

Oracle® VM

Utilities Guide for Release 3



E35333-05
March 2014

Oracle® VM: Utilities Guide for Release 3

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	v
1 Audience	v
2 Documentation Accessibility	v
3 Related Documents	v
4 Command Syntax	vi
5 Conventions	vi
1 Oracle VM Utilities Overview	1
1.1 Downloading and Installing Oracle VM Utilities	1
1.1.1 Installing Oracle VM Utilities on Oracle VM Manager	1
1.1.2 Installing Oracle VM Utilities on Oracle Linux	2
1.1.3 Directory Structure of Oracle VM Utilities	3
1.2 Introduction to Oracle VM Utilities	3
1.3 Introduction to Oracle VM Guest Additions	4
2 Using the Oracle VM Utilities	5
2.1 Using Oracle VM Manager Control	6
2.1.1 Command Line Options and Parameters of ovm_managercontrol	6
2.1.2 Examples of ovm_managercontrol Usage	6
2.2 Using Oracle VM Server Control	8
2.2.1 Command Line Options and Parameters of ovm_servercontrol	8
2.2.2 Examples of ovm_servercontrol Usage	9
2.3 Using Oracle VM Virtual Machine Control	11
2.3.1 Command Line Options and Parameters of ovm_vmcontrol	11
2.3.2 Examples of ovm_vmcontrol Usage	12
2.4 Using Oracle VM Retrieve Disk	13
2.4.1 Command Line Options and Parameters of ovm_vmdisk	13
2.4.2 Examples of ovm_vmdisk Usage	13
2.5 Using Oracle VM Virtual Machine Messaging	14
2.5.1 Command Line Options and Parameters of ovm_vmmmessage	14
2.5.2 Examples of ovm_vmmmessage Usage	15
2.6 Using Oracle VM Pool Control	16
2.6.1 Command Line Options and Parameters of ovm_poolcontrol	16
2.6.2 Examples of ovm_poolcontrol Usage	16
2.7 Using Oracle VM Repository Control	17
2.7.1 Command Line Options and Parameters of ovm_repocontrol	17
2.7.2 Examples of ovm_repocontrol Usage	18
2.8 Using Oracle VM Hostd For Metrics Messaging	19
2.8.1 Command Line Options	20
2.8.2 Example Usage	20
2.8.3 Using <code>vm-dump-metrics</code> To Retrieve Host Statistics Within a VM	20
3 Using the Oracle VM Guest Additions	23
3.1 Features of the Oracle VM Guest Additions	23
3.2 Installing Oracle VM Guest Additions	23
3.3 Upgrading the Oracle VM Guest Additions	24
3.4 The Oracle VM Guest Additions Daemon	25
3.4.1 Configuring the Oracle VM Guest Additions Daemon to Run as a Service	26
3.4.2 Using the Oracle VM Guest Additions Daemon to Enable First-Boot Configuration	26
3.4.3 Using the Messaging Channel	26
3.5 Displaying the Guest IP Address	27
3.6 The Oracle VM Template Configuration Script and Modules	27
3.6.1 Overview of the Template Configuration Script	28
3.6.2 Enabling and Disabling Configuration Modules	29

3.6.3 Key-Value Pairs Used By Available Configuration Modules	29
3.6.4 Triggering Configuration Changes	30
3.6.5 Developing Oracle VM Template Configuration Modules	31
4 Other Oracle VM Tools	37
4.1 Oracle VM Administrator Tool	37
4.1.1 Managing Users with Oracle VM Administrator Tool	37
4.1.2 Log Rotation Options	42
4.2 Oracle VM Diagnostic Capture	43

Preface

Table of Contents

1 Audience	v
2 Documentation Accessibility	v
3 Related Documents	v
4 Command Syntax	vi
5 Conventions	vi

The Oracle VM Utilities Guide is your reference to the administrative and diagnostic command line tools provided as an add-on or as part of the Oracle VM installation. The command line tools described in this guide are intended to help administrators perform certain tasks and collect troubleshooting information more easily. The operations made possible by these tools are typically not available through the *Oracle VM Manager* user interface. Usage of the tools requires root privileges on the system.

1 Audience

This document is intended for system administrators who install, configure and manage the Oracle VM environment. We assume that you have a solid understanding of the product and are familiar with virtualization in general, Web technologies and the Oracle Linux operating system.

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 Security Guide](#)
- [Oracle VM Command Line Interface User's 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/rdsd/c0t1d0s6 of=/dev/rst0 bs=10b \ count=10000</pre>
braces { }	Braces indicate required items: <pre>.DEFINE {macro}</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 Oracle VM Command Line Interface 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
boldface	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 Oracle VM Utilities Overview

Table of Contents

1.1 Downloading and Installing Oracle VM Utilities	1
1.1.1 Installing Oracle VM Utilities on Oracle VM Manager	1
1.1.2 Installing Oracle VM Utilities on Oracle Linux	2
1.1.3 Directory Structure of Oracle VM Utilities	3
1.2 Introduction to Oracle VM Utilities	3
1.3 Introduction to Oracle VM Guest Additions	4

The Oracle VM Utilities are a collection of command line scripts that allow you to perform a set of basic management tasks on *Oracle VM Servers* and virtual machines in an Oracle VM environment. These utilities are particularly useful to administrators who need to execute certain operations quickly and/or repeatedly. Using the command line scripts makes these tasks quicker and easier to perform.



Warning

The command line scripts in the Oracle VM Utilities are provided as-is for your convenience. With the exception of the operations required to perform *hard partitioning* as described in the *Oracle VM User's Guide* and in the whitepaper *Hard Partitioning With Oracle VM Server for x86* located at <http://www.oracle.com/technetwork/server-storage/vm/ovm-hardpart-168217.pdf>, they are not officially and formally supported by Oracle.

1.1 Downloading and Installing Oracle VM Utilities

The Oracle VM Utilities are available for download as a `.zip` file via the Oracle VM Downloads page:

<http://www.oracle.com/technetwork/server-storage/vm/downloads/index.html>

The Oracle VM Utilities can be installed directly on the system running *Oracle VM Manager* or, alternately, on any server or desktop system running Oracle Linux. The sections below describe how and where to install these utilities. An overview of the installation directory structure and files is also provided.

1.1.1 Installing Oracle VM Utilities on Oracle VM Manager

When installing Oracle VM Utilities on the same server as Oracle VM Manager, it is recommended that you put them in the software directory of the Oracle VM Manager application. The Oracle VM Utilities are bundled in a `.zip` archive. To install, simply download the archive and extract in the appropriate directory.

In the example below, we have already downloaded the `.zip` file and show how to copy the `.zip` file into the installation directory, where it is unzipped.

```
# cp ovm_utils_archive.zip /u01/app/oracle/ovm-manager-3/
# cd /u01/app/oracle/ovm-manager-3
# unzip ovm_utils_archive.zip
```

At this point, the command line scripts are ready to use. Note that these scripts execute a Java program and use the default Java VM on the *host computer*. On the Oracle VM Manager host, the appropriate Java VM is installed as part of the Oracle VM Manager application in `/u01/app/oracle/java`. No further configuration is required.

1.1.2 Installing Oracle VM Utilities on Oracle Linux

When installing Oracle VM Utilities on an Oracle Linux server or desktop computer, you may put them in the installation directory of your choice; for example: `/usr/local/bin`. The Oracle VM Utilities are bundled in a `.zip` archive. To install, simply download the archive, copy and extract in the appropriate directory.

```
# cp ovm_utils_archive.zip /usr/local/bin/oracle/
# cd /usr/local/bin/oracle
# unzip ovm_utils_archive.zip
```



Caution

Oracle VM Utilities **do not** work with the Open JDK or GNU Compiler for Java (GCJ). You must install a standard Java VM, version 1.6.x, and make sure it is defined as the default Java VM by the `JAVA_HOME` environment variable or included in the `PATH` variable on your system.

The command line scripts in Oracle VM Utilities execute a Java program and use the default Java VM on the host computer. To verify the exact path to the Java executable and the active Java version, use the following commands:

```
# which java
/usr/java
# java -version
java version "1.6.0_26"
Java(TM) SE Runtime Environment (build 1.6.0_26-b03)
Java(TM) Server VM (build 20.1-b02, mixed mode)
```

If a standard Java VM, version 1.6.x, is not available on your system, download and install the version suited for your platform from <http://java.com/en/download/>. To ensure that the appropriate Java VM version is used, set the `JAVA_HOME` and `PATH` variables as follows:

1. Edit the Oracle Linux shell profile.

```
# vi /etc/profile
- OR -
# vi /etc/bashrc
```

2. Add the lines below to the profile. If your Java path does not match `/usr/java`, replace with the actual path on your system.

```
JAVA_HOME=/usr/java
export JAVA_HOME
PATH=$PATH:$JAVA_HOME/bin
export PATH
```

3. Save the file. Log out and log back in to activate your changes.
4. Verify the Java path and version again, to make sure that the correct Java VM is used.

```
# which java
/usr/java
# java -version
java version "1.6.0_26"
Java(TM) SE Runtime Environment (build 1.6.0_26-b03)
Java(TM) Server VM (build 20.1-b02, mixed mode)
```

At this point, the command line scripts are ready to use.

1.1.3 Directory Structure of Oracle VM Utilities

When you unzip the Oracle VM Utilities archive, a subdirectory named `ovm_utils` is created, containing these utility scripts:

- `ovm_managercontrol`
- `ovm_servercontrol`
- `ovm_vmcontrol`
- `ovm_vmdisks`
- `ovm_vmmmessage`
- `ovm_poolcontrol`
- `ovm_repocontrol`

For each of these utility scripts, a *man page* is also included. These help files can be found in `.../ovm_utils/man/man8`.

```
# ls /u01/app/oracle/ovm-manager-3/ovm_utils/man/man8/  
ovm_managercontrol.8  ovm_repocontrol.8    ovm_vmcontrol.8    ovm_vmmmessage.8  
ovm_poolcontrol.8    ovm_servercontrol.8  ovm_vmdisks.8
```

Use the `man` command to display the help; for example:

```
# man man/man8/ovm_servercontrol.8
```

1.2 Introduction to Oracle VM Utilities

The Oracle VM Utilities are command line scripts used to execute certain basic operations on your Oracle VM environment. These utilities make a connection to the Oracle VM Manager host server, using the server host name and an administrative user name and password. After authentication, commands can be submitted to Oracle VM Manager from the command line.

The Oracle VM Utilities are:

- `ovm_servercontrol`: command line script to perform administrative operations on a physical Oracle VM Server
- `ovm_vmcontrol`: command line script to perform administrative operations on a virtual machine
- `ovm_vmdisks`: command line script to list virtual and raw physical disks attached to a virtual machine in order to facilitate backup
- `ovm_vmmmessage`: command line script to send and retrieve messages in the form of key/value pairs to and from a running virtual machine that has Oracle VM Guest Additions installed.
- `ovm_poolcontrol`: command line script to perform administrative operations on a pool of Oracle VM Servers
- `ovm_repocontrol`: command line script to retrieve information about storage repositories and refresh their contents

Command line parameters and options, along with several examples, can be found in [Chapter 2, Using the Oracle VM Utilities](#)

**Caution**

Certain functions of the Oracle VM Utilities are restricted to version 3.1 and higher. If you attempt to execute an incompatible script on Oracle VM version 3.0, a warning message will be displayed to inform you that the feature is not supported.

1.3 Introduction to Oracle VM Guest Additions

The Oracle VM Utilities include a *messaging* tool, which allows sending key-value pairs to a virtual machine, or guest, and retrieve such messages from the guest. The `ovm_vmmmessage` utility requires the Oracle VM Guest Additions to be installed on the guest side. The current edition of Oracle VM Guest Additions includes message channel and guest IP information. The Oracle VM Guest Additions allows direct integration between guest software and the virtualization layer, to assist in orchestration and automation of complex, multi-VM deployments.

A brief description of the installation, configuration and features of Oracle VM Guest Additions can be found in [Chapter 3, Using the Oracle VM Guest Additions](#)

Chapter 2 Using the Oracle VM Utilities

Table of Contents

2.1 Using Oracle VM Manager Control	6
2.1.1 Command Line Options and Parameters of ovm_managercontrol	6
2.1.2 Examples of ovm_managercontrol Usage	6
2.2 Using Oracle VM Server Control	8
2.2.1 Command Line Options and Parameters of ovm_servercontrol	8
2.2.2 Examples of ovm_servercontrol Usage	9
2.3 Using Oracle VM Virtual Machine Control	11
2.3.1 Command Line Options and Parameters of ovm_vmcontrol	11
2.3.2 Examples of ovm_vmcontrol Usage	12
2.4 Using Oracle VM Retrieve Disk	13
2.4.1 Command Line Options and Parameters of ovm_vmdisk	13
2.4.2 Examples of ovm_vmdisk Usage	13
2.5 Using Oracle VM Virtual Machine Messaging	14
2.5.1 Command Line Options and Parameters of ovm_vmmmessage	14
2.5.2 Examples of ovm_vmmmessage Usage	15
2.6 Using Oracle VM Pool Control	16
2.6.1 Command Line Options and Parameters of ovm_poolcontrol	16
2.6.2 Examples of ovm_poolcontrol Usage	16
2.7 Using Oracle VM Repository Control	17
2.7.1 Command Line Options and Parameters of ovm_repocontrol	17
2.7.2 Examples of ovm_repocontrol Usage	18
2.8 Using Oracle VM Hostd For Metrics Messaging	19
2.8.1 Command Line Options	20
2.8.2 Example Usage	20
2.8.3 Using <code>vm-dump-metrics</code> To Retrieve Host Statistics Within a VM	20

This chapter gives you more detail on using the Oracle VM Utilities, including command line examples.



Note

While many of the command line tools provided by Oracle VM Utilities are still useful, as of Oracle VM 3.2.1 many of the actions that they allow you to perform can be achieved using the Oracle VM Command Line Interface, which provides a wider range of control over objects within Oracle VM Manager. The Oracle VM Command Line Interface tends to provide better information about objects and affords more flexibility when dealing with particular object types. See the [Oracle VM Command Line Interface User's Guide](#) for more information.

The Oracle VM Utilities make a connection to the Oracle VM Manager host server, using the server host name and an administrative user name and password. Oracle VM Manager listens on port 54321(TCP, local access only) or 54322 (Secure TCP). All utilities share a common set of parameters to connect to the Oracle VM Manager instance. The following parameters are required and used by each :

- -u <user name of an Oracle VM Manager admin user>
- -p <password corresponding with the admin user name>
- -h <host name of the server running Oracle VM Manager>

2.1 Using Oracle VM Manager Control

The `ovm_managercontrol` utility operates at the level of the Oracle VM Manager. It allows the administrator to execute basic commands applicable to the configuration of the entire environment managed by the Oracle VM Manager.

2.1.1 Command Line Options and Parameters of `ovm_managercontrol`

The `ovm_managercontrol` utility has the following command line options:

- **-u** user name of an Oracle VM Manager admin user (required)
- **-p** password corresponding with the admin user name (required)
- **-h** host name of the server running Oracle VM Manager (required)
- **-X** use SSL to connect to Oracle VM Manager at `tcps://host:54322`
- **-k** name of the keystore file used for SSL
- **-P** password to open the keystore file
- **-G** enable GPG check
- **-K** URL to the GPG key
- **-B** base URL for the YUM server repository
- **-n** *CPU compatibility group* name
- **-s** name of the Oracle VM Server to add to or remove from a CPU compatibility group
- **-T** timeout in seconds for virtual machine console session (use together with `-c getsessiontimeout/setsessiontimeout`)
- **-c** console command (required)

[**commands:** `yuminfo`, `setupyum`, `addkeystore`, `keystoreinfo`, `createcpugroup`, `removecpugroup`, `addservertocpugroup`, `removeserverfromcpugroup`, `listcpugroups`, `getsessiontimeout`, `setsessiontimeout`]



Note

See the examples in the next section for typical combinations of parameters and options.

2.1.2 Examples of `ovm_managercontrol` Usage

Below are examples of `ovm_managercontrol` commands and their respective output.

- Set up a keystore file for SSL connectivity and check the keystore settings:

```
# ./ovm_managercontrol -u admin -p password -h localhost -c addkeystore -k mykeystore.ks
-P password
Oracle VM Manager Control utility 0.5.2.
Connected.
Command : addkeystore
```

```
Adding keystore.
Restart of Oracle VM Manager is required for this setting to take effect.
 (/etc/init.d/ovmm stop ; /etc/init.d/ovmm start)
Exit...

# ./ovm_managercontrol -u admin -p password -h localhost -c keystoreinfo
Oracle VM Manager Control utility 0.5.2.
Connected.
Command : keystoreinfo
Key Store information :
    Key Path      : '/u01/app/oracle/ovm-manager-3/keystore/mykeystore.ks'
Exit...
```

- Configure the YUM repository for Oracle VM Server updates:

```
# ./ovm_managercontrol -u admin -p password -h localhost -c setupyum \\  
-B http://yumserver.domain.com/repo -G -K http://secure.domain.com/key.gpg
```

This action can also be achieved using the Oracle VM Command Line Interface.

- Configure a *CPU compatibility group* and add an Oracle VM Server:

```
# ./ovm_managercontrol -u admin -p password -h localhost -c createcpugroup -n MyCPUgroup_Opteron
Oracle VM Manager Control utility 0.5.2.
Connected.
Command : createcpugroup
    Creating CPU Compatibility Group 'MyCPUgroup_Opteron'.
Exit...

# ./ovm_managercontrol -u admin -p password -h localhost -c addservertocpugroup -s  
MyServer1 -n MyCPUgroup_Opteron
Oracle VM Manager Control utility 0.5.2.
Connected.
Command : addservertocpugroup
    Adding Server to CPU Compatibility group.
Exit...
```

These actions can also be achieved using the Oracle VM Command Line Interface.

- Display the CPU compatibility group configuration, that is: the compatibility groups and their member servers:

```
# ./ovm_managercontrol -u admin -p password -h localhost -c listcpugroups
Oracle VM Manager Control utility 0.5.2.
Connected.
Command : listcpugroups
Group : MyCPUgroup_Opteron
    Server : MyServer1
    Server : MyServer2
    Server : MyServer3
    Server : MyServer6
    Server : MyServer7
    Server : MyServer8

Group : MyCPUgroup_Xeon
    Server : MyServer4
    Server : MyServer5
Exit...
```

This action can also be achieved using the Oracle VM Command Line Interface.

- Set the session timeout value for the RAS Proxy, which is used to facilitate console access to your virtual machines:

```
# ./ovm_managercontrol -u admin -p Welcome1 -h localhost -c setsessiontimeout -T 300
```

```
Oracle VM Manager
  Control utility 0.5.2.
Manager version :
  3.2.1.516
Connected.
Command :
  setsessiontimeout
Set RAS timeout to
  300 seconds.
Exit...
```

- Get the session timeout value for the RAS Proxy, which is used to facilitate console access to your virtual machines:

```
./ovm_managercontrol -u admin -p Welcomel -h localhost -c getsessiontimeout
Oracle VM Manager
  Control utility 0.5.2.
Manager version :
  3.2.1.516
Connected.
Command :
  getsessiontimeout
RAS Engine Timeout =
  300
Exit...
```

2.2 Using Oracle VM Server Control

The `ovm_servercontrol` utility operates on a physical Oracle VM Server. It allows the administrator to execute basic server commands on a given Oracle VM Server, query its status and retrieve essential information.

All of the actions available in this tool are better handled using the Oracle VM Command Line Interface, but are included for legacy reasons.

2.2.1 Command Line Options and Parameters of `ovm_servercontrol`

The `ovm_servercontrol` utility has the following command line options:

- **-u** user name of an Oracle VM Manager admin user (required)
- **-p** password corresponding with the admin user name (required)
- **-h** host name of the server running Oracle VM Manager (required)
- **-X** use SSL to connect to Oracle VM Manager at `tcps://host:54322`
- **-s** Oracle VM Server name
- **-I** Oracle VM Server ID
- **-P** agent password for the Oracle VM Server to be discovered (use together with `-c discover`)
- **-r** storage repository name (for NFS export)
- **-i** ID of the NFS export
- **-o** options for the NFS export
- **-C** client host name for the NFS export

- **-A** acknowledge *events* (use together with -c events)

Use **-A all** to acknowledge events for the Oracle VM Server and the virtual machines it hosts

- **-c** console command (required)

[**commands:** *start, stop, restart, kill, discover, mainton, maintoff, status, info, events, lock, upgrade, refresh, list, listnfsexports, createnfsexport, deletenfsexport*]



Note

The *mainton* and *maintoff* commands activate or deactivate the Oracle VM Server maintenance mode.

The *discover* command performs a *discovery* operation on an Oracle VM Server that was not yet previously discovered by Oracle VM Manager.

The *upgrade* command initiates a yum upgrade on the given Oracle VM Server.

The *list* command does not need an Oracle VM Server name argument. It lists every Oracle VM Server registered with the system.

2.2.2 Examples of ovm_servercontrol Usage

Below are examples of *ovm_servercontrol* commands and their respective output.

- List all registered Oracle VM Servers:

```
# ./ovm_servercontrol -u admin -p password -h localhost -c list
Oracle VM Server Control utility 0.5.2.
Connected.
Command : list
Server : 'OVS_01'
  ID      : '00:e0:81:4d:40:16:00:e0:81:4d:40:17:ff:ff:ff:ff'
  pool    : 'MyServerPool1'
  status  : 'Running'

Server : 'OVS_02'
  ID      : '00:e0:81:4d:5e:82:00:e0:81:4d:5e:83:ff:ff:ff:ff'
  pool    : 'unassigned'
  status  : 'Running'

Server : 'OVS_03'
  ID      : '00:e0:81:4d:5e:16:00:e0:81:4d:5e:17:ff:ff:ff:ff'
  pool    : 'MyServerPool1'
  status  : 'Running'

Server : 'OVS_04'
  ID      : '00:e0:81:4d:40:f5:00:e0:81:4d:40:be:00:e0:81:4d'
  pool    : 'unassigned'
  status  : 'Running'

Exit...
```

- Check the status of an Oracle VM Server:

```
# ./ovm_servercontrol -u admin -p password -h localhost -s OVS_03 -c status
Oracle VM Server Control utility 0.5.2.
Connected.
Command : status
Server : 'OVS_03' status : 'Running' serverpool : 'MyServerPool1'.
```

Exit...

- Display detailed information of an Oracle VM Server:

```
# ./ovm_servercontrol -u admin -p password -h localhost -s OVS_01 -c info
Oracle VM Server Control utility 0.5.2.
Connected.
Command : info
Server : OVS_01
      Status      : Running
      Management IP: 10.172.76.90
      Hostname     : OVS_01
      ID           : 00:e0:81:4d:40:16:00:e0:81:4d:40:17:ff:ff:ff:ff
      Pool         : MyServerPool1
      CPU threads  : 4
      CPU Usage    : 1.877896
      Total Memory : 32767
      Usable Memory: 31261
      Maintenance : false
      Master Server: true
      Up to date   : true
      Virtual Machines on this server :
        Name      : 'MyVM01'
        Status     : 'Stopped'
        Name      : 'MyVM02'
        Status     : 'Running'
Exit...
```

- Restart an Oracle VM Server:

```
# ./ovm_servercontrol -u admin -p password -h localhost -s OVS_02 -c restart
Oracle VM Server Control utility 0.5.2.
Connected.
Command : restart
restartserver 'OVS_02' completed.
Exit...
```

- Expose an OCFS2 repository using NFS:



Note

This example shows the creation of an NFS export and then lists the existing exports. The storage repository is exposed to two different clients, but then one export is deleted via its ID. The mount command at the end shows that the remaining export is still available.

```
# ./ovm_servercontrol -u admin -p password -h localhost -c createnfsexport -s OVS_04 \
-C NFS_02 -o rw,no_root_squash -r MyIscsiRepo
Oracle VM Server Control utility 0.5.2.
Connected.
Command : createnfsexport
Creating nfs export for repository 'MyIscsiRepo' on server.
Created repository export
Exit...

# ./ovm_servercontrol -u admin -p Manager1 -h localhost -c listnfsexports -s OVS_04
Oracle VM Server Control utility 0.5.2.
Connected.
Command : listnfsexports
NFS Exports :
      id           : 0004fb0000230000978a28e2dc85e06b
      client        : NFS_01
      options       :
      repository    : MyIscsiRepo
      path           : /OVS/Repositories/0004fb000003000031fac25b24740cca/
```



```

---
      id      : 0004fb00002300004b161fd1a0d106b7
      client   : NFS_02
      options  : rw,no_root_squash
      repository : MyIscsiRepo
      path     : /OVS/Repositories/0004fb000003000031fac25b24740cca/
---
Exit...

# ./ovm_servercontrol -u admin -p password -h localhost -c deletenfsexport -s OVS_04 \
-i 0004fb0000230000978a28e2dc85e06b
Oracle VM Server Control utility 0.5.2.
Connected.
Command : deletenfsexport
Deleting nfs export with ID '0004fb0000230000978a28e2dc85e06b'
Deleted repository export
Exit...

[root@NFS_02]# mount OVS_04:/OVS/Repositories/0004fb000003000031fac25b24740cca /mnt/repoexport
[root@NFS_02]# ls /mnt/repoexport
Assemblies  ISOs  lost+found  Templates  VirtualDisks  VirtualMachines

```

2.3 Using Oracle VM Virtual Machine Control

The `ovm_vmcontrol` utility operates on a virtual machine (VM). It passes basic virtual machine control commands from the command line to Oracle VM Manager. It allows you to start, stop, suspend, resume, kill, restart a VM. It is also possible to simply query the status of the virtual machine (Running, Stopped), and retrieve essential VM information such as memory, server pool, number of virtual CPUs and so on.

2.3.1 Command Line Options and Parameters of `ovm_vmcontrol`

The `ovm_vmcontrol` utility has the following command line options:

- **-u** user name of an Oracle VM Manager admin user (required)
- **-p** password corresponding with the admin user name (required)
- **-h** host name of the server running Oracle VM Manager (required)
- **-X** use SSL to connect to Oracle VM Manager
- **-v** virtual machine name
- **-U** virtual machine UUID
- **-t** name of target Oracle VM Server for migration of a virtual machine (use together with **-c migrate**)
- **-s** comma separated list of physical thread numbers to bind virtual CPUs to (use together with **-c vcpuset**).
- **-T** tag list, a comma separated list of strings to be associated with a virtual machine as metadata (use together with **-c settags**)
- **-F** force the virtual disks associated with a virtual machine to be deleted (use together with **-c delete**)
- **-A** acknowledge virtual machine events (use together with **-c events**)
- **-C** change a virtual machine's on_crash setting (use together with **-c setoncrash**). Options include: `preserve`, `restart`, `none`.
- **-c** console command (required)

[**commands:** `start`, `stop`, `suspend`, `resume`, `status`, `restart`, `kill`, `delete`, `info`, `fixcfg`, `events`, `lock`, `vcpuset`, `vcpuget`, `migrate`, `settags`, `gettags`, `list`, `setoncrash`, `getoncrash`]



Note

As you can see, the `ovm_vmcontrol` console command list contains a `lock` command. This special command allows you to lock a virtual machine for a certain period of time. When a virtual machine is locked by the utility, it is not possible for other tools or users (through the Oracle VM Manager user interface) to perform operations on that specific VM. This is particularly useful if you are an administrator and need to make a VM backup while the VM is stopped: you want to prevent anyone starting the VM before the backup finishes.

The `list` command does not need a VM name or UUID argument. It lists every VM and its status.

The `fixcfg` command verifies the `vm.cfg` file of the virtual machine and attempts to correct invalid entries. For example, an entry for `vncpasswd` will be cleared with this command, because the virtual machine console does not support this setting.

The `vcpuset` command hard-binds or pins virtual CPUs to threads. For example, `-c vcpuset -s (0,1,2)` physically binds `vcpu0` to `thread0`, `vcpu1` to `thread1`, `vcpu2` to `thread2`. Use the `vcpuget` command to retrieve information about pinned vCPUs for the selected virtual machine.

The `settags` and `gettags` commands are used to manage tag metadata for a given virtual machine.

2.3.2 Examples of ovm_vmcontrol Usage

Below are examples of `ovm_vmcontrol` commands and their respective output.

- Display detailed information of a particular virtual machine:

```
# ./ovm_vmcontrol -u admin -p password -h localhost -v MyVM02 -c info
Oracle VM VM Control utility 0.5.2.
Connected.
Command : info
Virtual Machine : MyVM02
  Status : Running
  Memory : 1024
  Uptime : 83 Minutes
  Server : OVS_01
  Pool   : MyServerPool1
  HA Mode: false
  VCPU   : 2
  Type   : Xen HVM
  OS     : Oracle Linux 6
vminfo 'MyVM02' completed.
```

- Check the status of a particular virtual machine:

```
# ./ovm_vmcontrol -u admin -p password -h localhost -v MyVM02 -c status
Oracle VM VM Control utility 0.5.2.
Connected.
Command : status
Virtual Machine : 'MyVM02' status : 'Running' on server 'OVS_01' serverpool : 'MyServerPool1'.
```

- Stop a particular virtual machine:

```
# ./ovm_vmcontrol -u admin -p password -h localhost -v MyVM01 -c stop
Oracle VM VM Control utility 0.5.2.
Connected.
Command : stop
stopvm 'MyVM01' completed.
```

- Start a particular virtual machine:

```
# ./ovm_vmcontrol -u admin -p password -h localhost -v MyVM01 -c start
Oracle VM VM Control utility 0.5.2.
Connected.
Command : start
startvm 'MyVM01' completed.
```

- Lock a particular virtual machine:

```
# ./ovm_vmcontrol -u admin -p password -h localhost -v MyVM01 -c lock
Oracle VM VM Control utility 0.5.2.
Connected.
Command : lock
Locking VM 'MyVM01'
Press Ctrl-C to unlock.
```

The lock command will sleep until you press **Ctrl+C** to release the lock. If the `ovm_control` utility is killed, the VM would be unlocked as well.

2.4 Using Oracle VM Retrieve Disk

The `ovm_vmdisks` utility is designed to help the administrator make backups of virtual machines, particularly when the VM's virtual disks are files in a storage repository on a remote NFS storage server.

2.4.1 Command Line Options and Parameters of `ovm_vmdisk`

The `ovm_vmdisks` utility takes a virtual machine name or UUID and lists out every virtual disk file for that VM, as well as the virtual machine configuration (`vm.cfg`) file. This allows you to take the file listings from the output of the utility and back them up to a server or tape.

The `ovm_vmdisks` utility has the following command line options:

- **-u** user name of an Oracle VM Manager admin user (required)
- **-p** password corresponding with the admin user name (required)
- **-h** host name of the server running Oracle VM Manager (required)
- **-X** use SSL to connect to Oracle VM Manager
- **-v** virtual machine name
- **-U** virtual machine UUID

If the virtual disks are actual physical devices directly attached to the VM, the utility lists the device mapper entry on the Oracle VM Server to which the VM is assigned. In case the virtual disks are files on an NFS server, the utility lists the NFS server name, mount point and file name/location.

2.4.2 Examples of `ovm_vmdisk` Usage

Below is an example of `ovm_vmdisk` usage. It shows a virtual machine with three disks, each attached in a different way:

```
# ./ovm_vmdisks -u admin -p password -h localhost -v MyVM01
Oracle VM Retrieve Disk utility 0.5.2.
Connected.
Virtual Machine : 'MyVM01' status : 'Running'.
Assigned Server : OVS_01
Virtual Disk : 'MyVM01_bootdisk' size : 20GB
  file=/OVS/Repositories/0004fb0000030000b0272c74e714ab12/VirtualDisks/ \
  0004fb00001200006e548a28cb886f42.img
  mountpoint=/dev/mapper/3300000006160a212/VirtualDisks/ \
  0004fb00001200006e548a28cb886f42.img
Virtual Disk : 'MyVM01_datadisk' size : 250GB
  file=/OVS/Repositories/0004fb00000300005b06362f2d866a98/VirtualDisks/ \
  0004fb0000120000f217eb8c0fa70eef.img
  mountpoint=nfs01:/mnt/vol2/repo03/VirtualDisks/ \
  0004fb0000120000f217eb8c0fa70eef.img
Physical Disk : 'pd003'
  device=/dev/mapper/3300000009b7b2cc4
Config File :
  file=/OVS/Repositories/0004fb00000300005b06362f2d866a98/ \
  VirtualMachines/0004fb00000600008757e38248a544e6/vm.cfg
  mountpoint=nfs01:/mnt/vol2/repo03/VirtualMachines/ \
  0004fb00000600008757e38248a544e6/vm.cfg
```

As you can see in the console output, VM 'MyVM01' is currently running. Note that backing up a running VM would be inconsistent, therefore it is highly recommended that you shut down the VM prior to making any backups. The console output shows that the virtual machine contains three disks, two virtual and one physical, and also provides the location of the virtual machine configuration file:

- **MyVM01_bootdisk:** a virtual disk on an ocfs2 storage repository. The ocfs2 repository was created on the device `/dev/mapper/3300000006160a212`
- **MyVM01_datadisk:** a virtual disk on an NFS storage server named `nfs01`. The full path to the virtual disk file on the NFS server is `/mnt/vol2/repo03/VirtualDisks/0004fb0000120000f217eb8c0fa70eef.img`
- **pd003:** a raw disk device at `/dev/mapper/3300000009b7b2cc4`
- **vm.cfg:** the virtual machine configuration file on NFS server `nfs01`. The full path to the configuration file is `nfs01:/mnt/vol2/repo03/VirtualMachines/0004fb00000600008757e38248a544e6/vm.cfg`

2.5 Using Oracle VM Virtual Machine Messaging

The `ovm_vmmmessage` utility lets you send a message to a running virtual machine, or to query the value of a message sent from within a virtual machine to Oracle VM Manager via the Oracle VM API messaging interface. These values are basic key/value pairs such as `foo=bar`.

New Oracle VM templates released by Oracle will contain a configuration utility called `ovmd`. This utility is used to perform first-boot installation configuration either locally from the virtual machine console or remotely through the messaging interface provided by this utility. The `ovmd` utility also allows the owner of the virtual machine to send messages back to Oracle VM Manager. For more information about `ovmd`, and the Oracle VM Guest Additions in general, see [Chapter 3, Using the Oracle VM Guest Additions](#).

2.5.1 Command Line Options and Parameters of `ovm_vmmmessage`

The `ovm_vmmmessage` utility has the following command line options:

- **-u** user name of an Oracle VM Manager admin user (required)
- **-p** password corresponding with the admin user name (required)

- **-h** host name of the server running Oracle VM Manager (required)
- **-X** use SSL to connect to Oracle VM Manager
- **-v** virtual machine name
- **-U** virtual machine UUID
- **-k** key to send
- **-V** value to send along with the key (required when using **-k**)
- **-q** key to query

Note that `ovm_vmmmessage` only works on a running virtual machine. To send messages you combine **-k** `<key>` and **-V** `<value>`. To retrieve messages, use **-q** `<key>`.



Caution

The virtual machine name is the name you assign during the creation of the VM. However, the same name could be assigned to several different VMs. If that is the case, you must use the **-U** option and provide the unique identifier (UUID) of the VM.

2.5.2 Examples of ovm_vmmmessage Usage

Below are examples of `ovm_vmmmessage` commands and their respective output.

- Send a message key/value pair to a virtual machine:

```
# ./ovm_vmmmessage -u admin -p password -h localhost -v MyVM02 -k foo -V bar
Oracle VM VM Message utility 0.5.2.
Connected.
VM : 'MyVM02' has status : Running.
Sending message.
Message sent successfully.
```

- Send a message key/value pair to a virtual machine, identifying the VM by its UUID:

```
# ./ovm_vmmmessage -u admin -p password -h localhost \
-U 0004fb00000600001c925eac2ad5d328 -k foo -V bar
Oracle VM VM Message utility 0.5.2.
Connected.
VM : 'MyVM02' has status : Running.
Sending message.
Message sent successfully.
```

- Retrieve a message from a virtual machine, identifying the VM by its UUID:

```
# ./ovm_vmmmessage -u admin -p password -h localhost \
-U 0004fb00000600001c925eac2ad5d328 -q foo
Oracle VM VM Message utility 0.5.2.
Connected.
VM : 'MyVM02' has status : Running.
Querying for key 'foo'.
Query successful.
Query for Key : 'foo' returned value 'bar'.
Key set 27 minutes ago.
```

2.6 Using Oracle VM Pool Control

The `ovm_poolcontrol` utility allows the administrator to quickly retrieve information about the status of a given server pool under the control of this Oracle VM Manager, and view and acknowledge *events* registered for both the server pool and its Oracle VM Server members.

2.6.1 Command Line Options and Parameters of `ovm_poolcontrol`

The `ovm_poolcontrol` utility has the following command line options:

- **-u** user name of an Oracle VM Manager admin user (required)
- **-p** password corresponding with the admin user name (required)
- **-h** host name of the server running Oracle VM Manager (required)
- **-X** use SSL to connect to Oracle VM Manager
- **-s** server pool name
- **-I** server pool UUID (which is a series of 16 hexadecimal numbers separated by colons)
- **-S** Oracle VM Server name (use together with `-c addserver/removeserver`)
- **-A** acknowledge *events* (use together with `-c events`)

[all, pool]: acknowledge *events* for both server pool and servers, or for the server pool only

- **-c** console command (required)

[commands: `list`, `addserver`, `removeserver`, `events`, `status`, `info`]



Note

With the `addserver` and `removeserver` commands, only a very basic add/remove of the server is performed; no assigning or unassigning of networks and repositories occurs.

2.6.2 Examples of `ovm_poolcontrol` Usage

Below are examples of `ovm_poolcontrol` commands and their respective output.

- List all server pools and their status:

```
# ./ovm_poolcontrol -u admin -p password -h localhost -c list
Oracle VM Pool Control utility 0.5.2.
Connected.
Command : list
Repository : 'MyServerPool1'
  ID       : '0004fb0000020000d2ee5d9394b371c5'
  Status   : 'Online'

Repository : 'MyServerPool2'
  ID       : '0004fb0000020000183c51866b2fa4a1'
  Status   : 'Online'

Exit...
```

- Display detailed information of a server pool:

```
# ./ovm_poolcontrol -u admin -p password -h localhost -s MyServerPool1 -c info
```

```

Oracle VM Pool Control utility 0.5.2.
Connected.
Command : info
Server Pool : MyServerPool1
  Status      : Online
  ID          : 0004fb0000020000d2ee5d9394b371c5
  Virtual IP  : 10.172.77.172
  Clustered   : true
  Master Server : MyServer1
  KeyMap      : en-us
  Load Balancer : Off
  Pool Memory  : 57342
  Available Memory: 54631
  Total Threads : 8
  CPU Utilization : 23%
  Servers in this server pool :
    Name      : 'MyServer1'
    Status    : 'Running'
    Name      : 'MyServer2'
    Status    : 'Running'
    Name      : 'MyServer3'
    Status    : 'Running'
Exit...

```

- Acknowledge server pool events :

```

# ./ovm_poolcontrol -u admin -p password -h localhost -s MyServerPool1 -c events -A pool
Oracle VM Pool Control utility 0.5.2.
Connected.
Command : events
Server Pool Events for pool 'MyServerPool'
Exit...

```

2.7 Using Oracle VM Repository Control

The `ovm_repocontrol` utility allows the administrator to quickly retrieve information about status and contents of a given storage repository, and send a refresh command to update the storage repository contents listed.

2.7.1 Command Line Options and Parameters of `ovm_repocontrol`

The `ovm_repocontrol` utility has the following command line options:

- **-u** user name of an Oracle VM Manager admin user (required)
- **-p** password corresponding with the admin user name (required)
- **-h** host name of the server running Oracle VM Manager (required)
- **-X** use SSL to connect to Oracle VM Manager
- **-r** storage repository name
- **-I** storage repository UUID
- **-s** name of an Oracle VM Server with access to the storage under the repository (use together with **-c** create)
- **-i** UUID of a LUN where a storage repository should be created (use together with **-c** create)
- **-c** console command (required)

[**commands:** `list`, `status`, `info`, `refresh`, `fixrepo`, `create`]

**Note**

The `refresh` command initiates a storage repository refresh operation through Oracle VM Manager.

The `list` command does not need a storage repository name or UUID argument. It lists every storage repository registered with the system.

The `fixrepo` command removes partial configuration information in case the creation of a storage repository on a physical disk has failed. This allows you to redo the creation operation for the storage repository.

2.7.2 Examples of ovm_repocontrol Usage

Below are examples of `ovm_servercontrol` commands and their respective output.

- List all registered storage repositories:

```
# ./ovm_repocontrol -u admin -p password -h localhost -c list
Oracle VM Repository Control utility 0.5.2.
Connected.
Command : list
Repository : 'MyIscsiRepo1'
  ID       : '0004fb0000030000b0272c74e714ab12'
  Status   : 'Online'

Repository : 'MyIscsiRepo2'
  ID       : '0004fb0000030000a1d2474a7325ab25'
  Status   : 'Online'

Repository : 'MyNfsRepo'
  ID       : '0004fb0000030000a3ff6bbfe68511a6'
  Status   : 'Online'

Exit...
```

- Create a new storage repository on a LUN:

```
# ./ovm_repocontrol -u admin -p password -h localhost -c create -r MyIscsiRepo3 -s MyServer4 \
-i 3600144f057ef8a0000004fb3194b0001
Oracle VM Repository Control utility 0.5.2.
Connected.
Command : create
Found LUN with id '3600144f057ef8a0000004fb3194b0001'
Creating filesystem...
Creating repository... Please wait...
Create repository completed.
Exit...
```

- Check the status of a storage repository:

```
# ./ovm_repocontrol -u admin -p password -h localhost -r MyIscsiRepo2 -c status
Oracle VM Repository Control utility 0.5.2.
Connected.
Command : status
Repository : 'MyIscsiRepo2'
  Status   : 'Online'

Exit...
```

- Display detailed information of a storage repository:

```
# ./ovm_repocontrol -u admin -p password -h localhost -r MyIscsiRepo1 -c info
Oracle VM Repository Control utility 0.5.2.
Connected.
```



```

Command : info
Repository : 'MyIscsiRepol'
  Total Space      : 7500000 MB
  Available Space  : 4146146 MB
  Assigned Servers :
    Server : MyServer6
    Server : MyServer7
Assemblies :
Virtual CDroms :
  CDrom : ubuntu-11.10-desktop-i386.iso
    uuid : 0004fb0000150000d8de2efc475b4ea5.iso
    VM list : (MyUbuntu1110)
  CDrom : EnterpriseLinux-R6U1-Server-x86_64.iso
    uuid : 0004fb0000150000da320985ba0c2f9b.iso
    VM list : (MyOL6)
Virtual Disks :
  Virtual Disk : MyOL6_bootdisk
    uuid : 0004fb000012000086bf184d2996f0cf.img
    VM list : (MyOL6)
  Virtual Disk : MyOL6_datadisk
    uuid : 0004fb00001200007d18ce0a22e16658.img
    VM list : (MyOL6)
  Virtual Disk : MySolaris11_bootdisk
    uuid : 0004fb00001200008a27187d18c6fe0a.img
    VM list : (MySolaris11)
  Virtual Disk : MyUbuntu1110_bootdisk
    uuid : 0004fb000012000096f0ce0a27d18c6f.img
    VM list : (MyUbuntu1110)
Virtual Machine Configs :
  Config : 0004fb0000220000b1fc2e486c1c1b7a
    VM : (MySolaris11)
  Config : 0004fb0000220000b68857e77465eafc
    VM : (MyOL6)
  Config : 0004fb0000220000eb037b82dee6b10b
    VM : (MyUbuntu1110)
Exit...

```

- Refresh the contents of a storage repository:

```

# ./ovm_repocontrol -u admin -p password -h localhost -r MyIscsiRepol -c refresh
Oracle VM Repository Control utility 0.5.2.
Connected.
Command : refresh
refreshing 3300000006160a212
Lun Refresh completed...
Exit...

```

2.8 Using Oracle VM Hostd For Metrics Messaging

The Oracle VM Utilities include tools that ensure that SAP applications are fully supported and certified to run on Oracle Linux guests running within a Oracle VM environment. These tools facilitate the communication of host-based statistics about the Oracle VM Server hosting a particular virtual machine, via the VM messaging facility provided by Oracle VM Manager, to the virtual machine itself.

Important

For this functionality to succeed, the guest virtual machine should be running Oracle Linux with the Oracle VM Guest Additions installed. See [Chapter 3, Using the Oracle VM Guest Additions](#) for more information.



Note

These are the only tools within the Oracle VM Utilities that are officially supported by Oracle.

These tools include `ovm_vmhostd` and the `vm-dump-metrics` script available in Patch 16269699 and above at:

<http://www.oracle.com/technetwork/server-storage/vm/downloads/index.html>

2.8.1 Command Line Options

The `ovm_vmhostd` has the following command line options:

- **-u** user name of an Oracle VM Manager admin user (required)
- **-p** password corresponding with the admin user name (required)
- **-h** host name of the server running Oracle VM Manager (required)
- **-X** use SSL to connect to Oracle VM Manager
- **-v** Virtual Machine name
- **-U** Virtual Machine UUID (optional alternative)

2.8.2 Example Usage

To run `ovm_vmhostd` on Oracle VM Manager you might run the following command:

```
# ./ovm_vmhostd -u admin -p MyPassword -h localhost -v myvm
```

From the example usage of the command, details of the underlying Oracle VM Server on which *myvm* is running are sent as messages via the Oracle VM Manager to the *myvm* virtual machine.

The `ovm_vmhostd` utility runs with daemon-like behaviour. It initially checks that the virtual machine is running, and then sends the server metrics as a message. It sleeps for 60 seconds and then sends a message updating the values for the original message. This behaviour continues until the process is killed. Typical output follows:

```
# ./ovm_vmhostd -u admin -p MyPassword -h localhost -v myvm
Oracle VM Hostd 0.6.3.
Connected.
Manager version : 3.2.1.516
VM : 'pvml' has status : Running.
VM: 'pvml' is online, sending metrics.
.Sleeping 60 seconds.
.Sleeping 60 seconds.
.Sleeping 60 seconds.
```

2.8.3 Using `vm-dump-metrics` To Retrieve Host Statistics Within a VM

The `vm-dump-metrics` script can be copied to the guest virtual machine where the message is received. This script is a simple shell script that can output the Oracle VM Server host information in XML format. This XML can be consumed by a SAP application running within the virtual machine.

By default, the `vm-dump-metrics` script outputs data to STDOUT, but can be changed to output data to file by editing the `OUTFILE` variable at the beginning of the script.

The `vm-dump-metrics` script queries `ovmd` to obtain the message with the key *vmhost*. If no message is obtained, it exits with status 1. If a message is found, the script parses this message to populate the resulting XML. Typical XML output generated by the `vm-dump-metrics` script follows:

```
<metrics>
  <metric type='real64' context='host'>
    <name>TotalCPUTime</name>
    <value>2694.3596</value>
```

```
</metric>
<metric type='uint64' context='host'>
  <name>PagedOutMemory</name>
  <value>0</value>
</metric>
<metric type='uint64' context='host'>
  <name>PagedInMemory</name>
  <value>0</value>
</metric>
<metric type='uint64' context='host'>
  <name>UsedVirtualMemory</name>
  <value>6747</value>
</metric>
<metric type='uint64' context='host'>
  <name>FreeVirtualMemory</name>
  <value>9817</value>
</metric>
<metric type='uint64' context='host'>
  <name>FreePhysicalMemory</name>
  <value>9817</value>
</metric>
<metric type='uint64' context='host'>
  <name>MemoryAllocatedToVirtualServers</name>
  <value>6747</value>
</metric>
<metric type='uint32' context='host'>
  <name>NumberOfPhysicalCPUs</name>
  <value>4</value>
</metric>
<metric type='string' context='host'>
  <name>HostSystemInfo</name>
  <value>ovm3</value>
</metric>
<metric type='string' context='host'>
  <name>VirtProductInfo</name>
  <value>Oracle VM 3</value>
</metric>
<metric type='string' context='host'>
  <name>VirtualizationVendor</name>
  <value>Oracle Corporation</value>
</metric>
<metric type='uint64' context='host'>
  <name>Time</name>
  <value>1360606566774</value>
</metric>
<metric type='string' context='host'>
  <name>HostName</name>
  <value>ovm3</value>
</metric>
<metric type='uint64' context='vm' id='0' uuid='0004fb00-0006-0000-d72b-647e20a85939'>
  <name>PhysicalMemoryAllocatedToVirtualSystem</name>
  <value>1024</value>
</metric>
<metric type='uint64' context='vm' id='0' uuid='0004fb00-0006-0000-d72b-647e20a85939'>
  <name>ResourceMemoryLimit</name>
  <value>1024</value>
</metric>
<metric type='uint32' context='vm' id='0' uuid='0004fb00-0006-0000-d72b-647e20a85939'>
  <name>ResourceProcessorLimit</name>
  <value>1</value>
</metric>
<metric type='real64' context='vm' id='0' uuid='0004fb00-0006-0000-d72b-647e20a85939'>
  <name>TotalCPUTime</name>
  <value>2694.3596</value>
</metric>
</metrics>
```

If you do not want to use the `vm-dump-metrics` script on your virtual machine, you can query `ovmd` directly and parse the output yourself:

```
# ovmd -g vmhost
com.sap.host.VirtualizationVendor=Oracle Corporation;com.sap.host.VirtProductInfo=Oracle VM 3;
com.sap.host.PagedInMemory=0;com.sap.host.PagedOutMemory=0;com.sap.host.PageRates=0;
com.sap.vm.uuid=0004fb0000060000d72b647e20a85939;com.sap.host.HostName=ovm3;
com.sap.host.HostSystemInfo=ovm3;com.sap.host.NumberOfPhysicalCPUs=4;com.sap.host.NumCPUs=4;
com.sap.host.TotalPhyMem=16383;com.sap.host.UsedVirtualMemory=6747;
com.sap.host.MemoryAllocatedToVirtualServers=6747;com.sap.host.FreeVirtualMemory=9817;
com.sap.host.FreePhysicalMemory=9817;com.sap.host.TotalCPUTime=381175.97;
com.sap.host.Time=1360606887997;com.sap.vm.PhysicalMemoryAllocatedToVirtualSystem=1024;
com.sap.vm.ResourceMemoryLimit=1024;com.sap.vm.TotalCPUTime=2696.2214;
com.sap.vm.ResourceProcessorLimit=1;
```

Chapter 3 Using the Oracle VM Guest Additions

Table of Contents

3.1 Features of the Oracle VM Guest Additions	23
3.2 Installing Oracle VM Guest Additions	23
3.3 Upgrading the Oracle VM Guest Additions	24
3.4 The Oracle VM Guest Additions Daemon	25
3.4.1 Configuring the Oracle VM Guest Additions Daemon to Run as a Service	26
3.4.2 Using the Oracle VM Guest Additions Daemon to Enable First-Boot Configuration	26
3.4.3 Using the Messaging Channel	26
3.5 Displaying the Guest IP Address	27
3.6 The Oracle VM Template Configuration Script and Modules	27
3.6.1 Overview of the Template Configuration Script	28
3.6.2 Enabling and Disabling Configuration Modules	29
3.6.3 Key-Value Pairs Used By Available Configuration Modules	29
3.6.4 Triggering Configuration Changes	30
3.6.5 Developing Oracle VM Template Configuration Modules	31

Oracle VM Guest Additions is a set of packages that can be installed on the *guest* operating system of a virtual machine running in the Oracle VM environment. These packages provide the tools to allow bi-directional communication directly between the *Oracle VM Manager*; and the operating system running within the virtual machine. This is a powerful tool that provides administrators fine-grained control over the configuration and behavior of components running within the virtual machine directly from Oracle VM Manager.

This chapter gives you more detail on the installation, configuration and features of the Oracle VM Guest Additions.

3.1 Features of the Oracle VM Guest Additions

Features of the Oracle VM Guest Additions include:

- Improved information about virtual machines within Oracle VM Manager, such as reporting on IP addressing.
- Ability to use the template configuration facility to automatically configure virtual machines as they are first started
- Ability to send messages directly to a virtual machine from Oracle VM Manager to trigger programmed events
- Ability to query a virtual machine to obtain information pertaining to previous messages
- Ability to interact with the Oracle VM Utilities `ovm_vmmessage` command

The Oracle VM Guest Additions allows direct integration between guest software and the virtualization layer, to assist in orchestration and automation of complex, multi-VM deployments.

3.2 Installing Oracle VM Guest Additions

For the Oracle VM templates without Oracle VM Guest Additions installed, the following software packages can be downloaded from [Oracle Unbreakable Linux Network \(ULN\)](#) by choosing the proper channel to

download and install into the existing Oracle Linux 5 or 6 guest with Oracle Unbreakable Enterprise Kernel (UEK).

```
kmod-ovmapi-uek
libovmapi
libovmapi-devel
ovmd
python-simplejson
xenstoreprovider
ovm-template-config
ovm-template-config-authentication
ovm-template-config-datetime
ovm-template-config-firewall
ovm-template-config-network
ovm-template-config-selinux
ovm-template-config-ssh
ovm-template-config-system
ovm-template-config-user
```

These packages can be downloaded from the following [ULN](#) channels:

- Oracle Linux 6 Add ons (x86_64)
- Oracle Linux 6 Add ons (i386)
- Enterprise Linux 5 Add ons (x86_64)
- Enterprise Linux 5 Add ons (i386)

In addition, the packages are available from Oracle's [Public YUM repository](#).

To install these packages, make sure that your virtual machine has public internet access and is connected to the appropriate ULN channel or YUM repository. Then use the following command syntax in your Oracle Linux guest, separating the package names by spaces:

```
# yum install libovmapi xenstoreprovider ovmd python-simplejson xenstoreprovider
```



Caution

When manually installing the downloaded packages, make sure that the `kmod-ovmapi-uek` version matches the UEK version of the virtual machine:

- UEK 2.36.32-100 requires `kmod-ovmapi-uek-1.0.0-27.100`
- UEK 2.36.32-200 requires `kmod-ovmapi-uek-1.0.0-27.200`
- UEK 2.36.32-300 requires `kmod-ovmapi-uek-1.0.0-27.300`

3.3 Upgrading the Oracle VM Guest Additions

For the Oracle VM templates that already have Guest Additions installed, you may need to ensure that certain packages are updated in order for Guest Additions to function correctly. Using the [Oracle Unbreakable Linux Network \(ULN\)](#) Yum repository you can run the following command to update the Guest Additions packages:

```
# yum update ovmd libovmapi xenstoreprovider \
  ovm-template-config \
  ovm-template-config-authentication \
  ovm-template-config-datetime \
  ovm-template-config-firewall \
  ovm-template-config-network \
  ovm-template-config-selinux \
```

```
ovm-template-config-ssh \
ovm-template-config-system \
ovm-template-config-user
```

If you are using a kernel version lower than UEK 2.6.39-300, and you want to continue to use the current kernel, you must also run the following command to update the `ovmapi.ko` module:

```
# yum update kmod-ovmapi-uek
```

Alternatively, you can update your kernel to the latest UEK version by running the following command:

```
# yum update kernel-uek
```



Note

If you are using Oracle Linux 5, you need to enable `ol5_UEK_latest` within `/etc/yum.repos.d/ULN-Base.repo`, before you attempt to update your kernel version. For example, the file should contain the following lines:

```
[ol5_UEK_latest]
name=Latest Unbreakable Enterprise Kernel for Oracle Linux $releasever ($basearch)
baseurl=http://public-yum.oracle.com/repo/OracleLinux/OL5/UEK/latest/$basearch/
gpgkey=http://public-yum.oracle.com/RPM-GPG-KEY-oracle-el5
gpgcheck=1
enabled=1
```

3.4 The Oracle VM Guest Additions Daemon

The Oracle VM Guest Additions daemon, `ovmd`, facilitates a bi-directional messaging channel between Oracle VM Manager and the guest. It allows first-boot installation configuration, and is capable of sending and receiving messages consisting of key-value pairs.

It is possible to send messages via Oracle VM Manager to the Oracle VM Guest Additions daemon running on any guest, using the `ovm_vmmmessage` utility that can be installed on any linux system with access to Oracle VM Manager or on the Oracle VM Manager host itself. See [Section 2.5, “Using Oracle VM Virtual Machine Messaging”](#) for more information on using this tool.

It is also possible to use the messaging facility included directly within the Oracle VM Manager user interface to send messages directly to the Oracle VM Guest Additions daemon running on any guest. See [Sending Messages to Virtual Machines](#) in the *Oracle VM User's Guide* for more information.

Used in conjunction with the `ovm-template-config` script, `ovmd` can be used to remotely configure system and application configuration parameters within a virtual machine as it boots. See [Section 3.6, “The Oracle VM Template Configuration Script and Modules”](#) for more information on this facility.

Oracle VM Manager makes use of `ovmd` in order to obtain IP addressing information from the guest to include in the the Oracle VM Manager user interface when displaying detailed virtual machine information. See [Section 3.5, “Displaying the Guest IP Address”](#).

You can run `ovmd` directly from the command line to perform actions outside of `ovmd`'s function as a daemon or system service. Running `ovmd` using the `--help` parameter provides you with a breakdown of the options supported when run directly from the command line:

```
# ovmd --help
usage: ovmd [options]

options:
-p, --set-param=PARAM    Set param in format: key=value.
-g, --get-param=KEY       Get param value.
-r, --delete-param=KEY    Delete param by key name.
-x, --delete-params       Delete all params.
```

```
-l, --list-params      List all params.
-e, --event=EVENT      Inject event.
-s, --script=SCRIPT    Run script.
-d, --debug=LEVEL      Set debug level: 0(DEBUG_OFF), 1(DEBUG_STDERR) or 2(DEBUG_SYSLOG).
                        Default: 2.
-f, --pid-file=FILENAME Set the pathname of the process ID (PID) file.
-t, --time-period=secs Set the period for daemon mode(default=10 seconds).
-v, --version          Show version number and exit.
-h, --help            Show this help information.
```

3.4.1 Configuring the Oracle VM Guest Additions Daemon to Run as a Service

To enable `ovmd` to run as a service on Oracle Linux, run the `chkconfig` command as root:

```
# chkconfig ovmd on
```

To start the `ovmd` service, run the following as root:

```
# /etc/init.d/ovmd start
```

Once configured as a service, `ovmd` listens for message requests sent via the Oracle VM Manager.

3.4.2 Using the Oracle VM Guest Additions Daemon to Enable First-Boot Configuration

If you are configuring a virtual machine to act as a template or if you intend to clone it, you may want to set it up so that at next boot it behaves as if it is booting for the first time, thereby prompting for configuration input either by the VM API or on the virtual machine console. This can be achieved by running the following commands as root within the virtual machine:

```
# ovmd -s cleanup
# service ovmd enable-initial-config
# shutdown -h now
```

On next boot, the virtual machine will act as if it is performing a first-time boot. If you have configured `ovmd` to run as a service, you are able to configure it remotely using the `messaging` facility and the `ovm-template-config` script.

See [Section 3.4.3, “Using the Messaging Channel”](#) for more information on using the messaging channel. Also see [Section 3.6, “The Oracle VM Template Configuration Script and Modules”](#) for information about the `ovm-template-config` script.

3.4.3 Using the Messaging Channel

This section provides an example of a message exchange between Oracle VM Manager and a running Oracle Linux virtual machine with Oracle VM Guest Additions installed. The `ovm_vmmessage` tool that is included with Oracle VM Utilities is used to communicate with the Oracle Linux guest. More information about the messaging utility can be found in [Section 2.5, “Using Oracle VM Virtual Machine Messaging”](#).

Sending a message from the guest to Oracle VM Manager.

Using `ovmd`, you send information from within the virtual machine to your Oracle VM Manager using the following syntax:

```
# ovmd -p key1=value1
```

The message appears in the Oracle VM Manager user interface, as a *Virtual Machine API Incoming Message* event for the virtual machine in question. When you expand the event, the description shows the key-value pair and the date and time when the information exchange took place.

The message from the guest can also be retrieved via the Oracle VM Manager command line utility `ovm_vmmmessage`. To do so, you query the key and the value is returned in the response:

```
# ./ovm_vmmmessage -u admin -p password -h localhost -v MyVM02 -q key1
Oracle VM VM Message utility 0.5.2.
Connected.
VM : 'MyVM02' has status : Running.
Querying for key 'key1'.
Query successful.
Query for Key : 'key1' returned value 'value1'.
Key set 7 minutes ago.
```

Sending a message from Oracle VM Manager to a virtual machine.

Using `ovm_vmmmessage`, you send information to a virtual machine using the following syntax:

```
# ./ovm_vmmmessage -u admin -p password -h localhost -v MyVM02 -k key2 -V value2
Oracle VM VM Message utility 0.5.2.
Connected.
VM : 'MyVM02' has status : Running.
Sending message.
Message sent successfully.
```

Using `ovmd` from within the guest, you can retrieve the message sent from Oracle VM Manager using the following syntax:

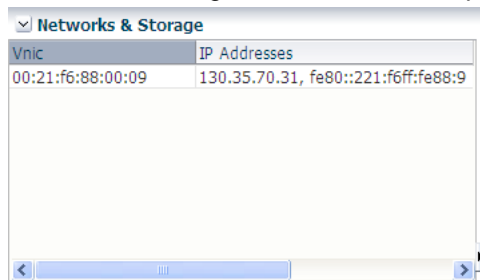
```
# ovmd --list
{"key1": "value1"}
{"key2": "value2"}
```

The `ovmd --list` command retrieves all messages, both sent and received. You can identify the specific message you are looking for by its key. To remove obsolete messages, use the following syntax:

```
# ovmd -r key1
# ovmd --list
{"key2": "value2"}
```

3.5 Displaying the Guest IP Address

When the Oracle VM Guest Additions are installed, the virtual machine IP address becomes visible in the Oracle VM Manager user interface, as part of the detailed virtual machine information.



3.6 The Oracle VM Template Configuration Script and Modules

The Oracle VM Guest Additions include a set of packages that can help with the automatic configuration of virtual machines as they are created from a template and booted for the first time. The master package for this facility is known as `ovm-template-config`. The Oracle VM template configuration script can be used to configure a virtual machine remotely using the Oracle VM *messaging* facility via `ovmd`.

3.6.1 Overview of the Template Configuration Script

The Oracle VM Template Configuration Script, `ovm-template-config` works in conjunction with a set of modular configuration scripts that function in a manner very similar to the standard Linux System V, `init.d` and `chkconfig`, script model. Control over how configuration modules are run is handled within the `/etc/template.d` directory on the guest. The configuration module scripts are stored within `/etc/template.d/scripts`.

The `ovm-template-config` script is the master script that is used to control all enabled modules. Running `ovm-template-config` with the `--help` parameter provides a usage breakdown:

```
ovm-template-config --help
usage: ovm-template-config [option] target

Targets:
    configure, unconfigure, reconfigure, cleanup, suspend, resume, migrate, shutdown

Examples:
    ovm-template-config --enumerate configure
    ovm-template-config --enumerate --script network configure
    ovm-template-config --stdin configure
    ovm-template-config --stdin --script network configure
    ovm-template-config --console-input configure
    ovm-template-config --ovf-transport-iso configure
    ovm-template-config --input >infd< --output >outfd< configure

options:
    --version                show program's version number and exit
    -h, --help               show this help message and exit
    -e, --enumerate          enumerate parameters for target
    --human-readable          print in human readable format when enumerate
                             parameters
    -i INPUT, --input=INPUT  input parameters from this file descriptor
    -o OUTPUT, --output=OUTPUT
                             output parameters to this file descriptor
    --stdin                  build parameters from stdin
    --console-input          build parameters from console input
    --ovf-transport-iso      build parameters from OVF transport ISO
    -s SCRIPT, --script=SCRIPT
                             specify script
    --logfile=LOGFILE        specify log file
    --loglevel=LOGLEVEL      specify log level
```

For remote configuration, `ovm-template-config` is used in conjunction with `ovmd` to capture configuration parameters that have been sent to the guest using the Oracle VM messaging facility. When this is the case, `ovm-template-config` targets are presented to `ovmd` as `--script` parameters:

```
# ovmd -s cleanup
# ovmd -s configure
```

Important

When performing remote configuration, using `ovmd` to process messages containing configuration keys, the *authentication* module must be enabled. Processing of messages can only be completed if the final message contains the root user password, which is used by the *authentication* module to obtain the privileges required in order to perform other configuration processing. See [Section 3.6.2, “Enabling and Disabling Configuration Modules”](#) for more information on this module.

See [Section 3.6.4, “Triggering Configuration Changes”](#) for more information on calling this script directly.

3.6.2 Enabling and Disabling Configuration Modules

When a module is enabled, symlinks to the module script are made to other subdirectories within `/etc/template.d` based on the type of target the module provides, in much the same way that the System V init process works. When a module gets added, the header of the module script is read to verify the name, priority and targets and then a symlink is made to the corresponding subdirectories under `/etc/template.d`.

Enabling and disabling targets for any module is handled using the `ovm-chkconfig`. Usage of this command is outlined using the `--help` parameter:

```
# ovm-chkconfig --help
usage: ovm-chkconfig [option] ...

examples:
  ovm-chkconfig --list [name]
  ovm-chkconfig --add name
  ovm-chkconfig --del name
  ovm-chkconfig --target <targets> name <on|off>

options:
  --version      show program's version number and exit
  -h, --help     show this help message and exit
  --list         list script status
  --add          add new script
  --del          delete script
  --target=TARGET specify the targets, separated by comma, e.g.:
                  "configure,unconfigure"
```

Use the `ovm-chkconfig` command with the `--list` parameter to see which modules are available and their target runtime status:

```
# ovm-chkconfig --list
```

name	configure	unconfigure	reconfigure	cleanup	suspend	resume	migrate	shutdown
authentication	on:90	off	off	off	off	off	off	off
datetime	on:50	off	off	off	off	off	off	off
firewall	on:41	off	off	off	off	off	off	off
network	off	off	off	off	off	off	off	off
selinux	off	off	off	off	off	off	off	off
ssh	off	off	off	off	off	off	off	off
system	off	off	off	off	off	off	off	off
user	off	off	off	off	off	off	off	off

To enable all of the targets supported by a module, use the `--add` parameter:

```
# ovm-chkconfig --add authentication
```

To disable all of the targets supported by a module, use the `--del` parameter:

```
# ovm-chkconfig --del datetime
```

It is also possible to enable or disable particular targets for a module, for instance:

```
ovm-chkconfig --target=cleanup user off
```

3.6.3 Key-Value Pairs Used By Available Configuration Modules

To obtain a full listing of all of the key pairs that are used to trigger configuration changes through the `ovm-template-config` configuration modules, run the following command on the guest system where `ovm-template-config` is installed:

```
# ovm-template-config --human-readable --enumerate configure
```

The output from this command is printed as a Python data structure, that is easy to parse and understand. Content can be limited to the information specific to a configuration module by using the `--script` parameter as presented below:

```
# ovm-template-config --human-readable --enumerate configure --script datetime
[('50',
 'datetime',
 [{u'description': u'System date and time in format year-month-day-hour-minute-second, e.g.,
    "2011-4-7-9-2-42".',
   u'hidden': True,
   u'key': u'com.oracle.linux.datetime.datetime'},
 {u'description': u'System time zone, e.g., "America/New_York".',
   u'hidden': True,
   u'key': u'com.oracle.linux.datetime.timezone'},
 {u'description': u'Whether to keep hardware clock in UTC: True or False.',
   u'hidden': True,
   u'key': u'com.oracle.linux.datetime.utc'},
 {u'description': u'Whether to enable NTP service: True or False.',
   u'hidden': True,
   u'key': u'com.oracle.linux.datetime.ntp'},
 {u'description': u'NTP servers separated by comma, e.g.,
    "time.example.com,0.example.pool.ntp.org".',
   u'hidden': True,
   u'key': u'com.oracle.linux.datetime.ntp-servers'},
 {u'description': u'Whether to enable NTP local time source: True or False.',
   u'hidden': True,
   u'key': u'com.oracle.linux.datetime.ntp-local-time-source'}]]]
```

From the output, it becomes clear as to which configuration modules are triggered at which runlevel and what keys and values they accept. Note that all key names are structured so that they are prefixed with *com.oracle*, to avoid conflicts with any custom modules that you may intend to develop for your own purposes.

Key value pairs are actually passed to `ovm-template-config` in JSON format:

```
{"com.oracle.linux.datetime.ntp": "True"}
{"com.oracle.linux.datetime.ntp-servers": "0.pool.ntp.org,1.pool.ntp.org,2.pool.ntp.org"}
{"com.oracle.linux.root-password": "mysecret"}
```

3.6.4 Triggering Configuration Changes

There are a variety of approaches that can be used to trigger a configuration change using `ovm-template-config`. Most commonly, this is done by setting the `ovmd` service to run using in *enable-initial-config* mode. This causes the virtual machine to wait to be provided with configuration parameters either on via the console, or via the `ovmd` messaging facility, after the next boot. This is the usual approach when configuring a virtual machine to act as a template.

To manually force `ovmd` to pass messages to the `ovm-template-config` script, simply run `ovmd` with the `--script` parameter set to point to one of the `ovm-template-config` targets:

```
# ovmd --list
{"com.oracle.linux.datetime.ntp": "True"}
{"com.oracle.linux.datetime.ntp-servers": "0.pool.ntp.org,1.pool.ntp.org,2.pool.ntp.org"}
{"com.oracle.linux.root-password": "mysecret"}
# ovmd -s configure
```



Note

To perform an action using messaging parameters and `ovm-template-config`, the *authentication* module must be enabled and your final message should include and authentication request in the form of a key-value message:

```
["com.oracle.linux.root-password": "mysecret"]}
```

When running `ovmd` like this, `ovmd` prepares two pipes (*infd* and *outfd*) and calls the `ovm-template-config` script transparently in the background in the following way:

```
# ovm-template-config --input <infd> --output <outfd> configure
```

During testing, it may be useful to simply pass configuration information directly to the script from STDIN on the command line. This can be achieved by calling the script directly with the `--stdin` parameter:

```
# ovm-template-config --stdin configure <<EOF
> {"com.oracle.linux.selinux.mode": "disabled"}
> {"com.oracle.linux.root-password": "ovsroot"}
> EOF
```

Configuration can also be achieved directly from the console by running the script using the `--console-input`. Doing this will prompt you for values for each of the keys that need to be defined for any enabled modules:

```
# ovm-template-config --console-input configure
```

3.6.5 Developing Oracle VM Template Configuration Modules

The provided module scripts are developed in Python. Theoretically, it is possible to develop module scripts in a different language, as long as the input, output and argument handling remains the same. The example provided in this section makes use of the Python programming language.

Each module script consists of 2 main parts:

1. the script header which contains information like script name, targets, priorities and description.
2. the actual script, which handles a small set of parameters.

For examples of functional module scripts, refer to the existing modules in the `/etc/template.d/scripts` directory.

Module Script Header

Module script headers require a very specific comment block in order for `ovm-chkconfig` to handle enabling and disabling your script functionality. The format for the script header is as follows:

```
### BEGIN PLUGIN INFO
# name: [script name]
# [target]: [priority]
# [target]: [priority]
# description: a description that can
#   cross multiple lines.
### END PLUGIN INFO
```

When developing your own module script, you must include a header following the exact same format. Provide your own script name, which will be used when calling `ovm-chkconfig`, the targets that your script will support, and the priority for your script. The priority will specify in what order the script gets executed. You do not have to implement all targets. If you have a configure target but no cleanup target, this is still acceptable. The configure target gets called when a first boot/initial start of the virtual machine happens. The cleanup target happens when you manually initiate a cleanup in your virtual machine or when you want to restore the virtual machine to its original state. An example of the network module script header is provided below:

```
### BEGIN PLUGIN INFO
```

```
# name: network
# configure: 50
# cleanup: 50
# description: Script to configure template network.
### END PLUGIN INFO
```

Module Script Body

The main requirement for the module script body is that it accepts at least one target parameter. Target parameters that might get presented by the `ovm-template-configure` script include:

- configure
- unconfigure
- reconfigure
- cleanup
- suspend
- resume
- migrate
- shutdown

Your script can handle any other arguments that you require. There is one optional parameter which is useful to implement and this is `-e` or `--enumerate`. `ovm-template-config` uses this to be able to enumerate or list the parameters for a target supported by your script.

A very basic template to use for your script body follows:

```
try:
    import json
except ImportError:
    import simplejson as json
from templateconfig.cli import main

def do_enumerate(target):
    param = []
    if target == 'configure':
        param += []
    elif target == 'cleanup':
        param += []
    return json.dumps(param)

def do_configure(param):
    param = json.loads(param)
    return json.dumps(param)

def do_cleanup(param):
    param = json.loads(param)
    return json.dumps(param)

if __name__ == '__main__':
    main(do_enumerate, {'configure': do_configure, 'cleanup': do_cleanup})
```

This script supports the configure and cleanup targets.

You can fill out the script with your own code. For instance, for the `do_enumerate` function, you would populate the parameters that are supported for each target in the script. An example from the firewall module is presented below:

```
def do_enumerate(target):
    param = []
    if target == 'configure':
        param += [{'key': 'com.oracle.linux.network.firewall',
                    'description': 'Whether to enable network firewall: True or False.',
                    'hidden': True}]
    return json.dumps(param)
```

Each target function begins by reading the JSON parameters passed to the script, using the `param = json.loads(param)` statement. From this point, code can be written to perform actions based on the values of the keys that the script expects to receive. Once again, the example provided below is from the firewall module:

```
def do_configure(param):
    param = json.loads(param)
    firewall = param.get('com.oracle.linux.network.firewall')
    if firewall == 'True':
        shell_cmd('service iptables start')
        shell_cmd('service ip6tables start')
        shell_cmd('chkconfig --level 2345 iptables on')
        shell_cmd('chkconfig --level 2345 ip6tables on')
    elif firewall == 'False':
        shell_cmd('service iptables stop')
        shell_cmd('service ip6tables stop')
        shell_cmd('chkconfig --level 2345 iptables off')
        shell_cmd('chkconfig --level 2345 ip6tables off')
    return json.dumps(param)
```

Module Script Packaging

Once you have written one or more configuration module scripts, you may want to package them as RPMs that can be deployed on other systems. In order to install and configure template configure scripts, they have to be packaged in an RPM, with a specific naming convention. Package the script as `ovm-template-config-[scriptname]`. Ideally in the post install of the RPM you should add the script automatically by executing `# /usr/sbin/ovm-chkconfig --add [scriptname]`. When de-installing a script/RPM, remove it at un-install time using `# /usr/sbin/ovm-chkconfig --del [scriptname]`. This is illustrated in the following example of an RPM spec file that can be used:

```
Name: ovm-template-config-example
Version: 3.0
Release: 1%{?dist}
Summary: Oracle VM template example configuration script.
Group: Applications/System
License: GPL
URL: http://www.oracle.com/virtualization
Source0: %{name}-%{version}.tar.gz
BuildRoot: %{mktemp -ud %{_tmppath}}/%{name}-%{version}-%{release}-XXXXXX
BuildArch: noarch
Requires: ovm-template-config

%description
Oracle VM template example configuration script.

%prep
%setup -q

%install
rm -rf $RPM_BUILD_ROOT
make install DESTDIR=$RPM_BUILD_ROOT
```

```
%clean
rm -rf $RPM_BUILD_ROOT

%post
if [ $1 = 1 ]; then
    /usr/sbin/ovm-chkconfig --add example
fi

%preun
if [ $1 = 0 ]; then
    /usr/sbin/ovm-chkconfig --del example
fi

%files
%defattr(-,root,root,-)
%{_sysconfdir}/template.d/scripts/example

%changelog
* Tue Mar 22 2011 John Smith - 3.0-1
- Initial build.
```

Ensure that you edit the example spec file to reference your own script name.

In order to create RPMs, you must install rpmbuild:

```
# yum install rpm-build
```

The following example Makefile may help to automate the build process:

```
DESTDIR=
PACKAGE=ovm-template-config-example
VERSION=3.0

help:
@echo 'Commonly used make targets:'
@echo '  install    - install program'
@echo '  dist       - create a source tarball'
@echo '  rpm        - build RPM packages'
@echo '  clean      - remove files created by other targets'

dist: clean
mkdir $(PACKAGE)-$(VERSION)
tar -cSp --to-stdout --exclude .svn --exclude .hg --exclude .hgignore \
--exclude $(PACKAGE)-$(VERSION) * | tar -x -C $(PACKAGE)-$(VERSION)
tar -czSpf $(PACKAGE)-$(VERSION).tar.gz $(PACKAGE)-$(VERSION)
rm -rf $(PACKAGE)-$(VERSION)

install:
install -D example $(DESTDIR)/etc/template.d/scripts/example

rpm: dist
rpmbuild -ta $(PACKAGE)-$(VERSION).tar.gz

clean:
rm -fr $(PACKAGE)-$(VERSION)
find . -name '*.py[cd]' -exec rm -f '{}' ';'
rm -f *.tar.gz

.PHONY: dist install rpm clean
```

Remember to edit this Makefile to reference your own script.

Create a working directory, copy over your script, the spec file and the Makefile. Run:

```
# make dist
```


to create a src tarball of your code and then run:

```
# make rpm
```

This will generate an RPM in the RPMS/noarch directory within your working directory. For example:
[RPMS/noarch/ovm-template-config-test-3.0-1.el6.noarch.rpm](#).

Chapter 4 Other Oracle VM Tools

Table of Contents

4.1 Oracle VM Administrator Tool	37
4.1.1 Managing Users with Oracle VM Administrator Tool	37
4.1.2 Log Rotation Options	42
4.2 Oracle VM Diagnostic Capture	43

This chapter discusses additional tools that are available for Oracle VM, which do not form part of the Oracle VM Utilities and Oracle VM Guest Additions packages. These tools are usually bundled with Oracle VM Manager and provide facilities to quickly perform certain configuration and diagnostic tasks directly from the command line.

4.1 Oracle VM Administrator Tool

The Oracle VM Administrator Tool, which can be invoked on the command line using the `ovm_admin` command, is used to perform typical administrative actions specific to Oracle VM Manager. These actions allow you to manage users that have access to Oracle VM Manager, reconfigure Oracle VM Manager's datastore, and control log rotation. To perform any action using the Oracle VM Administrator Tool, you must use the password that is configured for the `weblogic` user.

The Oracle VM Administrator Tool is installed as part of the default Oracle VM Manager installation process. The full path to the Oracle VM Administrator Tool is: `/u01/app/oracle/ovm-manager-3/bin/ovm_admin`

The following is listed in the help page on the command line:

```
Usage: ./ovm_admin [options]

Options:
--help: Shows this message
--createuser: Create new Oracle VM Manager admin user
--deleteuser: <admin> Delete Oracle VM Manager admin user
--listusers: List Oracle VM Manager users
--modifyuser: Modify Oracle VM Manager user password
--lockusers: <tries> Max login tries before locking account. This setting is global.
--unlockuser: <admin> Unlock user account
--modifyds <SID> <host> <port> [<type>] Modify Data Store 'OVMDS'. Options of <type>:
    oracle, mysql
--listconfig: List configuration
--rotatelogsdaily: <time> Rotate Logs Daily (HH:MM)
--rotatelogsbysize: <size> Rotate Logs By Size (KB)
```



Warning

Do not use the `--modifyds` option in this command unless instructed to do so by an Oracle Support representative. This option can result in your Oracle VM Manager deployment becoming completely unusable. This option is used to migrate an existing Oracle VM Manager deployment to an alternate database backend. It does not handle the migration of the data within the database.

4.1.1 Managing Users with Oracle VM Administrator Tool

The Oracle VM Administrator Tool provides you with the ability to perform various user management functions directly from the command line. By default, the Oracle VM Manager installation process only

creates and configures a single Oracle VM Manager administrative user. While this is often sufficient for many customers, creating separate administrative user accounts may be useful for security and auditing purposes.

4.1.1.1 Creating a New Oracle VM Manager User

A new user can be created for the Oracle VM Manager application using the Oracle VM Administrator Tool by running the following command:

```
# ./ovm_admin --createuser
```

The tool returns the following output:

```
Oracle VM Manager Release 3.3.1 Admin tool

Please enter the username : [ovmuser]

Please enter the password for [ovmuser] (minimum 8 chars. with one numeric/special char.) :
Please re-enter the password :
```



Note

Your password must conform to the password requirements suggested by the Oracle VM Administrator Tool, or the creation of your user fails in the final step.

```
Please enter the password for weblogic :
```

At this point you must enter the password for the Weblogic system. If you performed a Simple Installation of Oracle VM Manager, this password is the same as your default Oracle VM Manager admin user's password.

```
Initializing WebLogic Scripting Tool (WLST) ...

Welcome to WebLogic Server Administration Scripting Shell

Type help() for help on available commands

Connecting to WebLogic server ...

Connected ...
Creating user '[ovmuser]' ...
Created user '[ovmuser]' successfully ...
Exiting...
```

4.1.1.2 Deleting an Oracle VM Manager User

Removing an Oracle VM Manager administrative user can be achieved using the Oracle VM Administrator Tool by running the following command:

```
# ./ovm_admin --deleteuser [ovmuser]
```

You are prompted for the Weblogic password. This is the password for the Weblogic system as it was set up during installation. If you performed a Simple Installation of Oracle VM Manager, this password is the same as your default Oracle VM Manager admin user's password. Typical output is presented below:

```
Oracle VM Manager Release 3.3.1 Admin tool

Please enter the password for weblogic :

Initializing WebLogic Scripting Tool (WLST) ...

Welcome to WebLogic Server Administration Scripting Shell
```

```
Type help() for help on available commands

Connecting to WebLogic server ...

Connected ...
Deleting user '[ovmuser]' ...
Deleted user '[ovmuser]' successfully ...
Exiting...
```

Important

There are some users stored within Weblogic that are critical to the healthy functioning of your Oracle VM Manager environment. Do not delete either of these accounts:

- *OracleSystemUser*
- *weblogic*

It is also generally advisable to keep the default *admin* user account, so that there is always at least one administrative account that is able to access the Oracle VM Manager application.

4.1.1.3 Changing an Oracle VM Manager User's Password

It is possible to change any Oracle VM Manager administrative user's password using the Oracle VM Administrator Tool by running the following command:

```
# ./ovm_admin --modifyuser
```

The tool returns the following output:

```
Oracle VM Manager Release 3.3.1 Admin tool

Please enter the username : [ovmuser]

Please enter the current password :
```



Note

You must be able to provide the user's current password in order to modify the user account.

If you need to reset an account due to a lost password, you should delete the user account and create a new account.

```
Please enter a new password for [ovmuser] (minimum 8 chars. with one numeric/special char.) :
Please re-enter the password :
```



Note

Your password must conform to the password requirements suggested by the Oracle VM Administrator Tool, or the creation of your user fails in the final step.

```
Please enter the password for weblogic :
```

At this point you must enter the password for the Weblogic system. If you performed a Simple Installation of Oracle VM Manager, this password is the same as your default Oracle VM Manager admin user's password.

```
Initializing WebLogic Scripting Tool (WLST) ...
```

```
Welcome to WebLogic Server Administration Scripting Shell

Type help() for help on available commands

Connecting to WebLogic server ...

Connected ...
Modifying user '[ovmuser]' ...
Modified user '[ovmuser]' successfully ...
Exiting...
```

4.1.1.4 Getting a List of Oracle VM Manager Users

You can use the Oracle VM Administrator Tool to obtain a list of users that have access to the Oracle VM Manager application by running the following command:

```
# ./ovm_admin --listusers
```

The tool prompts you for the Weblogic password and returns output similar to the following:

```
Oracle VM Manager Release 3.3.1 Admin tool

Please enter the password for weblogic :

Initializing WebLogic Scripting Tool (WLST) ...

Welcome to WebLogic Server Administration Scripting Shell

Type help() for help on available commands

Connecting to WebLogic server ...

Connected ...
Listing Oracle VM users ...
User : OracleSystemUser
User : weblogic
User : admin
User : [ovmuser]
Listed users successfully ...
Exiting...
```

Some of the users stored within Weblogic and listed are critical to the healthy functioning of your Oracle VM Manager environment. These include:

- *OracleSystemUser*
- *weblogic*

The default *admin* user account is also typically listed. Any other user accounts listed, such as the [ovmuser] account, have been added to the system after installation.

4.1.1.5 Locking User Accounts

In order to protect unauthorized access to Oracle VM Manager it is possible to configure an account locking facility that is triggered after a number of failed attempts to log in. This is achieved using the Oracle VM Administrator Tool in the following way:

```
# ./ovm_admin --lockusers [3]
```



Note

Account locking is enabled by default according to the base Weblogic configuration. The default settings allow for 5 invalid login attempts before the account is locked.

The lock period is set to 30 minutes. The only way to change the lock period is to edit the underlying Weblogic configuration directly. For more information on configuring the Weblogic lockout parameters, please refer to the Weblogic documentation on this at:

http://docs.oracle.com/cd/E13222_01/wls/docs81/ConsoleHelp/security_realm_userlockout.html

Important

This is a global parameter that applies to all users. Setting this parameter on an instance of Oracle VM Manager that makes use of a single administrator account can result in this account being locked for 30 minutes before anybody is able to use it again. To recover from this it is possible to unlock the account. See [Section 4.1.1.6, “Unlocking User Accounts”](#).

You are prompted to enter the Weblogic password in order to apply this setting. Typical output from the command follows:

```
Oracle VM Manager Release 3.3.1 Admin tool

Please enter the password for weblogic :

Initializing WebLogic Scripting Tool (WLST) ...

Welcome to WebLogic Server Administration Scripting Shell

Type help() for help on available commands

Connecting to WebLogic server ...

Connected ...
Setting Invalid Login attempts to '[3]' ...
Exiting...
Restart of Oracle VM Manager is required for Data Store change to take effect ...
```

Note that you are required to restart Oracle VM Manager in order for the setting to take effect.

An account is locked for 30 minutes before it is automatically unlocked again.

4.1.1.6 Unlocking User Accounts

When account locking is enabled (see [Section 4.1.1.5, “Locking User Accounts”](#)), it is possible for Oracle VM Manager user accounts to become locked for up to 30 minutes if a user fails to authenticate after the number of attempts that has been configured for this facility. When a user's account has become locked and the user enters the correct username and password combination, an error appears when the user attempts to authenticate:

```
Unexpected error during login (javax.security.auth.login.LoginException),
please consult logs for details.
```

An investigation of the `AdminServer.log` reveals:

```
000000000183> >1358953290200< >BEA-090078< >User ovmuser in security realm myrealm
has had 3 invalid login attempts, locking account for 30 minutes.<
```

It is possible to override the 30 minute lock on an account by using the Oracle VM Administrator Tool in the following way:

```
# ./ovm_admin --unlockuser [ovmuser]
```

You are prompted for the Weblogic account password in order to complete the operation.

4.1.2 Log Rotation Options

The Oracle VM Administrator Tool allows you to control how and when log files are rotated. There are two options available:

- `--rotatelogsdaily`: set the logs to be rotated on a daily basis at an allocated time
- `--rotatelogsbysize`: set the logs to be rotated when they reach a specified size

In both cases, you are prompted for the Weblogic password in order to update the configuration.

4.1.2.1 Rotate Logs Daily

To set the logs to rotate daily at an allocated time, run the Oracle VM Administrator Tool in the following way:

```
# ./ovm_admin --rotatelogsdaily [00:30]
```

The time provided is specified in the format HH:MM.

Typical output from the command follows:

```
Oracle VM Manager Release 3.3.1 Admin tool
Please enter the password for weblogic :
Initializing WebLogic Scripting Tool (WLST) ...
Welcome to WebLogic Server Administration Scripting Shell
Type help() for help on available commands
Connecting to WebLogic server ...
Connected ...
Configure log rotation setting to rotate daily at [00:30] ...
Modified log rotation setting successfully ...
Exiting...
```

4.1.2.2 Rotate Logs By Size

To set the logs to rotate when they reach a specified size, run the Oracle VM Administrator Tool in the following way:

```
# ./ovm_admin --rotatelogsbysize [1024]
```

The size provided is specified according to the number of kilobytes before rotation.

Typical output from the command follows:

```
Oracle VM Manager Release 3.3.1 Admin tool
Please enter the password for weblogic :
Initializing WebLogic Scripting Tool (WLST) ...
Welcome to WebLogic Server Administration Scripting Shell
Type help() for help on available commands
```



```
Connecting to WebLogic server ...  
Connected ...  
Configure log rotation setting to rotate the logs based on size ([1024] KB) ...  
Modified log rotation setting successfully ...  
Exiting...
```

4.2 Oracle VM Diagnostic Capture

For diagnostic purposes, Oracle Support Services use a script called [VMPInfo3](#) that automatically collects vital troubleshooting information from your Oracle VM environment. This script is installed when you install Oracle VM Manager and is located at `/u01/app/oracle/ovm-manager-3/ovm_shell/tools/vmpinfo/vmpinfo3.sh` on the Oracle VM Manager system.

The script is usually run with a minimum of two parameters, as follows:

```
./vmpinfo3.sh --username=[admin] --password=[passwd]
```

The script creates a tarball in the `/tmp` directory. When the script has finished running, it notifies you of the filename for the tarball that it has created and instructs you to send the file to Oracle Support. An example of the output follows:

```
=====
Please send /tmp/vmpinfo3-3.x.y.z-20130123-163252.tar.gz to Oracle OVM support
=====
```

For detailed information about the script, its purpose and usage, please consult the support note with [Doc ID 1521931.1](#). You can also find this document by logging on to [My Oracle Support](#) and searching the knowledge base for "*vmpinfo3*".

