servers .....	141
A.139 Showing all object types .....	142
A.140 Showing the CLI/Oracle VM Manager version number .....	142
A.141 Starting an Oracle VM Server .....	143
A.142 Starting a virtual machine .....	143
A.143 Stopping an Oracle VM Server .....	144
A.144 Stopping a virtual machine .....	144
A.145 Suspending a virtual machine .....	144
A.146 Pushing the list of NTP servers to Oracle VM Servers .....	145
A.147 Generating multiple VNICs .....	146

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=id | name=name }
```

## Description

This command aborts a running job.

## Options

The following table shows the available options for this command.

Option	Description
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyJob</i> .

## Examples

### Example A.1 Aborting a job

```
OVM> abort job id=1342399962239
```

## See Also

- [Section A.63, “getJobsUsingRange”](#)
- [Section A.64, “getLatestNumberOfJobs”](#)
- [Section A.73, “list”](#)
- [Section A.94, “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.

Option	Description
eventId= <i>value</i>	The ID of the event. To get the ID of an event, use the <a href="#">getEvents</a> command.

## Examples

### Example A.2 Acknowledging an event

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

## See Also

- [Section A.61, “getEvents”](#)

## A.3 addAccessHost

Adds an access host to an iSCSI server.

## Syntax

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

Where *instance* is:

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

## Description

This command adds an access host to an iSCSI SAN server. 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.

Option	Description
<code>accessHost=<i>value</i></code>	The hostname or IP address for the access host.
<code>accessPort=<i>value</i></code>	The port on which to connect to the access host. The default port of <code>3260</code> is used if no value is provided.
<code>accessUsername=<i>value</i></code>	The username to use when using CHAP authentication.
<code>accessPassword=<i>value</i></code>	The password for the <code>accessUsername</code> user.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyISCSIServer</code> .

## Examples

### Example A.3 Adding a SAN server access host

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

## See Also

- [Section A.87, “removeAccessHost”](#)

- [Section A.25, “create SanServer”](#)
- [Section A.47, “edit SanServer”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.4 addAvailableVnic Vm

Adds an available VNIC to a virtual machine.

### Syntax

```
addAvailableVnic Vm instance
```

Where *instance* is:

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

### Description

This command adds an available VNIC to a virtual machine. The VNIC to add does not need to be explicitly defined, and the next available unused VNIC is used.

### Options

The following table shows the available options for this command.

Option	Description
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyVM</i> .

### Examples

#### Example A.4 Adding the next available VNIC to a virtual machine

```
OVM> addAvailableVnic Vm name=MyVM
```

### See Also

- [Section A.33, “create Vnic”](#)
- [Example A.147, “Generating multiple VNICs”](#)
- [Section A.15, “add Vnic”](#)
- [Section A.57, “edit Vnic”](#)
- [Section A.86, “remove Vnic”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.5 addNfsRefreshServer

Adds a refresh server to a file server.

### Syntax

```
addNfsRefreshServer FileServer instance nfsRefreshServer=value
```

Where *instance* is:

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

### 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.

Option	Description
<code>FileServer <i>instance</i></code>	The name or ID of the file server.
<code>nfsRefreshServer=<i>value</i></code>	The name or ID of the Oracle VM Server to be used as a refresh server.
<code>{ <i>id=id</i>   <i>name=name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code> .

### Examples

#### Example A.5 Adding a refresh server to file server

```
OVM> addNfsRefreshServer FileServer name=MyNFSServer nfsRefreshServer=MyServer1
```

### See Also

- [Section A.20, “create FileServer”](#)
- [Section A.88, “removeNfsRefreshServer”](#)
- [Section A.75, “refresh”](#)
- [Section A.94, “show”](#)

## A.6 add BondPort

Adds an bonded port to a network object.

### Syntax

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

Where *instance* is:

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

## Description

This command adds a bonded port to a network object.

## Options

The following table shows the available options for this command.

Option	Description
{ <a href="#">Network</a>   <a href="#">VlanGroup</a> }	The network object to which to add the bonded port.
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <code>name=MyNetwork</code> .

## Examples

### Example A.6 Adding a bonded port to a network

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

## See Also

- [Section A.19, “create BondPort”](#)
- [Section A.39, “edit BondPort”](#)
- [Section A.77, “remove BondPort”](#)
- [Section A.80, “remove Port”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.7 add FileSystem

Adds a file system to a file server access group.

### Syntax

```
add FileSystem instance to NfsAccessGroup instance
```

Where *instance* is:

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

### Description

This command adds a file system to a file server access group.



## Options

The following table shows the available options for this command.

Option	Description
{ <i>id=id</i>   <i>name=name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyNfsAccessGroup</i> .

## Examples

### Example A.7 Adding a file system to a file server access group

```
OVM> add FileSystem name=nfs:/mnt/vol1/repo01 to NfsAccessGroup name=MyNfsAccessGroup
```

## See Also

- [Section A.78, “remove FileSystem”](#)
- [Section A.22, “create NfsAccessGroup”](#)
- [Section A.43, “edit NfsAccessGroup”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.35, “delete”](#)

## A.8 add PhysicalDisk

Adds a physical disk to a SAN storage access group.

## Syntax

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

Where *instance* is:

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

## 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.

Option	Description
{ <i>id=id</i>   <i>name=name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyAccessGroup</i> .

## Examples

### Example A.8 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.23, “create PhysicalDisk”](#)
- [Section A.44, “edit PhysicalDisk”](#)
- [Section A.79, “remove PhysicalDisk”](#)
- [Section A.35, “delete”](#)
- [Section A.75, “refresh”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.9 add Port

Adds an Ethernet port to a network object.

### Syntax

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

Where *instance* is:

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

### Description

This command adds an Ethernet port to a network object.

### Options

The following table shows the available options for this command.

Option	Description
{ BondPort   Network   VlanGroup }	The network object to which to add the Ethernet port.
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyNetwork</i> .

## Examples

### Example A.9 Adding an Ethernet port to a network

```
OVM> add Port id=0004fb0000200000ed75de88c4dcb296 to Network name=MyVMNetwork
```

## See Also

- [Section A.45, “edit Port”](#)
- [Section A.80, “remove Port”](#)
- [Section A.21, “create Network”](#)
- [Section A.42, “edit Network”](#)
- [Section A.19, “create BondPort”](#)
- [Section A.29, “create VlanGroup”](#)
- [Section A.53, “edit VlanGroup”](#)
- [Section A.30, “create VlanSegment”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.10 add Server

Adds an Oracle VM Server to an object.

### Syntax

```
add Server instance to { ServerPool | FileServer | SanServer | Repository | NfsAccessGroup } instance
```

Where *instance* is:

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

### Description

This command adds an Oracle VM Server to either a server pool, file server, SAN server, storage repository or file server access group. When you add an Oracle VM Server to a file server, SAN server or 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.

### Options

The following table shows the available options for this command.

Option	Description
{ <a href="#">ServerPool</a>   <a href="#">FileServer</a>   <a href="#">SanServer</a>   <a href="#">Repository</a>   <a href="#">NfsAccessGroup</a> }	The object on which to assign the Oracle VM Server as an admin server.
{ <i>id=id</i>   <i>name=name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyServer</i> .

---

## Examples

### Example A.10 Adding an Oracle VM Server to a server pool

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

### Example A.11 Adding an admin server to file server

```
OVM> add Server name=MyServer to FileServer name=MyNFSServer
```

### Example A.12 Adding an admin server to a SAN server

```
OVM> add Server name=MyServer to SanServer name=MyISCSIServer
```

### Example A.13 Adding an Oracle VM Server to a file server access group

```
OVM> add Server name=MyServer to NfsAccessGroup name=MyNFSAccessGroup
```

### Example A.14 Adding (presenting) an Oracle VM Server to a storage repository

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

## See Also

- [Section A.11, “add ServerPool”](#)
- [Section A.36, “discoverServer”](#)
- [Section A.48, “edit Server”](#)
- [Section A.75, “refresh”](#)
- [Section A.100, “start”](#)
- [Section A.101, “stop”](#)
- [Section A.89, “restart”](#)
- [Section A.72, “kill”](#)
- [Section A.71, “initiateYumUpgrade”](#)
- [Section A.81, “remove Server”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.11 add ServerPool

Adds a server pool to a storage repository.

### Syntax

```
add ServerPool instance to Repository instance
```

Where *instance* is:

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

## 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.

## Options

The following table shows the available options for this command.

Option	Description
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyServerPool</i> .

## Examples

### Example A.15 Presenting a storage repository to a server pool

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

## See Also

- [Section A.82, “remove ServerPool”](#)
- [Section A.10, “add Server”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.12 add StorageInitiator

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

## Syntax

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

Where *instance* is:

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

## 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.

Option	Description
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyAccessGroup</code> .

## Examples

### Example A.16 Adding a storage initiator

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

## See Also

- [Section A.83, “remove StorageInitiator”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.13 add Tag

Adds a tag to an object.

## Syntax

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

Where *instance* is:

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

## 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.

Option	Description
<code>{ ServerPool   Server   Vm }</code>	The object on which to add the tag.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyTag</code> .

## Examples

### Example A.17 Adding a tag to a server pool

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

## See Also

- [Section A.27, “create Tag”](#)

- [Section A.50, “edit Tag”](#)
- [Section A.84, “remove Tag”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.14 add Vm

Adds a virtual machine to an Oracle VM Server.

### Syntax

```
add Vm instance to Server instance
```

Where *instance* is:

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

### Description

This command adds a virtual machine to an Oracle VM Server. 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.

Option	Description
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyServer</i> .

### Examples

#### Example A.18 Adding a virtual machine to an Oracle VM Server

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

### See Also

- [Section A.31, “create Vm”](#)
- [Section A.70, “importVirtualMachine”](#)
- [Section A.85, “remove Vm”](#)
- [Section A.74, “migrate Vm”](#)
- [Section A.17, “clone Vm”](#)
- [Section A.100, “start”](#)

- [Section A.94, “show”](#)
- [Section A.73, “list”](#)

## A.15 add Vnic

Adds a VNIC to a virtual machine.

### Syntax

```
add Vnic instance to Vm instance
```

Where *instance* is:

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

### Description

This command adds a VNIC to a virtual machine. To add the next available VNIC to a virtual machine without explicitly specifying one, use the [addAvailableVnic](#) command.

### Options

The following table shows the available options for this command.

Option	Description
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <a href="#">name=MyVM</a> .

### Examples

#### Example A.19 Adding a VNIC to a virtual machine

```
OVM> add Vnic name=00:21:f6:00:00:00 to Vm name=MyVM
```

### See Also

- [Section A.33, “create Vnic”](#)
- [Example A.147, “Generating multiple VNICs”](#)
- [Section A.4, “addAvailableVnic Vm”](#)
- [Section A.57, “edit Vnic”](#)
- [Section A.86, “remove Vnic”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.16 clone VirtualDisk



Clones a virtual disk.

## Syntax

```
clone VirtualDisk instance target=value cloneType= { Sparse | nonSparse }
```

Where *instance* is:

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

## Description

This command clones a virtual disk.

## Options

The following table shows the available options for this command.

Option	Description
<code>target=<i>value</i></code>	The repository on which to locate the cloned virtual disk.
<code>cloneType= { Sparse   nonSparse }</code>	Whether to clone a sparse or non-sparse virtual disk.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyVMDisk</code> .

## Examples

### Example A.20 Cloning a virtual disk

```
OVM> clone virtualDisk name=MyVMDisk target=MyRepository cloneType=Sparse
```

## See Also

- [Section A.69, “importVirtualDisk”](#)
- [Section A.28, “create VirtualDisk”](#)
- [Section A.52, “edit VirtualDisk”](#)
- [Section A.35, “delete”](#)
- [Section A.75, “refresh”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.17 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
```

Where *instance* is:

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

## 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.

Option	Description
<code>destType= { Vm   VmTemplate }</code>	The object to create from the virtual machine, either a virtual machine or a template.
<code>destName=<i>value</i></code>	The name of the cloned virtual machine or template.
<code>serverPool=<i>value</i></code>	The server pool on which to deploy the cloned virtual machine or template.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyVMTemplate</code> .

## Examples

### Example A.21 Cloning a virtual machine to a virtual machine

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

### Example A.22 Cloning a virtual machine to a template

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

### Example A.23 Cloning a template to a virtual machine

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

### Example A.24 Cloning a template to a template

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

## See Also

- [Section A.31, “create Vm”](#)
- [Section A.14, “add Vm”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.18 create AccessGroup

Creates a SAN storage access group.

## Syntax

```
create AccessGroup name=value [description=value] on SanServer instance
```

Where *instance* is:

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

## Description

This command creates a SAN storage access group. Generic storage plug-ins are not supported with this command.

## Options

The following table shows the available options for this command.

Option	Description
<code>name=<i>value</i></code>	A name for the access group.
<code>description=<i>value</i></code>	Optional description for the access group.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MySANServer</code> .



### 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.25 Creating an access group on a SAN server

```
OVM> create AccessGroup name=MyGroup on SanServer name=MyISCSERVER
```

## See Also

- [Section A.37, “edit AccessGroup”](#)
- [Section A.8, “add PhysicalDisk”](#)
- [Section A.79, “remove PhysicalDisk”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.19 create BondPort

Creates a bond port.

## Syntax

```
create BondPort addressType= { none | staticIp | dynamic } ethernetPort1=value
ethernetPort2=value mode= { activeBackup | linkAggregation | adaptiveLoadBalancing }
mtu=value [ipAddress=value] [netMask=value] name=value [description=value] on Server
instance
```

Where *instance* is:

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

## Description

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

## Options

The following table shows the available options for this command.

Option	Description
addressType= { none   staticIp   dynamic }	The IP address type.
ethernetPort1=value	The name or ID of the Ethernet port.
ethernetPort2=value	The name or ID of the Ethernet port.
mode= { activeBackup   linkAggregation   adaptiveLoadBalancing }	The network bonding mode.
mtu=value	The MTU value.
ipAddress=value	The IP address.
netMask=value	The netmask.
name=value	A name to identify the bond.
description=value	Optional description for the bond.
{ id=id   name=name }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyServer</i> .



### 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.26 Creating a bond port

```
OVM> create BondPort addressType=none ethernetPort1=0004fb000020000095d52158dc7056b9 \
ethernetPort2=0004fb000020000080ea2d6e38bbc838 mode=linkAggregation mtu=1500
name='Bond Port 2' on Server name=MyServer
```

## See Also

- [Section A.6, “add BondPort”](#)
- [Section A.39, “edit BondPort”](#)
- [Section A.77, “remove BondPort”](#)
- [Section A.80, “remove Port”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.20 create FileServer

Discovers a file server.

### Syntax

```
create FileServer plugin=name accessHost=value [ adminHost=value
adminUserName=value adminPassword=value ][ uniformedExports= {yes | no } ][
name=value ][ description=value ]
```

### Description

This command discovers a file server and adds it to Oracle VM Manager. If you are adding a non-generic file server, for example a Sun ZFS Storage Appliance, also enter the additional plug-in options to enable Oracle VM Manager to access the file server's configuration management functions using the [adminHost](#) option.

After discovering a file server, you should:

- Add one or more admin servers to the file server using the [add Server](#) command.
- Add one or more refresh servers to the file server using the [addNfsRefreshServer](#) command.
- Optionally, if you are using non-uniformed file system exports, you can create a file server access group using the [create NfsAccessGroup](#) 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.

Option	Description
<code>plugin=<i>name</i></code>	The storage plug-in to use for the file server to be discovered. The plugin name must match an existing plugin name exactly. To obtain a list of existing plugin names, do: <pre>OVM&gt; list FileServerPlugin</pre>

Option	Description
	If a vendor specific plug-in is configured it is made available as an option here. To see the list of options, use the <code>?</code> option, for example: <pre>OVM&gt; create FileServer plugin=?</pre>
<code>accessHost=value</code>	The host name or IP address for the file server to be discovered.
<code>adminHost=value</code>	The host name or IP address where administrative access to the file server is allowed.
<code>adminUserName=value</code>	A user name with administrative access to the file server, used with <code>adminHost</code> .
<code>adminPassword=value</code>	The administrator password for the <code>adminUserName</code> user.
<code>uniformedExports= { yes   no }</code>	Whether the file server has uniformed file system exports. The default is <code>yes</code> .
<code>name=value</code>	A name to identify the file server.
<code>description=value</code>	Optional description for the file server.



#### 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.27 Discovering a file server

```
OVM> create FileServer plugin="Oracle Generic Network File System" \
  accessHost=10.172.76.125 name=MyNFSServer
```

## See Also

- [Section A.40, “edit FileServer”](#)
- [Section A.10, “add Server”](#)
- [Section A.81, “remove Server”](#)
- [Section A.5, “addNfsRefreshServer”](#)
- [Section A.75, “refresh”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.21 create Network

Creates an Ethernet-based network.

## Syntax

```
create Network [roles= { MANAGEMENT | LIVE_MIGRATE | CLUSTER_HEARTBEAT |
VIRTUAL_MACHINE | STORAGE }] [vlanSegment=value] [server=value] name=value
[description=value]
```

## Description

This command creates an Ethernet-based network.

## Options

The following table shows the available options for this command.

Option	Description
<code>roles= { MANAGEMENT   LIVE_MIGRATE   CLUSTER_HEARTBEAT   VIRTUAL_MACHINE   STORAGE }</code>	The network roles. Enter options separated by commas (,), for example: <code>roles='VIRTUAL_MACHINE,STORAGE'</code>
<code>vlanSegment=value</code>	The name or ID of a VLAN segment.
<code>server=value</code>	The name or ID of an Oracle VM Server.
<code>name=value</code>	A name to identify the network.
<code>description=value</code>	Optional description for the network.



### 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.28 Creating a network

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

## See Also

- [Section A.9, “add Port”](#)
- [Section A.45, “edit Port”](#)
- [Section A.80, “remove Port”](#)
- [Section A.42, “edit Network”](#)
- [Section A.19, “create BondPort”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.22 create NfsAccessGroup

Creates an access group on a file server.

### Syntax

```
create NfsAccessGroup name=value [description=value] on FileServer instance
```

Where *instance* is:

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

### Description

This command creates an access group on a file server.

### Options

The following table shows the available options for this command.

Option	Description
<code>name=value</code>	A name for the access group.
<code>description=value</code>	Optional description for the access group.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyNFSServer</code> .



#### 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.29 Creating an access group on a file server

```
OVM> create NfsAccessGroup name=MyNFSAccessGroup on FileServer name=MyNFSServer
```

### See Also

- [Section A.43, “edit NfsAccessGroup”](#)
- [Section A.7, “add FileSystem”](#)
- [Section A.78, “remove FileSystem”](#)
- [Section A.10, “add Server”](#)
- [Section A.81, “remove Server”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)



- [Section A.35, “delete”](#)

## A.23 create PhysicalDisk

Creates a physical disk on a volume group.

### Syntax

```
create PhysicalDisk size=value [extraInfo=value] shareable= { yes | no }
[thinProvision= { yes | no }] name=value [description=value] on VolumeGroup instance
```

Where *instance* is:

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

### 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.

Option	Description
<code>size=<i>value</i></code>	The size of the physical disk in GiB.
<code>extraInfo=<i>value</i></code>	
<code>shareable= { yes   no }</code>	Whether the physical disk is shareable. Shareable disks have read/write privileges in multiple virtual machines and should be used with caution.
<code>thinProvision= { yes   no }</code>	Whether to create a thin or non-thin physical disk.
<code>name=<i>value</i></code>	A name to identify the physical disk.
<code>description=<i>value</i></code>	Optional description for the physical disk.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyVolumeGroup</code> .



#### 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.30 Creating a physical disk on a volume group

```
OVM> create PhysicalDisk size=10 name=MyPhysicalDisk shareable=no on VolumeGroup \
id='Storage_Volume_Group @ 0004fb0000090000325a36dad3b3b7d8'
```

### See Also

- [Section A.8, “add PhysicalDisk”](#)

- [Section A.44, “edit PhysicalDisk”](#)
- [Section A.79, “remove PhysicalDisk”](#)
- [Section A.35, “delete”](#)
- [Section A.75, “refresh”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.24 create Repository

Creates a storage repository.

### Syntax

```
create Repository { fileSystem=value | serverPool=value physicalDisk=value } [
sharePath=value ] name=value [ description=value ]
```

### Description

This command creates a storage repository. You must supply either a `fileSystem`, or a `serverPool` and `physicalDisk` value(s).

After you create a repository, you should refresh it, and also refresh any disks individually to make them available in Oracle VM Manager.

### Options

The following table shows the available options for this command.

Option	Description
<code>fileSystem=value</code>	The name or identifier of the file system to use for the storage repository.
<code>serverPool=value</code>	The name or identifier of the server pool to which the storage repository should be made available. This must be used with the <code>physicalDisk</code> option.
<code>physicalDisk=value</code>	The name or identifier of the physical disk to use for the storage repository. This must be used with the <code>serverPool</code> option.
<code>sharePath=value</code>	A path to a subdirectory on the selected file system.
<code>name=value</code>	A name to identify the storage repository.
<code>description=value</code>	Optional description for the storage repository.



#### 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.31 Creating a storage repository on a file server

```
OVM> create Repository name=MyRepository fileSystem=nfs:/mnt/vol1/repo01
```

### Example A.32 Creating a storage repository on a physical disk

```
OVM> create Repository name=MySANRepository serverPool=MyServerPool \
physicalDisk=0004fb0000180000ba63ec88aa330640
```

## See Also

- [Section A.10, “add Server”](#)
- [Section A.75, “refresh”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.25 create SanServer

Discovers a SAN server.

## Syntax

```
create SanServer plugin=name storageType= { FibreChannelStorageArray |
iSCSIStorageArray } [accessHost=value] [accessPort=value] [ accessUsername=value
accessPassword=value ] [useChap= { yes | no }] [ adminHost=value adminUserName=value
adminPassword=value ] name=value [description=value]
```

## Description

This command discovers a SAN server and adds it to Oracle VM Manager. If you are adding a non-generic SAN server also enter the additional plug-in options to enable Oracle VM Manager to access the file server'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 SAN server, you should add storage initiators to it, add admin servers to it, then refresh it.

## Options

The following table shows the available options for this command.

Option	Description
<code>plugin=<i>name</i></code>	The storage plug-in to use for the SAN server. If a vendor specific plug-in is configured it is made available as an option here. To see the list of options, use the <code>?</code> option, for example:  <pre>OVM&gt; create SanServer plugin=?</pre>

Option	Description
<code>storageType= { FibreChannelStorageArray   iSCSIStorageArray }</code>	The storage type for the SAN server.
<code>accessHost=value</code>	The hostname or IP address for the SAN server. This is not applicable to fibre channel storage arrays.
<code>accessPort=value</code>	The port on which access to the SAN server 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.
<code>accessUsername=value</code>	A username with administrative access to the SAN server, used with <code>accessHost</code> . This option should only be used where CHAP is enabled on the SAN server.
<code>accessPassword=value</code>	The password for the <code>accessUsername</code> user. This option should only be used where CHAP is enabled on the SAN server.
<code>useChap= { yes   no }</code>	Whether to use CHAP authentication.
<code>adminHost=value</code>	The host name or IP address where administrative access to the SAN server is allowed.
<code>adminUserName=value</code>	A user name with administrative access to the SAN server, used with <code>adminHost</code> .
<code>adminPassword=value</code>	The administrator password for the <code>adminUserName</code> user.
<code>name=value</code>	A name to identify the SAN server.
<code>description=value</code>	Optional description for the SAN server.



### 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.33 Discovering a SAN server

```
OVM> create SanServer plugin="Oracle Generic SCSI Plugin(1.2.1)" name=MyISCSIServer \  
storageType=iSCSIStorageArray accessHost=10.172.76.130 accessport=3260
```

## See Also

- [Section A.3, “addAccessHost”](#)
- [Section A.47, “edit SanServer”](#)
- [Section A.10, “add Server”](#)
- [Section A.81, “remove Server”](#)
- [Section A.75, “refresh”](#)
- [Section A.18, “create AccessGroup”](#)

- [Section A.12, “add StorageInitiator”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.26 create ServerPool

Creates a server pool.

### Syntax


```
create ServerPool virtualIP=value clusterEnable= { yes | no } filesystem=value
physicalDisk=value keymapName=value migrateUsingSsl= { yes | no } [startPolicy= {
BEST_SERVER | CURRENT_SERVER }] name=value [description=value]
```

### Description

This command creates a server pool.

### Options

The following table shows the available options for this command.

Option	Description
<code>virtualIP=value</code>	The virtual IP address for the server pool.
<code>clusterEnable= { yes   no }</code>	Whether to enable a clustered server pool. If this parameter is not included, the default is <code>no</code> , so the server pool is not clustered. If this parameter is set to <code>yes</code> , you must also include either the <code>filesystem</code> or <code>physicalDisk</code> option to provide a location for the server pool file system.
<code>filesystem=value</code>	The file system to use for the server pool file system.
<code>physicalDisk=value</code>	The physical disk to use for the server pool file system.  <div style="display: flex; align-items: center;">  <div> <p><b>Note</b></p> <p>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 servers in the server pool.</p> </div> </div>
<code>keymapName=value</code>	The key mapping to be used when connecting to a virtual machine's console.
<code>migrateUsingSsl= { yes   no }</code>	Whether to enable secure migration of virtual machines using SSL.
<code>startPolicy= { BEST_SERVER   CURRENT_SERVER }</code>	The policy by which virtual machines are located when created in the server pool. If none is provided, the <code>CURRENT_SERVER</code> option is used by default.

Option	Description
<code>name=value</code>	A name to identify the server pool.
<code>description=value</code>	Optional description for the server pool.

**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.34 Creating a clustered server pool

```
OVM> create ServerPool virtualIP=10.172.77.195 clusterEnable=yes \
  filesystem=nfs:/mnt/voll/poolfs01 name=MyServerPool description='Clustered server pool'
```

### Example A.35 Creating an unclustered server pool

```
OVM> create ServerPool virtualIP=10.172.77.195 clusterEnable=no name=MyServerPool \
  description='Unclustered server pool'
```

## See Also

- [Section A.49, “edit ServerPool”](#)
- [Section A.10, “add Server”](#)
- [Section A.81, “remove Server”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.27 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.

Option	Description
<code>name=value</code>	A name to identify the tag.
<code>description=value</code>	Optional description for the tag.

**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.36 Creating a tag

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

## See Also

- [Section A.50, “edit Tag”](#)
- [Section A.13, “add Tag”](#)
- [Section A.84, “remove Tag”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.28 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=id | name=name }
```

### Description

This command creates a virtual disk in a storage repository.

### Options

The following table shows the available options for this command.

Option	Description
<code>size=value</code>	The size of the virtual disk in GiB.

Option	Description
<code>shareable= { yes   no }</code>	Whether the virtual disk is shareable. Shareable disks have read/write privileges in multiple virtual machines and should be used with caution.
<code>sparse= { yes   no }</code>	Whether to create a sparse or non-sparse virtual disk.
<code>name=value</code>	A name to identify the virtual disk.
<code>description=value</code>	Optional description for the virtual disk.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyRepository</code> .

**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.37 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.69, “importVirtualDisk”](#)
- [Section A.52, “edit VirtualDisk”](#)
- [Section A.16, “clone VirtualDisk”](#)
- [Section A.35, “delete”](#)
- [Section A.75, “refresh”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.29 create VlanGroup

Creates a VLAN group.

### Syntax

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

### Description

Creates a VLAN group. To add a VLAN segment to the group, use the [create VlanSegment](#) command.



## Options

The following table shows the available options for this command.

Option	Description
<code>name=value</code>	A name for the VLAN group.
<code>description=value</code>	Optional description for the VLAN group.



### 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.38 Creating a VLAN group

```
OVM> create VlanGroup name=MyVLANGroup description="My VLAN Group"
```

## See Also

- [Section A.53, “edit VlanGroup”](#)
- [Section A.30, “create VlanSegment”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.30 create VlanSegment

Creates a VLAN segment.

### Syntax

```
create VlanSegment index=value name=value [description=value] on VlanGroup
instance
```

Where *instance* is:

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

### Description

This command creates a VLAN segment and associates it to a VLAN group.

### Options

The following table shows the available options for this command.

Option	Description
<code>index=value</code>	The index value of the VLAN segment. This is the equivalent to the <b>VLAN ID Range</b> field in the <b>Create</b> or <b>Edit VLAN Group</b> dialog box in the Oracle VM Manager user interface. Enter a value between 2 and 4094.
<code>name=value</code>	A name to identify the VLAN segment.
<code>description=value</code>	Optional description for the VLAN segment.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyVLANGroup</code> .

**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.39 Creating a VLAN segment

```
OVM> create VlanSegment index=3 name=MyVlanSegment on VlanGroup name=MyVlanGroup
```

## See Also

- [Section A.29, “create VlanGroup”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.31 create Vm

Creates a virtual machine.

## Syntax

```
create Vm memory=value memoryLimit=value cpuCountLimit=value cpuCount=value
cpuPriority=value cpuUtilizationCap=value highAvailability= { yes | no }
osType= { WIN_2003 | WIN_2008 | WIN_7 | WIN_VISTA | OTHER_WIN | OL_4 | OL_5 | OL_6 | RHL_4 |
RHL_5 | RHL_6 | LINUX_RECOVERY | OTHER_LINUX | SOLARIS_10 | SOLARIS_11 | OTHER_SOLARIS |
NONE }
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 } networkBootPath=value repository=value name=value
[description=value]{ on Server instance | ServerPool instance }
```

Where *instance* is:

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

## Description

This command creates a virtual machine.

## Options

The following table shows the available options for this command.

Option	Description
<code>memory=value</code>	The memory size the virtual machine is allocated in MB.
<code>memoryLimit=value</code>	The maximum memory size the virtual machine can be allocated in MB.
<code>cpuCountLimit=value</code>	The maximum number of processors the virtual machine can be allocated.
<code>cpuCount=value</code>	The number of processors the virtual machine is allocated.
<code>cpuPriority=value</code>	The CPU priority of the virtual machine. A value between 1 and 100; the higher the number, the more priority the CPU is given.
<code>cpuUtilizationCap=value</code>	The maximum percentage to which the virtual CPUs can receive scheduled time. A value between 1 and 100; the higher the number, the more scheduled time the CPU is given.
<code>highAvailability= { yes   no }</code>	Whether to enable High Availability.
<code>osType= { WIN_2003   WIN_2008   WIN_7   WIN_VISTA   OTHER_WIN   OL_4   OL_5   OL_6   RHL_4   RHL_5   RHL_6   LINUX_RECOVERY   OTHER_LINUX   SOLARIS_10   SOLARIS_11   OTHER_SOLARIS   NONE }</code>	The operating system of the virtual machine.
<code>mouseType= { OS_DEFAULT   PS2_MOUSE   USB_MOUSE   USB_TABLET }</code>	The mouse type of the virtual machine.
<code>domainType= { XEN_HVM   XEN_HVM_PV_DRIVERS   XEN_PVM   LDOMS_PVM   UNKNOWN }</code>	The domain type of the virtual machine.
<code>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 }</code>	The keyboard mapping to use for the virtual machine.
<code>bootOrder= { PXE   DISK   CDROM }</code>	<p>The boot media order for the virtual machine. Enter options separated by commas (,), for example:</p> <pre>bootOrder= ' CDROM, DISK '</pre> <p>If you use the <code>PXE</code> boot option to boot from network-based installation media, also use the <code>networkBootPath</code> parameter.</p>
<code>networkBootPath=value</code>	The location at which the installation media (mounted ISO file) is located when creating a PVM guest.

Option	Description
<code>repository=value</code>	The repository in which to create the virtual machine configuration file.
<code>name=value</code>	A name to identify the virtual machine.
<code>description=value</code>	Optional description for the virtual machine.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code> .

**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.40 Creating a virtual machine

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

## See Also

- [Section A.17, “clone Vm”](#)
- [Section A.55, “edit Vm”](#)
- [Section A.14, “add Vm”](#)
- [Section A.100, “start”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.32 create VmDiskMapping

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

### Syntax

```
create VmDiskMapping slot=value [storageDevice=value] name=value
[description=value] on Vm instance
```

Where *instance* is:


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

### Description

This command maps a virtual disk or CDROM to a virtual machine disk slot. To create an empty CDROM drive, do not provide the `storageDevice` 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.

Option	Description
<code>slot=value</code>	The slot number for the disk in the virtual machine, starting at 0.
<code>storageDevice=value</code>	The name or ID of the disk; either a virtual disk, a virtual CDROM or a physical disk. If this option is not provided, an empty CDROM is created.
<code>name=value</code>	A name to identify the disk mapping.  <div style="display: flex; align-items: center;">  <div> <p><b>Tip</b></p> <p>To find this name after a virtual disk is mapped to a virtual machine, use the <code>list VmDiskMapping</code> command. You will need the name or ID of this to delete a disk mapping from a virtual machine with the <code>delete VmDiskMapping</code> command.</p> </div> </div>
<code>description=value</code>	Optional description for the disk mapping object.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyVM</code> .



### 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.41 Mapping a virtual disk to a virtual machine

```
OVM> create VmDiskMapping slot=0 storageDevice=MyVMDisk name=BootDisk on Vm name=MyVM
```

### Example A.42 Mapping an ISO file to a virtual machine

```
OVM> create VmDiskMapping slot=1 storageDevice=OracleLinux-R6-U2-Server-x86_64-dvd.iso \
name=CDROM on Vm name=MyVM
```

### Example A.43 Mapping an empty CDROM drive

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

## See Also

- [Section A.56, “edit VmDiskMapping”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)

- [Section A.94, “show”](#)

## A.33 create Vnic

Creates a VNIC on a network.

### Syntax

```
create Vnic network=value name=value [description=value]
```

### Description

This command creates a VNIC. To create multiple VNICs, use the `vnicCreate` command. Assigning VNICs to a specific network is optional and can be achieved by specifying the network ID or name as an additional parameter.

### Options

The following table shows the available options for this command.

Option	Description
<code>network=<i>value</i></code>	Optional parameter to assign the VNIC to a network. The name or ID of the network on which to create the VNIC must be specified as the value..
<code>name=<i>value</i></code>	A MAC address for the VNIC.
<code>description=<i>value</i></code>	Optional description for the VNIC.



#### 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 VNIC

```
OVM> create Vnic name=00:21:f6:00:00:18 network=MyVMNetwork
```

### See Also

- [Section A.104, “vnicCreate”](#)
- [Section A.4, “addAvailableVnic Vm”](#)
- [Section A.15, “add Vnic”](#)
- [Section A.57, “edit Vnic”](#)
- [Section A.86, “remove Vnic”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)

- [Section A.94, “show”](#)

## A.34 createTemplatesFromAssembly

Creates virtual machine templates from an assembly file.

### Syntax

```
createTemplatesFromAssembly assembly= { id | name }
```

### Description

This command unpacks all the virtual machine files from an assembly file and creates a virtual machine template for each source virtual machine in the assembly file. The templates are created in the same repository as the original assembly file. After unpacking the virtual machines from the assembly and creating templates, use the [clone Vm](#) command to clone and deploy virtual machines from each template.

### Options

The following table shows the available options for this command.

Option	Description
<code>assembly= { <i>id</i>   <i>name</i> }</code>	The ID or name of the virtual machine assembly file.



#### 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 a virtual machine template from an assembly

```
OVM> createTemplatesFromAssembly assembly=myassembly.ova
```

### See Also

- [Section A.66, “importAssembly”](#)
- [Section A.17, “clone Vm”](#)
- [Section A.38, “edit Assembly”](#)
- [Section A.35, “delete”](#)
- [Section A.75, “refresh”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.35 delete

Deletes an object.

## Syntax

```
delete { AccessGroup | Assembly | BondPort | FileServer | FileSystem | Network |
NfsAccessGroup | PhysicalDisk | Repository | SanServer | Server | ServerPool | Tag
| VirtualCdrom | VirtualDisk | VlanGroup | VlanSegment | Vm | VmDiskMapping | Vnic }
instance
```

Where *instance* is:

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

## 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.

Option	Description
{ AccessGroup   Assembly   BondPort   FileServer   FileSystem   Network   NfsAccessGroup   PhysicalDisk   Repository   SanServer   Server   ServerPool   Tag   VirtualCdrom   VirtualDisk   VlanGroup   VlanSegment   Vm   VmDiskMapping   Vnic }	The object to delete.
{ id=id   name=name }	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code> .

## Examples

### Example A.46 Deleting an Oracle VM Server

```
OVM> delete Server name=MyServer
```

### Example A.47 Deleting a virtual machine or virtual machine template

```
OVM> delete Vm id=0004fb00001400007be890778aedc7b8
```

### Example A.48 Deleting a network

```
OVM> delete Network name=MyVMNetwork
```

### Example A.49 Deleting a virtual machine disk mapping

```
OVM> delete VmDiskMapping id=0004fb00001300009d46acbb77de919e
```

## See Also

- [Section A.73, "list"](#)



- [Section A.94, “show”](#)

## A.36 discoverServer

Discovers an Oracle VM Server.

### Syntax

```
discoverServer ipAddress= { ip_address | hostname } port=port_number
username=username password=password takeOwnership= { yes | no }
```

### Description

This command discovers an Oracle VM Server and adds it to Oracle VM Manager.

### Options

The following table shows the available options for this command.

Option	Description
<code>ipAddress= { ip_address   hostname }</code>	The IP address or hostname of the Oracle VM Server.
<code>port=port_number</code>	The port number on which to access the Oracle VM Agent on the Oracle VM Server.
<code>username=username</code>	The username to use when connecting to the Oracle VM Agent on the Oracle VM Server.
<code>password=password</code>	The password to use when connecting to the Oracle VM Agent on the Oracle VM Server.
<code>takeOwnership= { yes   no }</code>	Whether to take ownership of the Oracle VM Server. <div style="border-left: 2px solid black; padding-left: 10px; margin-top: 10px;"> <p><b>Important</b></p> <p>The <code>takeOwnership</code> option should only be used if the Oracle VM Server is not already owned by an existing Oracle VM Manager installation. If specified for an Oracle VM Server that is already owned, the option is silently ignored.</p> </div>

### Examples

#### Example A.50 Discovering an Oracle VM Server

```
OVM> discoverServer ipAddress=10.172.76.73 port=8899 username=oracle password=password \
takeOwnership=yes
```

### See Also

- [Section A.48, “edit Server”](#)
- [Section A.10, “add Server”](#)
- [Section A.81, “remove Server”](#)

- [Section A.100, “start”](#)
- [Section A.89, “restart”](#)
- [Section A.101, “stop”](#)
- [Section A.72, “kill”](#)
- [Section A.71, “initiateYumUpgrade”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.37 edit AccessGroup

Edits a SAN storage access group.

### Syntax

```
edit AccessGroup instance [name=value] [description=value]
```

Where *instance* is:

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

### Description

This command edits a SAN storage access group. Generic storage 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.

Option	Description
<code>name=<i>value</i></code>	A name for the access group.
<code>description=<i>value</i></code>	Optional description for the access group.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyAccessGroup</code> .

### Examples

#### Example A.51 Editing an access group on a SAN server

```
OVM> edit AccessGroup name=MyGroup name=MyISCSIServer
```

### See Also

- [Section A.18, “create AccessGroup”](#)

- [Section A.8, “add PhysicalDisk”](#)
- [Section A.79, “remove PhysicalDisk”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.38 edit Assembly

Edits an assembly.

### Syntax

```
edit Assembly instance [name=value] [description=value]
```

Where *instance* is:

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

### Description

This command edits 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.

Option	Description
<code>name=value</code>	A name to identify the assembly.
<code>description=value</code>	Optional description for the assembly.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyAssembly.ova</code> .

### Examples

#### Example A.52 Editing an assembly

```
OVN> edit Assembly name=myassembly.ova description='Oracle Linux Release 6'
```

### See Also

- [Section A.66, “importAssembly”](#)
- [Section A.35, “delete”](#)
- [Section A.75, “refresh”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.39 edit BondPort

Edits a bond port.

### Syntax

```
edit BondPort instance [addressType= { none | staticIp | dynamic }] [mode= {
activeBackup | linkAggregation | adaptiveLoadBalancing }] [mtu=value] [ipAddress=value]
[netMask=value] [name=value] [description=value]
```

Where *instance* is:

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

### Description

This command edits 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.

Option	Description
<code>addressType= { none   staticIp   dynamic }</code>	The IP address type.
<code>mode= { activeBackup   linkAggregation   adaptiveLoadBalancing }</code>	The network bonding mode.
<code>mtu=value</code>	The MTU value.
<code>ipAddress=value</code>	The IP address.
<code>netMask=value</code>	The netmask.
<code>name=value</code>	A name to identify the bond.
<code>description=value</code>	Optional description for the bond.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyBondPort</code> .

### Examples

#### Example A.53 Editing a bond port

```
OVM> edit BondPort id=0004fb0000200000884da42c23947622 mode=linkAggregation
```

### See Also

- [Section A.19, “create BondPort”](#)
- [Section A.6, “add BondPort”](#)
- [Section A.77, “remove BondPort”](#)

- [Section A.80, “remove Port”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.40 edit FileServer

Edits a file server.

### Syntax

```
edit FileServer instance [ accessHost=value ] [ adminHost=value adminUserName=value
adminPassword=value ] [ uniformedExports= { yes | no } ] [ name=value ] [
description=value ]
```

Where *instance* is:

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

### Description

This command edits a file server. If you are adding a non-generic file server, for example a Sun ZFS Storage Appliance, also enter the additional plug-in options to enable Oracle VM Manager to access the file server's configuration management functions using the `adminHost` option.

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.

Option	Description
<code>accessHost=value</code>	The host name or IP address for the file server.
<code>adminHost=value</code>	The host name or IP address where administrative access to the file server is allowed.
<code>adminUserName=value</code>	A user name with administrative access to the file server, used with <code>adminHost</code> .
<code>adminPassword=value</code>	The administrator password for the <code>adminUserName</code> user.
<code>uniformedExports= { yes   no }</code>	Whether the file server has uniformed file system exports. The default is <code>yes</code> .
<code>name=value</code>	A name to identify the file server.
<code>description=value</code>	Optional description for the file server.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyNFSServer</code> .

### Examples

#### Example A.54 Editing a file server

```
OVM> edit FileServer id=0004fb00000900000ef55b2f96a564c8 name=MyNFSServer \
```

```
description='My NFS Server'
```

## See Also

- [Section A.20, “create FileServer”](#)
- [Section A.10, “add Server”](#)
- [Section A.81, “remove Server”](#)
- [Section A.75, “refresh”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.41 edit FileSystem

Edits a file system.

### Syntax

```
edit FileSystem instance [name=value] [description=value]
```

Where *instance* is:

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

### Description

This command edits a 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.

Option	Description
<code>name=<i>value</i></code>	A name to identify the file system.
<code>description=<i>value</i></code>	Optional description for the file system.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyFilesystem</code> .

### Examples

#### Example A.55 Editing a file system

```
OVM> edit FileSystem id=0004fb0000050000002618dec56ee0e8 name=MyFilesystem \
description='My File System'
```

## See Also

- [Section A.75, “refresh”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.42 edit Network

Edits an Ethernet-based network.

### Syntax

```
edit Network instance [roles= { MANAGEMENT | LIVE_MIGRATE | CLUSTER_HEARTBEAT |
VIRTUAL_MACHINE | STORAGE }] [vlanSegment=value] [name=value] [description=value]
```

Where *instance* is:

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

### Description

This command edits 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.

Option	Description
<code>roles= { MANAGEMENT   LIVE_MIGRATE   CLUSTER_HEARTBEAT   VIRTUAL_MACHINE   STORAGE }</code>	The network roles. Enter options separated by commas (,), for example: <code>roles='VIRTUAL_MACHINE,STORAGE'</code>
<code>vlanSegment=value</code>	The name or ID of a VLAN segment.
<code>name=value</code>	A name to identify the network.
<code>description=value</code>	Optional description for the network.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyNetwork</code> .

### Examples

#### Example A.56 Editing a network

```
OVM> edit Network name=MyVMNetwork roles='VIRTUAL_MACHINE,LIVE_MIGRATION'
```

### See Also

- [Section A.9, “add Port”](#)

- [Section A.45, “edit Port”](#)
- [Section A.80, “remove Port”](#)
- [Section A.21, “create Network”](#)
- [Section A.19, “create BondPort”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.43 edit NfsAccessGroup

Edits an access group on a file server.

### Syntax

```
edit NfsAccessGroup instance name=value [description=value]
```

Where *instance* is:

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

### Description

This command edits an access group on a file server.

### Options

The following table shows the available options for this command.

Option	Description
<code>name=<i>value</i></code>	A name for the access group.
<code>description=<i>value</i></code>	Optional description for the access group.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyNFSServer</code> .

### Examples

#### Example A.57 Editing an access group on a file server

```
OVM> edit NfsAccessGroup name=MyNFSAccessGroup name="My new NFS Server name" \
description="This is my NFS access group"
```

### See Also

- [Section A.22, “create NfsAccessGroup”](#)
- [Section A.7, “add FileSystem”](#)
- [Section A.78, “remove FileSystem”](#)
- [Section A.10, “add Server”](#)



- [Section A.81, “remove Server”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.35, “delete”](#)

## A.44 edit PhysicalDisk

Edits a physical disk.

### Syntax

```
edit PhysicalDisk instance [name=value] [shareable= { yes | no }] [size=value]
[description=value] [extraInfo=value]
```

Where *instance* is:

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

### Description

This command edits a physical disk. Local storage and generic storage 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.

Option	Description
<code>name=<i>value</i></code>	A name to identify the physical disk.
<code>shareable= { yes   no }</code>	Whether the physical disk is shareable. Shareable disks have read/write privileges in multiple virtual machines and should be used with caution.
<code>size=<i>value</i></code>	The size of the physical disk in GiB.
<code>description=<i>value</i></code>	Optional description for the physical disk.
<code>extraInfo=<i>value</i></code>	
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyDisk</code> .

### Examples

#### Example A.58 Editing a physical disk

```
OVM> edit PhysicalDisk id=0004fb000018000034a2da375d08990e name=MyPhysicalDisk size=10
```

### See Also

- [Section A.23, “create PhysicalDisk”](#)

- [Section A.8, “add PhysicalDisk”](#)
- [Section A.79, “remove PhysicalDisk”](#)
- [Section A.35, “delete”](#)
- [Section A.75, “refresh”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.45 edit Port

Edits an Ethernet port.

### Syntax

```
edit Port instance [addressType= { none | staticIp | dynamic }] [mtu=value]
[ipAddress=value] [netMask=value] [name=value] [description=value]
```

Where *instance* is:

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

### Description

This command edits 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.

Option	Description
addressType= { none   staticIp   dynamic }	The network addressing type.
mtu= <i>value</i>	The MTU value.
ipAddress= <i>value</i>	The IP address to use for the port.
netMask= <i>value</i>	The netmask to use for the port.
name= <i>value</i>	A name to identify the port.
description= <i>value</i>	Optional description for the port.
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name='Port (2)'</i> .

### Examples

#### Example A.59 Editing a port

```
OVM> edit Port id=0004fb0000200000b0f9d86788b94a0e addressType=none \
```

```
mtu=1500 name='My Port'
```

## See Also

- [Section A.9, “add Port”](#)
- [Section A.19, “create BondPort”](#)
- [Section A.6, “add BondPort”](#)
- [Section A.77, “remove BondPort”](#)
- [Section A.80, “remove Port”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.46 edit Repository

Edits a storage repository.

### Syntax

```
edit Repository instance [ownership= { yes | no }] [name=value] [description=value]
```

Where *instance* is:

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

### Description

This command edits 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.

Option	Description
[ownership= { yes   no }]	Whether to take or relinquish ownership of the storage repository.
name= <i>value</i>	A name to identify the storage repository.
description= <i>value</i>	Optional description for the storage repository.
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example name=MyRepository.

## Examples

### Example A.60 Editing a storage repository

```
OVM> edit Repository id=0004fb00000300003ab65ab35e3fea7a name=MyRepository \
```

```
description="My Storage Repository"
```

## See Also

- [Section A.24, “create Repository”](#)
- [Section A.10, “add Server”](#)
- [Section A.81, “remove Server”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.47 edit SanServer

Edits a SAN server.

### Syntax

```
edit SanServer instance [ accessHost=value ][ accessPort=value ][
accessUsername=value accessPassword=value ][ useChap= { yes | no } ][
adminHost=value adminUserName=value adminPassword=value ][ name=value ][
description=value ]
```

Where *instance* is:

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

### Description

This command edits a SAN 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.

Option	Description
<code>accessHost=value</code>	The host name or IP address for the SAN server. This is not applicable to fibre channel storage arrays.
<code>accessPort=value</code>	The port on which access to the SAN server is allowed.
<code>accessUsername=value</code>	A user name with administrative access to the SAN server, used with <code>accessHost</code> .
<code>accessPassword=value</code>	The administrator password for the <code>accessUsername</code> user.
<code>useChap= { yes   no }</code>	Whether to use CHAP authentication.
<code>adminHost=value</code>	The host name or IP address where administrative access to the SAN server is allowed.

Option	Description
<code>adminUserName=value</code>	A user name with administrative access to the SAN server, used with <code>adminHost</code> .
<code>adminPassword=value</code>	The administrator password for the <code>adminUserName</code> user.
<code>name=value</code>	A name to identify the SAN server.
<code>description=value</code>	Optional description for the SAN server.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MySANServer</code> .

## Examples

### Example A.61 Editing a SAN server

```
OVM> edit SanServer name=MyISCSIServer accessHost=10.172.76.130 accessport=3260
```

## See Also

- [Section A.25, “create SanServer”](#)
- [Section A.10, “add Server”](#)
- [Section A.81, “remove Server”](#)
- [Section A.75, “refresh”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.48 edit Server

Edits an Oracle VM Server.

## Syntax

```
edit Server instance [maintenanceMode= { yes | no }] [roles= { VIRTUAL_MACHINE |
UTILITY }] [takeOwnership= { yes | no }] [name=value] [description=value]
```

Where *instance* is:

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

## Description

This command edits an Oracle VM Server. Use this command to place an Oracle VM Server into or out of maintenance mode, and to take or relinquish ownership of 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.

Option	Description
<code>maintenanceMode= { yes   no }</code>	Whether to place the Oracle VM Server into maintenance mode.
<code>roles= { VIRTUAL_MACHINE   UTILITY }</code>	The role(s) of the Oracle VM Server. Enter options separated by commas (,), for example: <code>roles=VIRTUAL_MACHINE,UTILITY</code>
<code>takeOwnership= { yes   no }</code>	Whether to take ownership of the Oracle VM Server.
<code>name=value</code>	A name to identify the Oracle VM Server.
<code>description=value</code>	Optional description for the Oracle VM Server .
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code> .

## Examples

### Example A.62 Editing an Oracle VM Server

```
OVM> edit Server name=MyServer description="My Oracle VM Server"
```

### Example A.63 Taking ownership of an Oracle VM Server

```
OVM> edit Server name=MyServer takeOwnership=yes
```

### Example A.64 Placing an Oracle VM Server into maintenance mode

```
OVM> edit Server name=MyServer maintenanceMode=yes
```

## See Also

- [Section A.36, “discoverServer”](#)
- [Section A.10, “add Server”](#)
- [Section A.81, “remove Server”](#)
- [Section A.100, “start”](#)
- [Section A.89, “restart”](#)
- [Section A.101, “stop”](#)
- [Section A.72, “kill”](#)
- [Section A.71, “initiateYumUpgrade”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.49 edit ServerPool

Edits a server pool.

### Syntax

```
edit ServerPool instance [keymapName=value] [migrateUsingSsl= { yes | no }]
[startPolicy= { BEST_SERVER | CURRENT_SERVER }] [masterServer=value] [name=value]
[description=value]
```

Where *instance* is:

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

### Description

This command edits a server pool. Use this command to change the master Oracle VM Server, change the keyboard mapping and set whether to use secure migration of virtual machines.

Although none of the options are mandatory, you must supply at least one option.

It is not possible to edit the cluster enable flag or the virtual IP address for a server pool if there are servers in the server pool already. Attempts to edit these attributes for a server pool that already contain servers fail and result in an error.

### Options

The following table shows the available options for this command.

Option	Description
<code>keymapName=value</code>	The key mapping to be used when connecting to a virtual machine's console.
<code>migrateUsingSsl= { yes   no }</code>	Whether to enable secure migration of virtual machines using SSL.
<code>startPolicy= { BEST_SERVER   CURRENT_SERVER }</code>	The policy by which virtual machines are located when created in the server pool. The default is CURRENT_SERVER.
<code>masterServer=value</code>	The master Oracle VM Server to use for the server pool.
<code>name=value</code>	A name to identify the server pool.
<code>description=value</code>	Optional description for the server pool.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code> .

### Examples

#### Example A.65 Editing a server pool

```
OVM> edit ServerPool name=MyServerPool name=MyOtherServerPool migrateUsingSsl=yes
```

#### Example A.66 Changing the master Oracle VM Server

```
OVM> edit ServerPool id=0004fb000002000037db5e362c85a3fe masterServer=MyServer
```

**Example A.67 Changing the virtual machine start policy**

```
OVM> edit ServerPool name=MyServerPool startPolicy=BEST_SERVER
```

**See Also**

- [Section A.26, “create ServerPool”](#)
- [Section A.10, “add Server”](#)
- [Section A.81, “remove Server”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

**A.50 edit Tag**

Edits a tag.

**Syntax**

```
edit Tag instance name=value [description=value]
```

Where *instance* is:

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

**Description**

This command edits a tag used to identify and group objects.

**Options**

The following table shows the available options for this command.

Option	Description
<code>name=value</code>	A name to identify the tag.
<code>description=value</code>	Optional description for the tag.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyTag</code> .

**Examples****Example A.68 Editing a tag**

```
OVM> edit Tag name=MyTag name=MyNewTagName description="My new tag name."
```

**See Also**

- [Section A.27, “create Tag”](#)



- [Section A.13, “add Tag”](#)
- [Section A.84, “remove Tag”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.51 edit VirtualCdrom

Edits an ISO file/CDROM.

### Syntax

```
edit VirtualCdrom instance [name=value] [description=value]
```

Where *instance* is:

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

### Description

This command edits 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.

Option	Description
<code>name=<i>value</i></code>	A name to identify the ISO file/CDROM.
<code>description=<i>value</i></code>	Optional description for the ISO file/CDROM.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyISOFile</code> .

### Examples

#### Example A.69 Editing a ISO file/CDROM

```
OVM> edit VirtualCdrom id=0004fb0000150000cd7223d8105042c5.iso name="OL6 U1" \
description="Oracle Linux 6 Update 1"
```

### See Also

- [Section A.68, “importVirtualCdrom”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.52 edit VirtualDisk

Edits a virtual disk.

### Syntax

```
edit VirtualDisk instance [size=value] [shareable= { yes | no }] [name=value]
[description=value]
```

Where *instance* is:

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

### Description

This command edits a virtual disk in 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.

Option	Description
<code>size=<i>value</i></code>	The size of the virtual disk in GiB.
<code>shareable= { <i>yes</i>   <i>no</i> }</code>	Whether the virtual disk is shareable. Shareable disks have read/write privileges in multiple virtual machines and should be used with caution.
<code>name=<i>value</i></code>	A name to identify the virtual disk.
<code>description=<i>value</i></code>	Optional description for the virtual disk.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyVirtualDisk</code> .

### Examples

#### Example A.70 Editing a virtual disk

```
OVM> edit VirtualDisk name=MyVMDisk name='New name for MyVMDisk' description='My virtual disk'
```

#### Example A.71 Resizing a virtual disk

```
OVM> edit VirtualDisk name=MyVMDisk size=20
```

### See Also

- [Section A.69, “importVirtualDisk”](#)
- [Section A.28, “create VirtualDisk”](#)
- [Section A.16, “clone VirtualDisk”](#)
- [Section A.35, “delete”](#)

- [Section A.75, “refresh”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.53 edit VlanGroup

Edits a VLAN group.

### Syntax

```
edit VlanGroup instance [name=value] [description=value]
```

Where *instance* is:

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

### Description

Edits a VLAN group. To edit the structure and content of a VLAN group, use the [add Port](#), [remove Port](#), [create VlanSegment](#) and [delete VlanSegment](#) commands.

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.

Option	Description
<code>name=<i>value</i></code>	A name for the VLAN group.
<code>description=<i>value</i></code>	Optional description for the VLAN group.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyVLANGroup</code> .

### Examples

#### Example A.72 Editing a VLAN group

```
OVM> edit VlanGroup name=MyVLANGroup description="My VLAN Group"
```

### See Also

- [Section A.29, “create VlanGroup”](#)
- [Section A.30, “create VlanSegment”](#)
- [Section A.9, “add Port”](#)
- [Section A.45, “edit Port”](#)
- [Section A.35, “delete”](#)

- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.54 edit VlanInterface

Edits a VLAN interface.

### Syntax

```
edit VlanInterface instance [mtu=value] [addressType= { none | staticIp | dynamic }]
[ipAddress=value] [netMask=value] [name=value] [description=value]
```

Where *instance* is:

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

### Description

This command edits 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.

Option	Description
<code>mtu=value</code>	The MTU value.
<code>addressType= { none   staticIp   dynamic }</code>	The network addressing type.
<code>ipAddress=value</code>	The IP address to use for the VLAN interface.
<code>netMask=value</code>	The netmask to use for the VLAN interface.
<code>name=value</code>	A name to identify the VLAN interface.
<code>description=value</code>	Optional description for the VLAN interface.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyVlanInterface</code> .

### Examples

#### Example A.73 Editing a VLAN interface

```
OVM> edit VlanInterface name=MyVlanInterface mtu=1500 addressType=dynamic
```

### See Also

- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.55 edit Vm

Edits a virtual machine.

## Syntax

```
edit Vm instance [memory=value] [memoryLimit=value] [cpuCountLimit=value]
[cpuCount=value] [cpuPriority=value] [cpuUtilizationCap=value] [highAvailability= {
yes | no }]
[osType= { WIN_2003 | WIN_2008 | WIN_7 | WIN_VISTA | OTHER_WIN | OL_4 | OL_5 | OL_6
|RHL_4 | RHL_5 | RHL_6 | LINUX_RECOVERY | OTHER_LINUX | SOLARIS_10 | SOLARIS_11 |
OTHER_SOLARIS | NONE }]
[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 }] [networkBootPath=value] [name=value]
[description=value]
```

Where *instance* is:

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

## Description

This command edits 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.

Option	Description
<code>memory=value</code>	The memory size the virtual machine is allocated in MB.
<code>memoryLimit=value</code>	The maximum memory size the virtual machine can be allocated in MB.
<code>cpuCountLimit=value</code>	The maximum number of processors the virtual machine can be allocated.
<code>cpuCount=value</code>	The number of processors the virtual machine is allocated.
<code>cpuPriority=value</code>	The CPU priority of the virtual machine. A value between 1 and 100; the higher the number, the more priority the CPU is given.
<code>cpuUtilizationCap=value</code>	The maximum percentage to which the virtual CPUs can receive scheduled time. A value between 1 and 100; the higher the number, the more scheduled time the CPU is given.
<code>highAvailability= { yes   no }</code>	Whether to enable High Availability.
<code>osType= { WIN_2003   WIN_2008   WIN_7   WIN_VISTA   OTHER_WIN   OL_4   OL_5   OL_6   RHL_4   RHL_5   RHL_6   LINUX_RECOVERY   OTHER_LINUX   SOLARIS_10   SOLARIS_11   OTHER_SOLARIS   NONE }</code>	The operating system of the virtual machine.

Option	Description
<code>mouseType= { OS_DEFAULT   PS2_MOUSE   USB_MOUSE   USB_TABLET }</code>	The mouse type of the virtual machine.
<code>domainType= { XEN_HVM   XEN_HVM_PV_DRIVERS   XEN_PVM   LDOMS_PVM   UNKNOWN }</code>	The domain type of the virtual machine.
<code>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 }</code>	The keyboard mapping to use for the virtual machine.
<code>bootOrder= { PXE   DISK   CDROM }</code>	The boot media order for the virtual machine. Enter options separated by commas (,), for example: <code>bootOrder= 'CDROM,DISK'</code>  If you use the <code>PXE</code> boot option to boot from network-based installation media, also use the <code>networkBootPath</code> parameter.
<code>networkBootPath=value</code>	The location at which the installation media (mounted ISO file) is located when creating a PVM guest.
<code>name=value</code>	A name to identify the virtual machine.
<code>description=value</code>	Optional description for the virtual machine.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyVM</code> .

## Examples

### Example A.74 Edit the disk boot order a virtual machine

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

## See Also

- [Section A.17, “clone Vm”](#)
- [Section A.14, “add Vm”](#)
- [Section A.100, “start”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.56 edit VmDiskMapping

Edits the virtual machine disk mapping object.

### Syntax

```
edit VmDiskMapping instance [name=value] [description=value]
```

Where *instance* is:

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


## 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.

Option	Description
<code>name=value</code>	A name to identify the disk mapping.  <div style="display: flex; align-items: center;">  <div> <p><b>Tip</b></p> <p>To find this name after a virtual disk is mapped to a virtual machine, use the <code>list VmDiskMapping</code> command.</p> </div> </div>
<code>description=value</code>	Optional description for the disk mapping object.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyDiskMap</code> .

## Examples

### Example A.75 Editing a virtual disk mapping object

```
OVM> edit VmDiskMapping id=0004fb0000130000409cd9340443e257 name=MyDiskMap
```

## See Also

- [Section A.32, “create VmDiskMapping”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.57 edit Vnic

Edits a VNIC.

## Syntax

```
edit Vnic instance [network=value] [name=value] [description=value]
```

Where *instance* is:

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

## Description

This command edits a VNIC on a network.

## Options

The following table shows the available options for this command.

Option	Description
<code>network=<i>value</i></code>	The name or ID of the network on which the VNIC is to reside. Editing this parameter will assign the VNIC to the specified network.
<code>name=<i>value</i></code>	A MAC address for the VNIC.
<code>description=<i>value</i></code>	Optional description for the VNIC.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyVNIC</code> .

## Examples

### Example A.76 Editing a VNIC

```
OVM> edit Vnic id=0004fb00000700007fa68ffd2011539f name=00:21:f6:00:00:18 network=MyVMNetwork
```

## See Also

- [Section A.33, “create Vnic”](#)
- [Example A.147, “Generating multiple VNICs”](#)
- [Section A.4, “addAvailableVnic Vm”](#)
- [Section A.15, “add Vnic”](#)
- [Section A.86, “remove Vnic”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.58 edit VolumeGroup

Edits a volume group object.

## Syntax

```
edit VolumeGroup instance [name=value] [description=value]
```

Where *instance* is:



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

## Description

This command edits a volume group object.

## Options

The following table shows the available options for this command.

Option	Description
<code>network=<i>value</i></code>	The name or ID of the volume group object.
<code>name=<i>value</i></code>	A name for the volume group object.
<code>description=<i>value</i></code>	Optional description for the volume group object.
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyVolumeGroup</code> .

## Examples

### Example A.77 Editing a volume group

```
OVM> edit VolumeGroup name=MyVolumeGroup name=MyNewName
```

## See Also

- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.59 edit YumConfig

Edits the Yum configuration.

## Syntax

```
edit YumConfig [baseUrl=value] [gpgKeyCheck= { yes | no }] [gpgKey=value]
```

## Description

This command edits the Yum configuration used to update and upgrade Oracle VM Servers.

## Options

The following table shows the available options for this command.

Option	Description
<code>baseUrl=<i>value</i></code>	The URL to access the Yum repository.
<code>gpgKeyCheck= { yes   no }</code>	Whether to use a GPG key for the Yum repository.
<code>gpgKey=<i>value</i></code>	The GPG key for the Yum repository.

---

## Examples

### Example A.78 Editing a Yum repository

```
OVM> edit YumConfig baseUrl=http://example.com/OVM3/Server/ gpgKeyCheck=yes \  
gpgKey=http://example.com/OVM3/RPM-GPG-KEY
```

## See Also

- [Section A.94, “show”](#)
- [Section A.71, “initiateYumUpgrade”](#)

## A.60 exit

Exits/Quits the CLI.

## Syntax

```
exit
```

## Description

This command exits/quits the CLI.

## Options

This command does not take any arguments or provide any options.

## Examples

### Example A.79 Exiting the CLI

```
OVM> exit
```

## See Also

- [Section A.65, “help”](#)

## A.61 getEvents

Lists the events for an object.

## Syntax

```
getEvents { AccessGroup | FileServer | FileSystem | PhysicalDisk | Repository |  
SanServer | Server | ServerPool | Vm | VolumeGroup } instance type= { All | Ackable }  
amount=value
```

Where *instance* is:

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

## Description

This command lists the events for an object.

## Options

The following table shows the available options for this command. To acknowledge an event, use the `ackEvent` command.

Option	Description
<code>{ AccessGroup   FileServer   FileSystem   PhysicalDisk   Repository   SanServer   Server   ServerPool   Vm   VolumeGroup }</code>	The object for which to list events.
<code>type= { All   Ackable }</code>	The event type. The <code>All</code> option lists all event types, and the <code>Ackable</code> option lists all events that can be acknowledged.
<code>amount=value</code>	The number of events to list.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code> .

## Examples

### Example A.80 Listing Oracle VM Server events

```
OVM> getEvents Server name=MyServer type=Ackable amount=10
```

## See Also

- [Section A.2, “ackEvent”](#)

## A.62 getIncompatibleReasons

Lists any reasons which may cause virtual machine operations to fail.

## Syntax

```
getIncompatibleReasons checkType= { VM_START | VM_MIGRATE_BY_USER | VM_PLACEMENT }
vm=value server=value
```

## Description

This command lists any reasons which may cause virtual machine operations to fail. Virtual machine operations may be placing a virtual machine (moving a stopped virtual machine), migrating a virtual machine (live migration of a running virtual machine), or starting a virtual machine. It may be useful to use this command to check for any possible failures before performing any of these virtual machine operations.

## Options

The following table shows the available options for this command.

Option	Description
<code>checkType= { VM_START   VM_MIGRATE_BY_USER   VM_PLACEMENT }</code>	The category type against which to check virtual machine migration compatibility. The options are: <ul style="list-style-type: none"> <li>• <code>VM_START</code>: Reasons why the virtual machine may not start on the target Oracle VM Server.</li> </ul>

Option	Description
	<ul style="list-style-type: none"> <li><code>VM_MIGRATE_BY_USER</code>: Reasons why the virtual machine may not be able to be migrated to the target Oracle VM Server.</li> <li><code>VM_PLACEMENT</code>: Reasons why the stopped virtual machine may not be placed on the target Oracle VM Server.</li> </ul>
<code>vm=value</code>	The name or ID of the virtual machine to check migration compatibility.
<code>server=value</code>	The name or ID of the Oracle VM Server that is the intended target of the virtual machine migration.

## Examples

### Example A.81 Listing virtual machine placement incompatibility reasons

```
OVM> getIncompatibleReasons checkType=VM_PLACEMENT vm=MyVM server=MyServer
```

## See Also

- [Section A.74, “migrate Vm”](#)

## A.63 getJobsUsingRange

Lists jobs within a date range.

## Syntax

```
getJobsUsingRange from=value to=value
```

## Description

This command lists jobs within a date range.

## Options

The following table shows the available options for this command.

Option	Description
<code>from=value</code>	The start date and time from which to list jobs. The format to use is <code>MM-dd-yyyy HH:mm</code> .
<code>to=value</code>	The end date and time from which to list jobs. The format to use is <code>MM-dd-yyyy HH:mm</code> .

## Examples

### Example A.82 Listing jobs in a date range

```
OVM> getJobsUsingRange from="07-20-2012 12:00" to="07-22-2012 24:00"
```

## See Also

- [Section A.64, “getLatestNumberOfJobs”](#)

- [Section A.1, “abort Job”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.64 getLatestNumberOfJobs

Lists the recent jobs.

### Syntax

```
getLatestNumberOfJobs amount=value
```

### Description

This command lists the recent jobs.

### Options

The following table shows the available options for this command.

Option	Description
<code>amount=<i>value</i></code>	The number of recent jobs to list.

### Examples

#### Example A.83 Listing recent jobs

```
OVM> getLatestNumberOfJobs amount=10
```

### See Also

- [Section A.63, “getJobsUsingRange”](#)
- [Section A.1, “abort Job”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.65 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.84 Using the Help Command

```
OVM> help
```

## See Also

- [Section A.95, “showallcustomcmds”](#)
- [Section A.96, “showcustomcmds”](#)
- [Section A.98, “showobjtypes”](#)

## A.66 importAssembly

Imports and adds an assembly to a storage repository.

## Syntax

```
importAssembly Repository instance server= { id | name } url=value
```

Where *instance* is:

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

## Description

This command imports and adds an assembly to a storage repository.

## Options

The following table shows the available options for this command.

Option	Description
<code>Repository=instance</code>	The storage repository in which to import the assembly.
<code>server= { id   name }</code>	The Oracle VM Server to use to perform the assembly import job. This can be either the name or the ID.
<code>url=value</code>	The URL of the assembly.
<code>{ id=id   name=name }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code> .

## Examples

### Example A.85 Importing an assembly to a storage repository

```
OVM> importAssembly Repository name=MyRepository server=MyServer \
url=http://example.com/assemblies/myassembly.ova
```

## See Also

- [Section A.34, “createTemplatesFromAssembly”](#)
- [Section A.38, “edit Assembly”](#)
- [Section A.35, “delete”](#)
- [Section A.75, “refresh”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.67 importTemplate

Imports and adds a virtual machine template to a storage repository.

### Syntax

```
importTemplate Repository=instance server= { id | name } url=value
```

Where *instance* is:

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

### 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.

Option	Description
<code>Repository=<i>instance</i></code>	The storage repository in which to import the virtual machine template.
<code>server= { <i>id</i>   <i>name</i> }</code>	The Oracle VM Server to use to perform the virtual machine template import job. This can be either the name or the ID.
<code>url=<i>value</i></code>	The URL of the virtual machine template.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code> .

### Examples

#### Example A.86 Importing a virtual machine template to a storage repository

```
OVM> importTemplate Repository name=MyRepository server=MyServer \
url=http://example.com/OVM_OL5U6_X86_64_PVM_10GB.tgz
```

## See Also

- [Section A.70, “importVirtualMachine”](#)
- [Section A.17, “clone Vm”](#)
- [Section A.75, “refresh”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.68 importVirtualCdrom

Imports and adds a virtual CDROM/ISO file to a storage repository.

### Syntax

```
importVirtualCdrom Repository=instance server= { id | name } url=value
```

Where *instance* is:

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

### 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.

Option	Description
<code>Repository=<i>instance</i></code>	The storage repository in which to import the virtual CDROM/ISO file.
<code>server= { <i>id</i>   <i>name</i> }</code>	The Oracle VM Server to use to perform the virtual CDROM/ISO file import job. This can be either the name or the ID.
<code>url=<i>value</i></code>	The URL of the virtual CDROM/ISO file.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code> .

### Examples

#### Example A.87 Importing a virtual CDROM/ISO file to a storage repository

```
OVM> importVirtualCdrom Repository name=MyRepository server=MyServer \
url=http://example.com/isos/myiso.iso
```

## See Also

- [Section A.51, “edit VirtualCdrom”](#)



- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.69 importVirtualDisk

Imports and adds a virtual disk file to a storage repository.

### Syntax

```
importVirtualDisk Repository=instance server= { id | name } url=value
```

Where *instance* is:

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

### 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.

Option	Description
<code>Repository=<i>instance</i></code>	The storage repository in which to import the virtual disk file.
<code>server= { <i>id</i>   <i>name</i> }</code>	The Oracle VM Server to use to perform the virtual disk file import job. This can be either the name or the ID.
<code>url=<i>value</i></code>	The URL of the virtual disk file.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code> .

### Examples

#### Example A.88 Importing a virtual disk file to a storage repository

```
OVM> importVirtualDisk Repository name=MyRepository server=MyServer \
url=http://example.com/vdisks/myvdisk.img
```

### See Also

- [Section A.28, “create VirtualDisk”](#)
- [Section A.52, “edit VirtualDisk”](#)
- [Section A.16, “clone VirtualDisk”](#)
- [Section A.35, “delete”](#)
- [Section A.75, “refresh”](#)
- [Section A.73, “list”](#)

- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.70 importVirtualMachine

Imports and adds a virtual machine to a storage repository.

### Syntax

```
importVirtualMachine Repository=instance server= { id | name } url=value
```

Where *instance* is:

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

### 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.

Option	Description
<code>Repository=<i>instance</i></code>	The storage repository in which to import the virtual machine.
<code>server= { <i>id</i>   <i>name</i> }</code>	The Oracle VM Server to use to perform the virtual machine import job. This can be either the name or the ID.
<code>url=<i>value</i></code>	The URL of the virtual machine. To import a multi-file virtual machine, enter each URL in a comma separated list, for example: <code>url=http://myexample.com/System-sda.img,http://myexample.com/vm.cfg</code>
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code> .

### Examples

#### Example A.89 Importing a virtual machine to a storage repository

```
OVM> importVirtualMachine Repository name=MyRepository server=MyServer \  
url="http://example.com/OVM_OL5U6_X86_64_PVM_10GB.tgz"
```

### See Also

- [Section A.14, “add Vm”](#)
- [Section A.67, “importTemplate”](#)
- [Section A.17, “clone Vm”](#)
- [Section A.75, “refresh”](#)

- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.71 initiateYumUpgrade

Upgrades an Oracle VM Server.

### Syntax

```
initiateYumUpgrade Server instance
```

Where *instance* is:

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

### Description

This command updates or upgrades an Oracle VM Server using the Yum repository. This command places the Oracle VM Server into maintenance mode, checks for any updates in the Yum 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.

### Options

The following table shows the available options for this command.

Option	Description
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyServer</i> .

### Examples

#### Example A.90 Upgrading an Oracle VM Server

```
OVM> initiateYumUpgrade Server name=MyServer
```

### See Also

- [Section A.48, “edit Server”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.72 kill

Kills an Oracle VM Server or virtual machine.

### Syntax

```
kill { Server | Vm } instance
```

Where *instance* is:

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

## Description

This command kills an Oracle VM Server or virtual machine.

## Options

The following table shows the available options for this command.

Option	Description
{ <i>Server</i>   <i>Vm</i> }	The object to kill, either an Oracle VM Server or a virtual machine.
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example name= <i>MyVM</i> .

## Examples

### Example A.91 Killing an Oracle VM Server

```
OVM> kill Server name=MyServer
```

### Example A.92 Killing a virtual machine

```
OVM> kill Vm name=MyVM
```

## See Also

- [Section A.100, “start”](#)
- [Section A.89, “restart”](#)
- [Section A.101, “stop”](#)

## A.73 list

Lists all instances of an object.

## Syntax

```
list { AccessGroup | Assembly | BondPort | FileServer | FileSystem | Job | Network |
NfsAccessGroup | PhysicalDisk | Port | Repository | SanServer | Server | ServerPool
| StorageInitiator | Tag | VirtualCdrom | VirtualDisk | VlanGroup | VlanInterface |
VlanSegment | Vm | VmDiskMapping | Vnic | VolumeGroup }
```

## Description

This command lists all instances of an object.

## Options

The following table shows the available options for this command.

Option	Description
{ <a href="#">AccessGroup</a>   <a href="#">Assembly</a>   <a href="#">BondPort</a>   <a href="#">FileServer</a>   <a href="#">FileSystem</a>   <a href="#">Job</a>   <a href="#">Network</a>   <a href="#">NfsAccessGroup</a>   <a href="#">PhysicalDisk</a>   <a href="#">Port</a>   <a href="#">Repository</a>   <a href="#">SanServer</a>   <a href="#">Server</a>   <a href="#">ServerPool</a>   <a href="#">StorageInitiator</a>   <a href="#">Tag</a>   <a href="#">VirtualCdrom</a>   <a href="#">VirtualDisk</a>   <a href="#">VlanGroup</a>   <a href="#">VlanInterface</a>   <a href="#">VlanSegment</a>   <a href="#">Vm</a>   <a href="#">VmDiskMapping</a>   <a href="#">Vnic</a>   <a href="#">VolumeGroup</a> }	<p>The object to list.</p> <p>The <a href="#">Vm</a> option lists both virtual machines and virtual machine templates. A virtual machine template is distinguishable from virtual machines in the output as the name is listed as a compressed archive file (for example, a <a href="#">.tgz</a>, or <a href="#">.zip</a> file), whereas a virtual machine does not.</p> <p>The <a href="#">VmDiskMapping</a> option lists the disk mapping objects for both virtual machines and virtual machine templates.</p>

## Examples

### Example A.93 Listing Oracle VM Servers

```
OVM> list Server
```

### Example A.94 Listing virtual machines and virtual machine templates

```
OVM> list Vm
```

### Example A.95 Listing networks

```
OVM> list Network
```

### Example A.96 Listing virtual machine and virtual machine templates disk mapping

```
OVM> list VmDiskMapping
```

## See Also

- [Section A.94, “show”](#)

## A.74 migrate Vm

Migrates a running virtual machine.

## Syntax

```
migrate Vm instance { destServer=value | destServerPool=value }
```

Where *instance* is:

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

## 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, or a stopped virtual machine to another server pool. To migrate the virtual machine to the **Unassigned Virtual Machines** folder (undeploy it), do not supply a destination.

To check whether there are any conditions that may cause the virtual machine migration to fail, use the [getIncompatibleReasons](#) command before migrating the virtual machine.

## Options

The following table shows the available options for this command.

Option	Description
{ <code>destServer=value</code>   <code>destServerPool=value</code> }	The name or ID of the Oracle VM Server or server pool on which to migrate the virtual machine.
{ <code>id=id</code>   <code>name=name</code> }	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code> .

## Examples

### Example A.97 Migrating a virtual machine to an Oracle VM Server

```
OVM> migrate Vm name=MyVM destServer=MyServer
```

### Example A.98 Migrating a virtual machine to another server pool

```
OVM> migrate Vm name=MyVM destServerPool=MyServerPool
```

### Example A.99 Migrating a virtual machine to the Unassigned Virtual Machines folder

```
OVM> migrate Vm name=MyVM
```

## See Also

- [Section A.62, “getIncompatibleReasons”](#)
- [Section A.14, “add Vm”](#)
- [Section A.85, “remove Vm”](#)
- [Section A.100, “start”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.75 refresh

Refreshes configuration information about an object in Oracle VM Manager.

## Syntax

```
refresh { Assembly | FileServer | FileSystem | PhysicalDisk | Repository | SanServer | Server } instance
```

Where *instance* is:

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

## Description

This command reads the configuration information about the object and updates the Oracle VM Manager database repository.

## Options

The following table shows the available options for this command.

Option	Description
{ <code>Assembly</code>   <code>FileServer</code>   <code>FileSystem</code>   <code>PhysicalDisk</code>   <code>Repository</code>   <code>SanServer</code>   <code>Server</code> }	The object to be refreshed.
{ <code>id=id</code>   <code>name=name</code> }	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=server1</code> .

## Examples

### Example A.100 Refreshing a file server

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

### Example A.101 Refreshing a SAN server

```
OVM> refresh SanServer name=MyISCSIServer
```

### Example A.102 Refreshing a physical disk

```
OVM> refresh PhysicalDisk id=0004fb0000180000efa8fd003a5f1613
```

### Example A.103 Refresh a file system

```
OVM> refresh FileSystem name=nfs:/mnt/voll/poolfs01
```

### Example A.104 Refresh a storage repository

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

## A.76 refreshStorageLayer

Refreshes the storage layer on an Oracle VM Server.

## Syntax

```
refreshStorageLayer Server instance
```

Where *instance* is:

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

## Description

This command refreshes the storage layer on an Oracle VM Server.

## Options

The following table shows the available options for this command.

Option	Description
{ <i>id=id</i>   <i>name=name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyServer</i> .

## Examples

### Example A.105 Refreshing Oracle VM Server storage

```
OVM> refreshStorageLayer Server name=MyServer
```

## See Also

- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.77 remove BondPort

Removes a bond port from a network object.

## Syntax

```
remove BondPort instance from { Network | VlanGroup } instance
```

Where *instance* is:

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

## Description

This command removes a bond port from a network object.

## Options

The following table shows the available options for this command.

Option	Description
{ <i>Network</i>   <i>VlanGroup</i> }	The network object from which to remove the bond port.
{ <i>id=id</i>   <i>name=name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyNetwork</i> .

## Examples

### Example A.106 Removing a bonded port from a network

```
OVM> remove BondPort id=0004fb000020000065822cb7bb9ec296 from Network name=MyVMNetwork
```

## See Also

- [Section A.19, “create BondPort”](#)



- [Section A.39, “edit BondPort”](#)
- [Section A.6, “add BondPort”](#)
- [Section A.80, “remove Port”](#)
- [Section A.9, “add Port”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.78 remove FileSystem

Removes a file system from a file server access group.

### Syntax

```
remove FileSystem instance from NfsAccessGroup instance
```

Where *instance* is:

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

### Description

This command removes a file system from a file server access group.

### Options

The following table shows the available options for this command.

Option	Description
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyNfsAccessGroup</i> .

### Examples

#### Example A.107 Removing a file system from a file server access group

```
OVM> remove FileSystem name=nfs:/mnt/voll/repo01 from NfsAccessGroup name=MyNFSAccessGroup
```

### See Also

- [Section A.7, “add FileSystem”](#)
- [Section A.22, “create NfsAccessGroup”](#)
- [Section A.43, “edit NfsAccessGroup”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

- [Section A.35, “delete”](#)

## A.79 remove PhysicalDisk

Removes a physical disk from a SAN storage access group.

### Syntax

```
remove PhysicalDisk instance from AccessGroup instance
```

Where *instance* is:

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

### 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.

Option	Description
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyDisk</i> .

### Examples

#### Example A.108 Removing physical disk from a SAN storage access group

```
OVM> remove PhysicalDisk id=0004fb00001800007ee6dbda7b4461cb from AccessGroup \
name='Default access group @ MyISCServer'
```

### See Also

- [Section A.23, “create PhysicalDisk”](#)
- [Section A.8, “add PhysicalDisk”](#)
- [Section A.44, “edit PhysicalDisk”](#)
- [Section A.35, “delete”](#)
- [Section A.75, “refresh”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.80 remove Port

Removes an Ethernet port from a network object.

## Syntax

```
remove Port instance from { BondPort | Network | VlanGroup } instance
```

Where *instance* is:

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

## Description

This command removes an Ethernet port from a network object.

## Options

The following table shows the available options for this command.

Option	Description
{ BondPort   Network   VlanGroup }	The network object from which to remove the Ethernet port.
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyNetwork</i> .

## Examples

### Example A.109 Removing a port from a network

```
OVM> remove Port id=0004fb0000200000be8fa354cb7d98ae from Network name=MyVMNetwork
```

## See Also

- [Section A.19, “create BondPort”](#)
- [Section A.39, “edit BondPort”](#)
- [Section A.6, “add BondPort”](#)
- [Section A.9, “add Port”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.81 remove Server

Removes an Oracle VM Server from an object.

## Syntax

```
remove Server instance from { ServerPool | FileServer | SanServer | Repository | NfsAccessGroup } instance
```

Where *instance* is:

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

## Description

This command removes an Oracle VM Server from either a server pool, file server, SAN server, storage repository or file server access group.

## Options

The following table shows the available options for this command.

Option	Description
{ <code>ServerPool</code>   <code>FileServer</code>   <code>SanServer</code>   <code>Repository</code>   <code>NfsAccessGroup</code> }	The object from which to remove the Oracle VM Server.
{ <code>id=id</code>   <code>name=name</code> }	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code> .

## Examples

### Example A.110 Removing an Oracle VM Server from a server pool

```
OVM> remove Server name=MyServer from ServerPool name=MyServerPool
```

### Example A.111 Removing an admin server from file server

```
OVM> remove Server name=MyServer from FileServer name=MyNFSServer
```

### Example A.112 Removing an admin server from a SAN server

```
OVM> remove Server name=MyServer from SanServer name=MyISCSIServer
```

### Example A.113 Removing an Oracle VM Server from a file server access group

```
OVM> remove Server name=MyServer from NfsAccessGroup name=MyNFSAccessGroup
```

### Example A.114 Removing (unpresenting) an Oracle VM Server from a storage repository

```
OVM> remove Server name=MyServer from Repository name=MyRepository
```

## See Also

- [Section A.36, “discoverServer”](#)
- [Section A.10, “add Server”](#)
- [Section A.48, “edit Server”](#)
- [Section A.75, “refresh”](#)
- [Section A.100, “start”](#)
- [Section A.101, “stop”](#)
- [Section A.89, “restart”](#)
- [Section A.72, “kill”](#)

- [Section A.71, “initiateYumUpgrade”](#)
- [Section A.81, “remove Server”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)
- [Section A.61, “getEvents”](#)

## A.82 remove ServerPool

Removes a server pool from a storage repository.

### Syntax

```
remove ServerPool instance from Repository instance
```

Where *instance* is:

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

### Description

This command unrepresents a storage repository from the Oracle VM Servers in a server pool. To unrepresent a storage repository to an individual Oracle VM Server, use the [remove Server](#) command.

### Options

The following table shows the available options for this command.

Option	Description
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyServerPool</i> .

### Examples

#### Example A.115 Unrepresenting a storage repository from a server pool

```
OVM> remove ServerPool name=MyServerPool from Repository name=MyNFSRepository
```

### See Also

- [Section A.11, “add ServerPool”](#)
- [Section A.81, “remove Server”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.83 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=id | name=name }
```

## 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.

Option	Description
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <code>name=MyAccessGroup</code> .

## Examples

### Example A.116 Removing a storage initiator

```
OVN> remove StorageInitiator name=iqn.1988-12.com.oracle:d72d82d0817f from AccessGroup \
name='Default access group @ MyISCSIServer'
```

## See Also

- [Section A.12, “add StorageInitiator”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.84 remove Tag

Removes a tag from an object.

## Syntax

```
remove Tag instance from { ServerPool | Server | Vm } instance
```

Where *instance* is:

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

## 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.

Option	Description
{ <a href="#">ServerPool</a>   <a href="#">Server</a>   <a href="#">Vm</a> }	The object from which to remove the tag.
{ <code>id=id</code>   <code>name=name</code> }	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyTag</code> .

## Examples

### Example A.117 Removing a tag from a server pool

```
OVM> remove Tag name=MyTag from ServerPool name=MyServerPool
```

## See Also

- [Section A.27, “create Tag”](#)
- [Section A.50, “edit Tag”](#)
- [Section A.13, “add Tag”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.85 remove Vm

Removes a virtual machine from an Oracle VM Server.

## Syntax

```
remove Vm instance from Server instance
```

Where *instance* is:

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

## Description

This command removes a virtual machine from an Oracle VM Server. In the Oracle VM Manager user interface, the virtual machine is moved to the **Unassigned Virtual Machines** folder in the **Servers and VMs** tab. 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.

Option	Description
{ <code>id=id</code>   <code>name=name</code> }	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyServer</code> .

## Examples

### Example A.118 Removing a virtual machine from an Oracle VM Server

```
OVM> remove Vm name=MyVM from Server name=MyServer
```

## See Also

- [Section A.14, “add Vm”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.86 remove Vnic

Removes a VNIC from a virtual machine.

## Syntax

```
remove Vnic instance from VM instance
```

Where *instance* is:

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

## Description

This command removes a VNIC from a virtual machine.

## Options

The following table shows the available options for this command.

Option	Description
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyVM</i> .

## Examples

### Example A.119 Removing a VNIC from a virtual machine

```
OVM> remove Vnic name=00:21:f6:00:00:00 from Vm name=MyVM
```

## See Also

- [Section A.33, “create Vnic”](#)
- [Example A.147, “Generating multiple VNICs”](#)
- [Section A.4, “addAvailableVnic Vm”](#)
- [Section A.15, “add Vnic”](#)



- [Section A.57, “edit Vnic”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.87 removeAccessHost

Removes an access host from an iSCSI server.

### Syntax

```
removeAccessHost SanServer instance accessHost=value
```

Where *instance* is:

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

### Description

This command removes an access host from an iSCSI server when using a SAN server with multipath capability. At least one access host must be set. Multipath is not supported with the generic SAN server storage plug-in. This is not applicable to fibre channel storage.

### Options

The following table shows the available options for this command.

Option	Description
<code>accessHost=<i>value</i></code>	The hostname or IP address for the access host. To find the hostname or IP address, use the <a href="#">show</a> command to display information about the SAN Server.
<code>{ <i>id=id</i>   <i>name=name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyISCSIServer</code> .

### Examples

#### Example A.120 Removing a SAN server access host

```
OVM> removeAccessHost SanServer name=MyISCSIServer accessHost=10.172.76.131
```

### See Also

- [Section A.3, “addAccessHost”](#)
- [Section A.25, “create SanServer”](#)
- [Section A.47, “edit SanServer”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.88 removeNfsRefreshServer

Removes an export server from a file server.

### Syntax

```
removeNfsRefreshServer FileServer instance nfsRefreshServer=value
```

Where *instance* is:

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

### 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.

Option	Description
<i>FileServer instance</i>	The name or ID of the file server.
<i>nfsRefreshServer=value</i>	The name or ID of the Oracle VM Server to be removed as a refresh server.
{ <i>id=id</i>   <i>name=name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyServer</i> .

### Examples

#### Example A.121 Removing a refresh server from a file server

```
OVM> removeNfsRefreshServer FileServer name=MyNFSServer nfsRefreshServer=MyServer1
```

### See Also

- [Section A.5, “addNfsRefreshServer”](#)
- [Section A.75, “refresh”](#)
- [Section A.94, “show”](#)

## A.89 restart

Restarting an Oracle VM Server or virtual machine.

### Syntax

```
restart { Server | Vm } instance
```

Where *instance* is:

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

## Description

This command restarts an Oracle VM Server or virtual machine.

## Options

The following table shows the available options for this command.

Option	Description
{ <i>Server</i>   <i>Vm</i> }	The object to restart, either an Oracle VM Server or a virtual machine.
{ id= <i>id</i>   name= <i>name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyVM</i> .

## Examples

### Example A.122 Restarting an Oracle VM Server

```
OVM> restart Server name=MyServer
```

### Example A.123 Restarting a virtual machine

```
OVM> restart Vm name=MyVM
```

## See Also

- [Section A.100, “start”](#)
- [Section A.101, “stop”](#)
- [Section A.72, “kill”](#)

## A.90 resume Vm

Resumes a suspended virtual machine.

## Syntax

```
resume Vm instance
```

Where *instance* is:

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

## Description

This command resumes a suspended virtual machine.

## Options

The following table shows the available options for this command.

Option	Description
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyVM</code> .

## Examples

### Example A.124 Resuming a virtual machine

```
OVM> resume Vm name=MyVM
```

## See Also

- [Section A.102, “suspend Vm”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.91 sendVmMessage

Sends a key/value pair to a running virtual machine.

## Syntax

```
sendVmMessage Vm instance key=value message=value log= { yes | no }
```

Where *instance* is:

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

## Description

This command sends a key/value pair to a running virtual machine.

## Options

The following table shows the available options for this command.

Option	Description
<code>key=<i>value</i></code>	The message key.
<code>message=<i>value</i></code>	The message content.
<code>log= { <i>yes</i>   <i>no</i> }</code>	Whether to log the message.
<code>{ id=<i>id</i>   name=<i>name</i> }</code>	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyVM</code> .

## Examples

### Example A.125 Sending a message to a virtual machine

```
OVM> sendVmMessage Vm name=MyVM key=com.oracle.linux.network.device.0 message=eth0 log=no
```

## A.92 set

Sets configuration options for the CLI.

## Syntax

```
set { EndlineChars= { CRLF | CR | LF } | OutputMode= { Verbose | Sparse | Xml } }
```

## Description

This command sets configuration options for the CLI.

## Options

The following table shows the available options for this command.

Option	Description
<code>EndlineChars= { CRLF   CR   LF }</code>	Sets the end of line character to use for 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 end of line characters to <code>CR</code> .  <code>CRLF</code> is used on Microsoft Windows systems, <code>CR</code> is used on early Apple systems, and <code>LF</code> is used on Linux, and Unix-like systems such a modern Apple systems.
<code>OutputMode= { Verbose   Sparse   Xml }</code>	Sets the output mode for command results. <code>Verbose</code> includes the command, status, time and time zone. <code>Sparse</code> returns just the results without the header files provided by <code>Verbose</code> . <code>Xml</code> returns the results in XML format.

## Examples

### Example A.126 Setting end of line characters

```
OVM> set EndlineChars=LF
```

### Example A.127 Setting output mode to XML

```
OVM> set OutputMode=Xml
```

## A.93 setNtp

Sets the list of NTP servers to use for synchronizing time between Oracle VM Servers.

## Syntax

```
setNtp list=value
```

## Description

This command sets the list of Network Time Protocol (NTP) time source servers to be used by Oracle VM Servers to maintain synchronization of time. To push the list to Oracle VM Servers, use the `syncNtp` command.

## Options

The following table shows the available options for this command.

Option	Description
<code>list=value</code>	A comma separated list of NTP servers, in the order they should be used by the Oracle VM Servers.

## Examples

### Example A.128 Setting the list of NTP servers

```
OVM> setNtp list=1.ntp.example.com,2.ntp.example.com,3.ntp.example.com,4.ntp.example.com
```

## See Also

- [Section A.97, “showNtp”](#)
- [Section A.103, “syncNtp”](#)

## A.94 show

Shows information about an object.

## Syntax

```
show {{{ AccessGroup | Assembly | BondPort | FileServer | FileSystem | Job | Network |
NfsAccessGroup | PhysicalDisk | Port | Repository | SanServer | Server | ServerPool
| StorageInitiator | Tag | VirtualCdrom | VirtualDisk | VlanGroup | VlanInterface |
VlanSegment | Vm | VmDiskMapping | Vnic name=name | VolumeGroup }instance} | YumConfig }
```

Where *instance* is:

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

## 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.

A Job object does not have a `name` attribute, only an `id` attribute. The `show Job name=name` command is the same as entering `show Job id=id`. You can use these two options interchangeably. Any `name` attributes are automatically converted to `ids`.

The `YumConfig` option does not require an `instance` identifier.

## Options

The following table shows the available options for this command.

Option	Description
{ <code>AccessGroup</code>   <code>Assembly</code>   <code>BondPort</code>   <code>FileServer</code>   <code>FileSystem</code>   <code>Job</code>   <code>Network</code>   <code>NfsAccessGroup</code>   <code>PhysicalDisk</code>   <code>Port</code>   <code>Repository</code>   <code>SanServer</code>   <code>Server</code>   <code>ServerPool</code>   <code>StorageInitiator</code>   <code>Tag</code>   <code>VirtualCdrom</code>   <code>VirtualDisk</code> }	The object about which to show information.

Option	Description
<a href="#">VlanGroup</a>   <a href="#">VlanInterface</a>   <a href="#">VlanSegment</a>   <a href="#">Vm</a>   <a href="#">VmDiskMapping</a>   <a href="#">Vnic name=name</a>   <a href="#">VolumeGroup</a> }	
<a href="#">YumConfig</a>	Select to display information about the Yum repository configuration.
{ <a href="#">id=id</a>   <a href="#">name=name</a> }	The instance of the object using either the <a href="#">id</a> or <a href="#">name</a> option, for example <a href="#">name=MyServer</a> .

## Examples

### Example A.129 Show details about an Oracle VM Server

```
OVM> show Server name=MyServer
```

### Example A.130 Show details about a virtual machine

```
OVM> show Vm name=MyVM
```

### Example A.131 Show details about a network

```
OVM> show Network id=0004fb0010ff705
```

### Example A.132 Show Yum repository configuration

```
OVM> show YumConfig
```

### Example A.133 Show 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 Utilities Guide](#) for information on using the Guest Additions packages.

## See Also

- [Section A.73, "list"](#)

## A.95 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.134 Showing all custom commands

```
OVM> showallcustomcmds
```

## See Also

- [Section A.65, “help”](#)
- [Section A.96, “showcustomcmds”](#)

## A.96 showcustomcmds

Shows available custom commands for an object type.

## Syntax

```
showcustomcmds {{ AccessGroup | Assembly | FileServer | FileSystem | Job | PhysicalDisk |  
Repository | SanServer | Server | ServerPool | VirtualDisk | Vm | VolumeGroup }}
```

## 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.

Option	Description
{ <a href="#">AccessGroup</a>   <a href="#">Assembly</a>   <a href="#">FileServer</a>   <a href="#">FileSystem</a>   <a href="#">Job</a>   <a href="#">PhysicalDisk</a>   <a href="#">Repository</a>   <a href="#">SanServer</a>   <a href="#">Server</a>   <a href="#">ServerPool</a>   <a href="#">VirtualDisk</a>   <a href="#">Vm</a>   <a href="#">VolumeGroup</a> }	The object type that you wish to list custom commands for.

## Examples

### Example A.135 Show Custom Commands for an Oracle VM Server

```
OVM> showcustomcmds Server
```



---

**Example A.136 Show Custom Commands for a virtual machine**

```
OVM> showcustomcmds VM
```

**Example A.137 Show Custom Commands for a Repository**

```
OVM> showcustomcmds Repository
```

## See Also

- [Section A.65, “help”](#)
- [Section A.95, “showallcustomcmds”](#)
- [Section A.98, “showobjtypes”](#)

## A.97 showNtp

Lists the NTP servers to use for synchronizing time between Oracle VM Servers.

### Syntax

```
showNtp
```

### Description

This command lists the Network Time Protocol (NTP) time source servers to be used by Oracle VM Servers to maintain synchronization of time. To edit the list, use the [setNtp](#) command. To push the list to Oracle VM Servers, use the [syncNtp](#) command.

### Examples

**Example A.138 Showing the list of NTP servers**

```
OVM> showNtp
```

## See Also

- [Section A.93, “setNtp”](#)
- [Section A.103, “syncNtp”](#)

## A.98 showobjtypes

Provides a list of all object types recognized within the system.

### Syntax

```
showobjtypes
```

### Description

This command provides a list of all object types recognized by the system. This command is useful to assist in determining which object types can be acted on.

---

## Options

This command does not take any arguments or provide any options.

## Examples

### Example A.139 Showing all object types

```
OVN> showobjtypes
```

## See Also

- [Section A.65, “help”](#)
- [Section A.96, “showcustomcmds”](#)

## A.99 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.140 Showing the CLI/Oracle VM Manager version number

```
OVN> showversion
```

## See Also

- [Section A.65, “help”](#)

## A.100 start

Starts an Oracle VM Server or virtual machine.

## Syntax

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

Where *instance* is:

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

## Description

This command starts an Oracle VM Server or virtual machine.

## Options

The following table shows the available options for this command.

Option	Description
{ <code>Server</code>   <code>Vm</code> }	The object to start, either an Oracle VM Server or a virtual machine.
{ <code>id=id</code>   <code>name=name</code> }	The instance of the object using either the <code>id</code> or <code>name</code> option, for example <code>name=MyVM</code> .

## Examples

### Example A.141 Starting an Oracle VM Server

```
OVM> start Server name=MyServer
```

### Example A.142 Starting a virtual machine

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

## See Also

- [Section A.89, “restart”](#)
- [Section A.101, “stop”](#)
- [Section A.72, “kill”](#)

## A.101 stop

Stops an Oracle VM Server or virtual machine.

## Syntax

```
stop { Server | Vm } instance
```

Where *instance* is:

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

## Description

This command stops an Oracle VM Server or virtual machine.

## Options

The following table shows the available options for this command.

Option	Description
{ <code>Server</code>   <code>Vm</code> }	The object to stop, either an Oracle VM Server or a virtual machine.

Option	Description
{ <i>id=id</i>   <i>name=name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyVM</i> .

## Examples

### Example A.143 Stopping an Oracle VM Server

```
OVM> stop Server name=MyServer
```

### Example A.144 Stopping a virtual machine

```
OVM> stop Vm name=MyVM
```

## See Also

- [Section A.100, “start”](#)
- [Section A.89, “restart”](#)
- [Section A.72, “kill”](#)

## A.102 suspend Vm

Suspends a running virtual machine.

## Syntax

```
suspend Vm instance
```

Where *instance* is:

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

## Description

This command suspends a running virtual machine.

## Options

The following table shows the available options for this command.

Option	Description
{ <i>id=id</i>   <i>name=name</i> }	The instance of the object using either the <i>id</i> or <i>name</i> option, for example <i>name=MyVM</i> .

## Examples

### Example A.145 Suspending a virtual machine

```
OVM> suspend Vm name=MyVM
```

## See Also

- [Section A.90, “resume Vm”](#)

- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

## A.103 syncNtp

Pushes the list of NTP servers to use for synchronizing time to Oracle VM Servers.

### Syntax

```
syncNtp
```

### Description

This command pushes the list of Network Time Protocol (NTP) time source servers to Oracle VM Servers to maintain synchronization of time. To edit the list, use the [setNtp](#) command. To show the list, use the [showNtp](#) command.

### Examples

#### Example A.146 Pushing the list of NTP servers to Oracle VM Servers

```
OVM> syncNtp
```

### See Also

- [Section A.93, “setNtp”](#)
- [Section A.97, “showNtp”](#)

## A.104 vnicCreate

Creates one or more VNICs.

### Syntax

```
vnicCreate segment4=value segment5=value segment6=value numberToGenerate=value
```

### Description

This command generates one more VNICs based on an initial MAC address. To create a single VNIC and assign it to a network, use the [create Vnic](#) command.

### Options

The following table shows the available options for this command.

Option	Description
<code>segment4=<i>value</i></code>	The 4th octet of the initial MAC address.
<code>segment5=<i>value</i></code>	The 5th octet of the initial MAC address.
<code>segment6=<i>value</i></code>	The 6th octet of the initial MAC address.
<code>numberToGenerate=<i>value</i></code>	The number of VNICs to create.

## Examples

### Example A.147 Generating multiple VNICs

```
OVM> vnicCreate segment4=00 segment5=00 segment6=50 numberToGenerate=20
```

## See Also

- [Section A.33, “create Vnic”](#)
- [Section A.4, “addAvailableVnic Vm”](#)
- [Section A.15, “add Vnic”](#)
- [Section A.57, “edit Vnic”](#)
- [Section A.86, “remove Vnic”](#)
- [Section A.35, “delete”](#)
- [Section A.73, “list”](#)
- [Section A.94, “show”](#)

---

# Index

## Symbols

?  
command, 5

## A

abort  
job, 47  
access group  
add file system, 52  
add Oracle VM Server, 55  
add physical disk, 53  
create, 63  
delete, 83  
edit, 86  
events, 110  
list, 120  
remove file system, 125  
remove physical disk, 126  
show, 138  
access host  
add, 49  
remove, 133  
AccessGroup  
showcustomcmds, 140  
acknowledge  
event, 48  
add  
access host, 49  
file system to access group, 52  
network bond, 51, 63  
NFS refresh server, 51  
Oracle VM Server to file server, 55  
Oracle VM Server to file server access group, 55  
Oracle VM Server to repository, 55  
Oracle VM Server to SAN server, 55  
Oracle VM Server to server pool, 55  
physical disk to access group, 53  
port, 54  
refresh server, 51  
server pool to repository, 56  
storage initiator, 57  
tag, 58  
virtual machine to Oracle VM Server, 59  
VNIC, 60  
assembly  
create virtual machine from, 83  
create virtual machine template from, 83  
delete, 83  
edit, 87  
import, 114

list, 120  
refresh, 122  
show, 138  
Assembly  
showcustomcmds, 140

## B

bond port  
add Ethernet port, 54  
delete, 83  
list, 120  
show, 138

## C

case sensitivity, 14  
CDROM  
adding to a virtual machine, 80  
delete, 83  
edit, 101  
import, 116  
list, 120  
mapping to a virtual machine, 80  
show, 138  
CLI  
connect to, 1  
clone  
disk, 61  
template, 61  
virtual disk, 60  
virtual machine, 61  
connect  
to CLI, 1  
create  
access group, 62  
disk, 69, 75  
Ethernet network, 66  
file server, 65  
file server access group, 68  
NFS access group, 68  
physical disk, 69  
repository, 50, 70  
SAN server, 71  
SAN storage access group, 63  
server pool, 73  
storage repository, 70  
tag, 74  
template from assembly, 83  
virtual disk, 75  
virtual machine, 78  
virtual machine from assembly, 83  
VLAN group, 76  
VLAN segment, 77  
VNIC, 82

---

VNICs, 145

## D

delete

- access group, 83
- assembly, 83
- bond port, 83
- CDROM, 83
- disk, 83
- disk mapping, 83
- file server, 83
- file system, 83
- ISO, 83
- network, 83
- Oracle VM Server, 83
- physical disk, 83
- refresh server, 134
- repository, 83
- SAN server, 83
- server pool, 83
- tag, 83
- virtual disk, 83
- virtual machine, 83
- virtual machine disk mapping, 83
- VirtualCDROM, 83
- VLAN group, 83
- VLAN segment, 83
- VNIC, 83
- volume group, 110

discover

- file server, 65
- Oracle VM Server, 85
- SAN server, 71

disk

- clone, 61
- create, 69, 75
- delete, 83
- edit, 93, 102
- events, 110
- import, 117
- list, 120
- show, 138

disk mapping

- delete, 83
- edit, 106
- list, 120
- show, 138

## E

edit

- access group, 86
- assembly, 87
- CDROM, 101

- disk, 93, 102
- disk mapping, 106
- Ethernet network, 91
- Ethernet port, 94
- file server, 89
- file server access group, 92
- file system, 90
- ISO, 101
- network bond, 88
- NFS access group, 92
- Oracle VM Server, 97
- physical disk, 93
- repository, 95
- SAN server, 96
- SAN storage access group, 86
- server pool, 99
- tag, 100
- virtual disk, 102
- virtual machine, 104
- virtual machine disk mapping, 106
- VLAN group, 103
- VLAN interface, 104
- VNIC, 107
- volume group, 108
- Yum configuration, 109
- end of line character
  - set, 5, 136
- escape
  - special characters, 5
- Ethernet network
  - create, 66
  - edit, 91
  - list, 138
- Ethernet port
  - add bond, 63
  - add to network object, 54
  - edit, 94
  - edit bond, 88
  - list, 138
- event
  - acknowledge, 48
- events
  - access group, 110
  - disk, 110
  - file server, 110
  - file system, 110
  - Oracle VM Server, 110
  - physical disk, 110
  - repository, 110
  - SAN server, 110
  - server pool, 110
  - virtual machine, 110
- exit, 110



---

## F

### file server

- add Oracle VM Server, 55
- add refresh server, 51
- create access group, 68
- delete, 83
- discover, 65
- edit, 89
- edit access group, 92
- events, 110
- list, 120
- refresh, 122
- remove Oracle VM Server, 127
- remove refresh server, 134
- show, 138

### file server access group

- add Oracle VM Server, 55
- delete, 83
- list, 120
- show, 138

### file system

- add to access group, 52
- delete, 83
- edit, 90
- events, 110
- list, 120
- refresh, 122
- remove from access group, 125
- show, 138

### FileServer

- showcustomcmds, 140

### FileSystem

- showcustomcmds, 140

## H

### help, 113

- command, 5

## I

### import

- assembly, 114
- CDROM, 116
- disk, 117
- ISO, 116
- template, 115
- virtual disk, 117
- virtual machine, 118

### ISCSI

- multipath configuration, 49, 133

### ISO

- delete, 83
- edit, 101
- import, 116

- list, 120

- show, 138

### ISO file

- mapping to a virtual machine, 80

## J

### job

- abort, 47
- list, 120
- list in date range, 112
- list recent, 113
- show, 138

### Job

- showcustomcmds, 140

## K

### kill

- Oracle VM Server, 119
- virtual machine, 119

## L

- list, 120

### lock

- timeout, 14

- log file, 15

## M

### maintenance mode

- Oracle VM Server, 97, 97

### master Oracle VM Server

- change, 99

### message

- send to virtual machine, 136

### migrate

- virtual machine, 121

### migration

- virtual machine, 111

### move

- virtual machine, 59, 131

### multipath

- add access hosts, 49
- remove access hosts, 133

## N

### network

- add port, 54
- add port bond, 51, 63
- create, 66
- delete, 83
- edit, 91
- edit Ethernet port, 94
- edit port bond, 88
- list, 120

---

- remove bond port, 124
- remove Ethernet port, 126
- show, 138
- network bond
  - add Ethernet port, 54
- network port
  - add to network, 54
  - bond, 51, 63
  - edit bond, 88
  - remove bond port, 124
  - remove Ethernet port, 126
- Network Time Protocol, 137, 141, 145
- NFS
  - add refresh server, 51
- NFS access group
  - create, 68
  - delete, 83
  - edit, 92
  - list, 120
  - show, 138
- NFS file system
  - add to access group, 52
  - remove from access group, 125
- NFS refresh server
  - remove, 134
- NTP, 137, 141, 145

## O

- Oracle VM Server
  - add to file server, 55
  - add to file server access group, 55
  - add to repository, 55
  - add to SAN server, 55
  - add to server pool, 55
  - add virtual machine, 59
  - create port bond, 63
  - delete, 83
  - discover, 85
  - edit, 97
  - edit Ethernet port, 94
  - edit port bond, 88
  - events, 110
  - kill, 119
  - list, 120
  - refresh, 122
  - refresh storage layer, 123
  - remove from file server, 127
  - remove from repository, 127
  - remove from SAN server, 127
  - remove from server pool, 127
  - remove virtual machine, 131
  - restart, 134
  - show, 138

- start, 142
- stop, 143
- take ownership, 97
- time synchronization, 137, 141, 145
- upgrade, 119
- output mode
  - set, 5, 5, 136

## P

- physical disk
  - add to access group, 53
  - create, 69
  - delete, 83
  - edit, 93
  - events, 110
  - list, 120
  - refresh, 122
  - remove from access group, 126
  - show, 138
- PhysicalDisk
  - showcustomcmds, 140
- place
  - virtual machine, 111
- port
  - add bond, 51, 63
  - add to network, 54
  - edit, 88, 94
  - list, 120
  - remove, 126
  - remove bond, 124
  - SSH, 14
- present
  - server pool to repository, 56
- public key
  - expiration, 14

## Q

- quit, 110

## R

- refresh, 122
  - Oracle VM Server storage layer, 123
- refresh server
  - add, 51
  - remove, 134
- regular expressions, 14
- remove
  - access host, 133
  - bond port, 124
  - Ethernet port, 126
  - file system from access group, 125
  - NFS refresh server, 134
  - Oracle VM Server from file server, 127

---

- Oracle VM Server from repository, 127
- Oracle VM Server from SAN server, 127
- Oracle VM Server from server pool, 127
- physical disk from access group, 126
- server pool from repository, 129
- storage initiator, 129
- tag, 130
- virtual machine from Oracle VM Server, 131
- VNIC, 132
- repository
  - add Oracle VM Server, 55
  - create, 50, 70
  - delete, 83
  - edit, 95
  - events, 110
  - list, 120
  - refresh, 122
  - remove Oracle VM Server, 127
  - show, 138
- Repository
  - showcustomcmds, 140
- restart
  - Oracle VM Server, 134
  - virtual machine, 134
- resume
  - virtual machine, 135
- roles
  - Oracle VM Server, 97, 97

## S

- SAN access group
  - list, 120
- SAN server
  - add Oracle VM Server, 55
  - delete, 83
  - discover, 71
  - edit, 96
  - events, 110
  - list, 120
  - refresh, 122
  - remove Oracle VM Server, 127
  - show, 138
- SAN storage
  - add physical disk to access group, 53
  - create access group, 63
  - edit access group, 86
  - remove physical disk from access group, 126
- SanServer
  - showcustomcmds, 140
- script
  - samples, 11
- server
  - refresh, 122

- Server
  - showcustomcmds, 140
- server pool
  - add Oracle VM Server, 55
  - add to repository, 56
  - create, 73
  - delete, 83
  - edit, 99
  - events, 110
  - list, 120
  - remove from repository, 129
  - remove Oracle VM Server, 127
  - show, 138
- ServerPool
  - showcustomcmds, 140
- set, 136
- show, 138
- showallcustomcmds, 139
- showcustomcmds, 140
- showobjtypes, 141
- showversion, 142
- special characters
  - escape, 5
- SSH
  - port, 14
  - timeout, 1, 14
- start
  - Oracle VM Server, 142
  - virtual machine, 111, 142
- stop
  - Oracle VM Server, 143
  - virtual machine, 143
- storage
  - refresh, 123
- storage initiator
  - add, 57
  - list, 120
  - remove, 130
  - show, 138
- storage repository
  - create, 70
  - present server pool, 56
  - unpresent server pool, 129
- suspend
  - virtual machine, 144

## T

- tag
  - add, 58
  - create, 74
  - delete, 83
  - edit, 100
  - list, 120

---

- remove, 130
- show, 138
- template
  - clone, 61
  - create from assembly, 83
  - import, 115
  - list, 120
- time synchronization, 137, 141, 145
- timeout
  - locks, 14
  - SSH, 1, 14
  - SSH public key, 14

## U

- unpresent
  - server pool from repository, 129
- upgrade
  - Oracle VM Server, 119

## V

- version
  - show, 142
- virtual disk
  - adding to a virtual machine, 80, 80
  - clone, 61
  - create, 75
  - delete, 83
  - edit, 102
  - import, 117
  - list, 120
  - mapping to a virtual machine, 80
  - show, 138
- virtual machine
  - add to Oracle VM Server, 59
  - add VNIC, 60
  - adding a CDROM, 80
  - adding a virtual disk, 80
  - adding an ISO file, 80
  - clone, 61
  - create, 78
  - delete, 83
  - delete disk mapping, 83
  - disk mapping, 80
  - edit, 104
  - events, 110
  - import, 118
  - kill, 119
  - list, 120
  - list disk mapping, 120
  - migrate, 121
  - migration, 111
  - move, 59, 131
  - place, 111

- remove from Oracle VM Server, 131
- remove VNIC, 132
- restart, 134
- resume, 135
- send message, 136
- show, 138
- show disk mapping, 138
- start, 111, 142
- stop, 143
- suspend, 144
- virtual machine disk mapping
  - edit, 106
- virtual machine template
  - list, 120
- VirtualCDROM
  - delete, 83
  - list, 120
- VirtualCDRom
  - show, 138
- VirtualDisk
  - showcustomcmds, 140
- VLAN group
  - add Ethernet port, 54
  - add port bond, 51
  - create, 76
  - delete, 83
  - edit, 103
  - list, 120
  - remove bond port, 124
  - remove Ethernet port, 126
  - show, 138
- VLAN interface
  - edit, 104
  - list, 120
  - show, 138
- VLAN segment
  - create, 77
  - delete, 83
  - list, 120
  - show, 138
- vm
  - import, 118
  - send message, 136
- VM
  - show IP address, 138
- Vm
  - showcustomcmds, 140
- VNIC
  - add to virtual machine, 60
  - create, 82
  - create multiple, 145
  - delete, 83
  - edit, 107
  - list, 120

---

- remove from virtual machine, 132
- show, 138
- volume group
  - edit, 108
  - events, 110
  - list, 120
  - show, 138
- VolumeGroup
  - showcustomcmds, 140

## **X**

- XML
  - set output, 136

## **Y**

- Yum
  - upgrade Oracle VM Server, 119
- Yum configuration
  - edit, 109
  - show, 138

