

Oracle Linux Manager

Client Life Cycle Management Guide for Release 2.10



F36090-18
July 2025



Oracle Linux Manager Client Life Cycle Management Guide for Release 2.10,
F36090-18

Copyright © 2022, 2025, Oracle and/or its affiliates.

Contents

Preface

Conventions	viii
Documentation Accessibility	viii
Access to Oracle Support for Accessibility	viii
Diversity and Inclusion	ix

1 About Tools for Administering Oracle Linux Manager

About the Oracle Linux Manager Web Interface	1-1
Overview Page	1-1
Systems Tab	1-2
Errata Tab	1-3
Channels Tab	1-4
Audit Tab	1-5
Configuration Tab	1-5
Schedule Tab	1-6
Users Tab	1-6
Admin Tab	1-7
About the spacecmd Command	1-7

2 Creating Software Channels and Repositories

About Software Channel Configuration	2-1
Configuring Software Channels for ULN	2-3
Obtaining Packages From the Oracle Linux Yum Server	2-4
Configuring Oracle Linux Manager Repositories	2-5
Using the Oracle Linux Manager Web Interface	2-6
Note About Using Special Characters	2-7
Using the repo_create Command	2-7
Configuring Software Channels	2-8
Using the Oracle Linux Manager Web Interface	2-9
Using the softwarechannel_create Command	2-11
Synchronizing Software Channels	2-13
Memory Considerations When Building Repositories	2-13

Using the Oracle Linux Manager Web Interface	2-15
Using the softwarechannel_syncrepos Command	2-16
Using the spacewalk-repo-sync Command	2-17
Cloning Software Channels	2-18
Using the Oracle Linux Manager Web Interface	2-18
Using the softwarechannel_clone Command	2-20
Using the spacewalk-clone-by-date Command	2-20
Adding Packages to Channels With Defined Modules	2-21
Managing Channel Life Cycles	2-21

3 Working With Activation Keys in Oracle Linux Manager

About Oracle Linux Manager Activation Keys	3-1
Setting Up Activation Keys	3-1
Using the Oracle Linux Manager Web Interface	3-2
Using the activationkey_create Command	3-3

4 Provisioning Client Systems

About Kickstart Trees, Distributions, and Profiles	4-1
Setting Up Kickstart Trees	4-2
Creating Kickstart Distributions	4-3
Using the Oracle Linux Manager Web Interface	4-3
Using the distribution_create Command	4-4
Configuring Kickstart Profiles	4-5
Using the Oracle Linux Manager Web Interface	4-6
Adding GPG Keys and SSL Certificates	4-6
Creating Profiles	4-6
Configuring and Modifying Profiles	4-8
Using the kickstart_create Command	4-11
Installing Client Systems by Using Kickstart	4-12
Configuring Cobbler and DHCP to Support Network Booting	4-13
Adding a PXE Client To Be Provisioned by Oracle Linux Manager	4-15
About Boot Loader Configuration Files	4-17
Configuring DHCP to Support iPXE Clients	4-19
Creating a Kickstart Profile in Cobbler	4-22
Adding a PXE Client To Be Provisioned by Cobbler	4-24
Removing a PXE Client Definition From Cobbler	4-26
Configuring Kernel Based Virtual Machines on Oracle Linux Manager	4-26
Provisioning KVM Hosts	4-26
Provisioning KVM Guests	4-29

5	Registering Client Systems With Oracle Linux Manager	
	Registering a Client System by Using Kickstart	5-2
	Installing Oracle Linux Manager Client Software and Registering the Client System	5-2
	Installing on Oracle Linux 7 Hosts	5-2
	Installing on Oracle Linux 8 Hosts	5-3
	Installing on Oracle Linux 9 Hosts	5-4
	Installing on CentOS 7 Hosts	5-5
	Installing on CentOS 8 Hosts	5-7
	Registering a Client System Only	5-8
	Upgrading the Spacewalk Client to Oracle Linux Manager 2.10	5-8
6	Configuring Client Systems for Immediate Updates	
	Enabling the OSA Daemon in Oracle Linux Manager	6-1
	Configuring a Kickstart Profile	6-1
	Configuring a Kickstart File	6-2
	Enabling the OSA Daemon Manually	6-2
	Querying Status of Client Systems	6-3
	Using the Oracle Linux Manager Web Interface	6-3
	Using the <code>system_details</code> Command	6-4
7	Configuring Client Systems for Remote Management	
	Using the Oracle Linux Manager Web Interface	7-1
	Configuring Kickstart Features	7-1
	Enabling Remote Management in a Kickstart Profile	7-1
	Enabling Remote Management in a Kickstart File	7-2
	Configuring Channels	7-2
	Using the <code>rhn-actions-control</code> Command	7-3
8	Configuring System Groups to Manage Client Systems	
	Setting Up System Groups	8-1
	Using the Oracle Linux Manager Web Interface	8-1
	Using the <code>group_create</code> Command	8-3
	Searching for Systems by Using the <code>system_search</code> Command	8-4
9	Updating Client Systems	
	Subscribing Client Systems to Software Channels	9-1
	Using the Oracle Linux Manager Web Interface	9-1
	Using Commands in a <code>spacecmd</code> Session	9-2

Using the spacewalk-channel Command	9-4
Applying Available Security Updates and Errata	9-4
Using the Oracle Linux Manager Web Interface	9-5
Using the system_applyerrata Command	9-6
Managing Client Packages for Systems	9-8
Using the Oracle Linux Manager Web Interface	9-8
Using Commands in a spacecmd Session	9-11
Managing Packages for System Groups	9-12
Using the Oracle Linux Manager Web Interface	9-12
Using Commands in a spacecmd Session	9-13

10 Controlling and Configuring Client Systems

Running Command Scripts on Remote Clients	10-1
Using the Oracle Linux Manager Web Interface	10-2
Using the system_runscript Command	10-3
Working With Scheduled Events	10-3
Working With Configuration Channels	10-4
Creating Customer Information Keys	10-4
Using the Oracle Linux Manager Web Interface	10-4
Using the custominfo_createkey Command	10-5
Applying Custom Information Keys	10-5
Setting Up Configuration Channels	10-5
Using the Oracle Linux Manager Web Interface	10-6
Using the configchannel_create Command	10-7
Subscribing Client Systems to Configuration Channels	10-8
Using the Oracle Linux Manager Web Interface	10-8
Using the system_addconfigchannels Command	10-9
Deploying Configuration Files on Client Systems	10-10
Using the Oracle Linux Manager Web Interface	10-10
Using the system_deployconfigfiles Command	10-11

11 Using OpenSCAP to Audit of Client Systems

Requirements for Using OpenSCAP in Oracle Linux Manager	11-1
Performing Audit Scans	11-1
Using the Oracle Linux Manager Web Interface	11-2
Using the scap_schedulexccdfscan Command	11-3

12 Configuring Ksplice Offline Clients

Supported Kernels	12-2
Configuring an Oracle Linux Manager Server to Act as a Ksplice Mirror	12-2
Provisioning Client Systems as Ksplice Offline Clients	12-3
Installing and Configuring Existing Client Systems as Ksplice Offline Clients	12-3
Implementing Ksplice Synchronization Filters on Oracle Linux Manager Server	12-4

13 Kickstart Options

14 Sample Minimum Package Lists

15 Configuration File Macros

16 Oracle Linux Manager XML/RPC API

Preface

WARNING:

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

[Oracle Linux Manager: Client Life Cycle Management Guide for Release 2.10](#) describes how to provision and manage Oracle Linux Manager 2.10 clients. Oracle Linux Manager is based on the Spacewalk open source software.

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.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <https://www.oracle.com/corporate/accessibility/>.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab>.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

About Tools for Administering Oracle Linux Manager

WARNING:

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

To administer Oracle Linux Manager for Oracle Linux, you use either the Oracle Linux Manager web interface or the `spacecmd` command, both of which become available after an Oracle Linux Manager installation. However, as an option, you can also create your own web and command interfaces by using Oracle Linux Manager XML/RPC API. For more information, see [Oracle Linux Manager XML/RPC API](#).

About the Oracle Linux Manager Web Interface

As a first task after a completed installation, you created the initial organization and the main Oracle Linux Manager administrator account by using the Oracle Linux Manager web interface. See [Configuring an Oracle Linux Manager Administrator and an Initial Organization in Oracle Linux Manager: Installation Guide for Release 2.10](#).

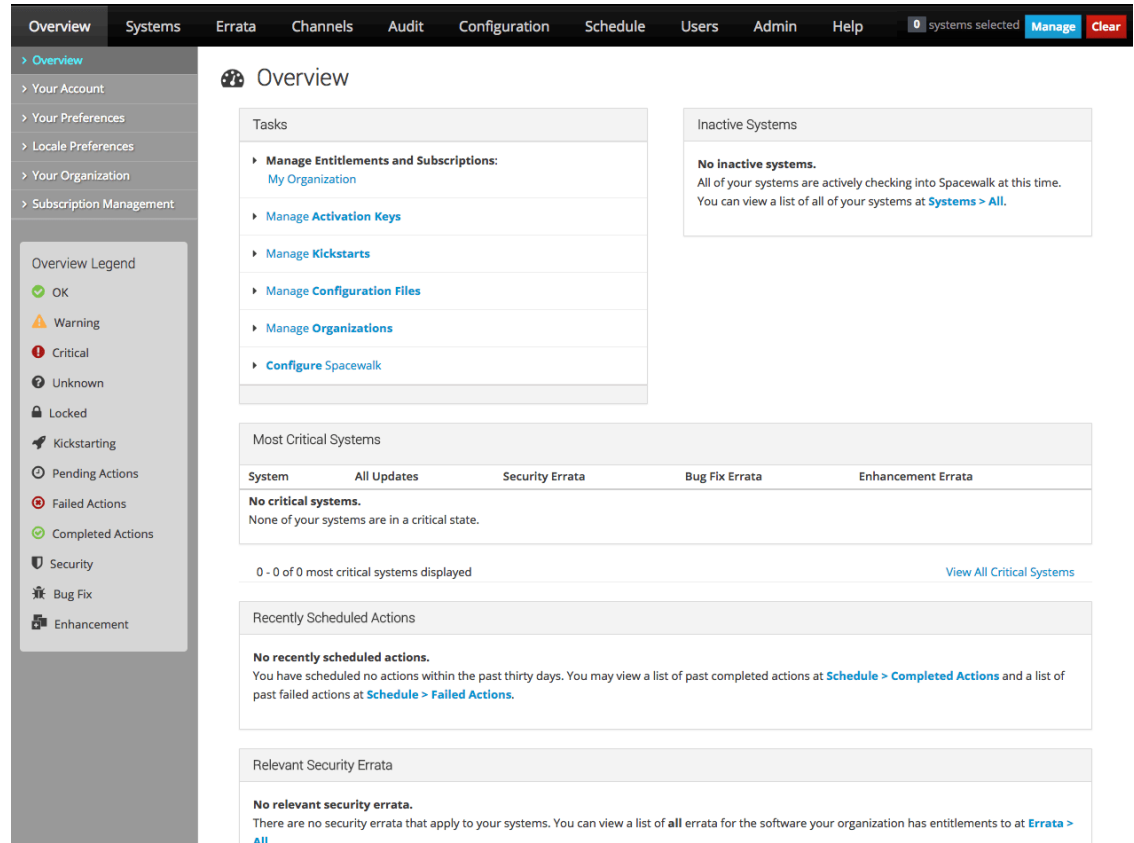
This section introduces you to other web interface pages. These pages correspond to specific tabs. Every page displays summary information as well as a legend pane that explains the meaning of the different icons that appear on that page.

To access the web interface, go to your Oracle Linux Manager server's URL, which is typically the server's FQDN such as `https://swksvr.mydom.com`. Log in with the Oracle Linux Manager administrator's credentials that you created during the initial configuration process.

Overview Page

The Overview page is the default landing page after a successful login.

Figure 1-1 Oracle Linux Manager Overview Page



The default Overview page presents a dashboard view of the state of Oracle Linux Manager server. The page displays basic information about systems that are inactive or in a critical state, recently scheduled actions, relevant security errata that you can apply to your systems, and lists of system groups and recently registered systems. The page also provides links to administrative tasks that you can perform.

To customize the layout of the Overview page:

1. Select **Your Preferences**.
2. On the Your Preferences page in the Overview Start Page section, select or deselect the check boxes against the information that you want or do not want to display.

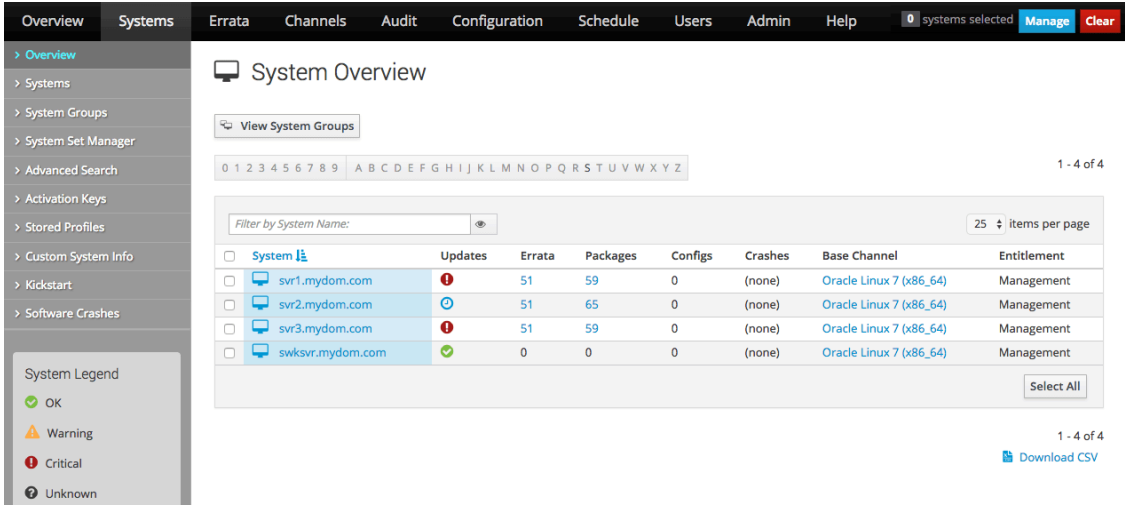
You can configure other preferences on this page, such as whether to receive email notifications, the number of entries per-page in lists, and the separator character for comma-separated values (CSV) files.

3. Click **Save Preferences**.

Systems Tab

The Systems tab opens the Systems Overview Page.

Figure 1-2 System Overview Page

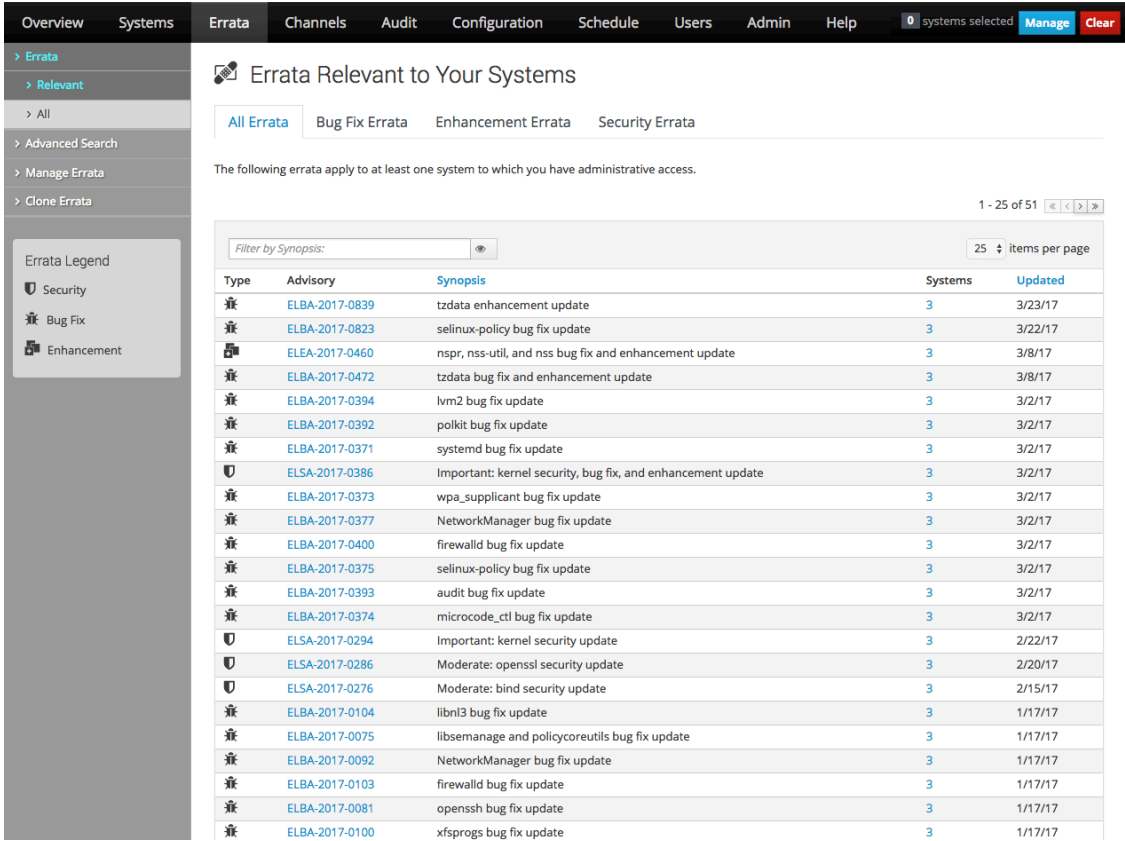


The System Overview page displays a summary of the number of available updates, errata, packages, configuration files and crashes, the name of the base channel, and any entitlements for each managed client system.

Errata Tab

The Errata tab opens the Errata Relevant to Your Systems page

Figure 1-3 Errata Relevant to Your Systems Page

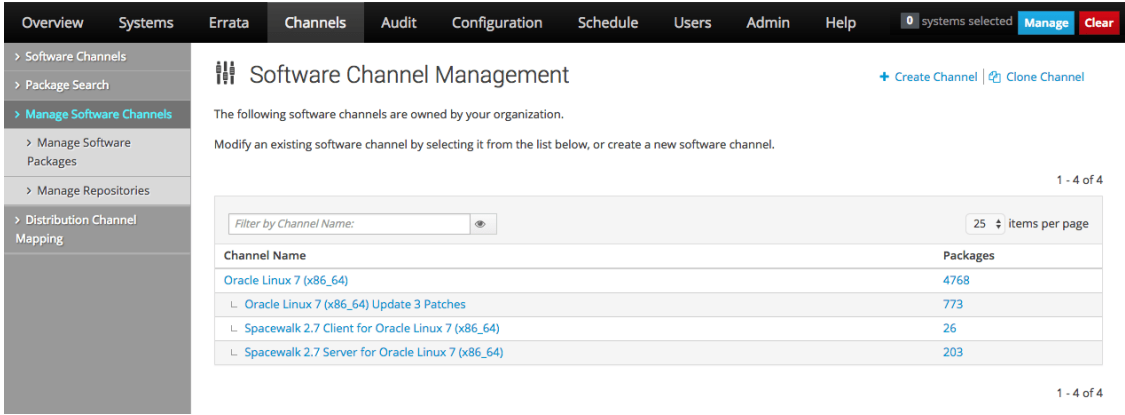


The Errata Relevant to Your Systems page displays information about the errata that are available for your registered systems.

Channels Tab

The Channels tab opens the Software Channel Management page.

Figure 1-4 Software Channel Management Page

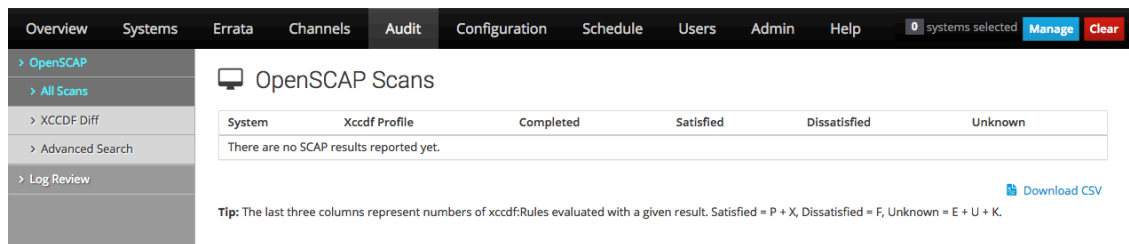


The Software Channel Management page displays the channels to which you can subscribe your registered systems. By default, only the base channels are shown. To display child channels, click the plus sign (+) next to the name of the base channel.

Audit Tab

The Audit tab opens the OpenSCAP Scans page.

Figure 1-5 OpenSCAP Scans Page

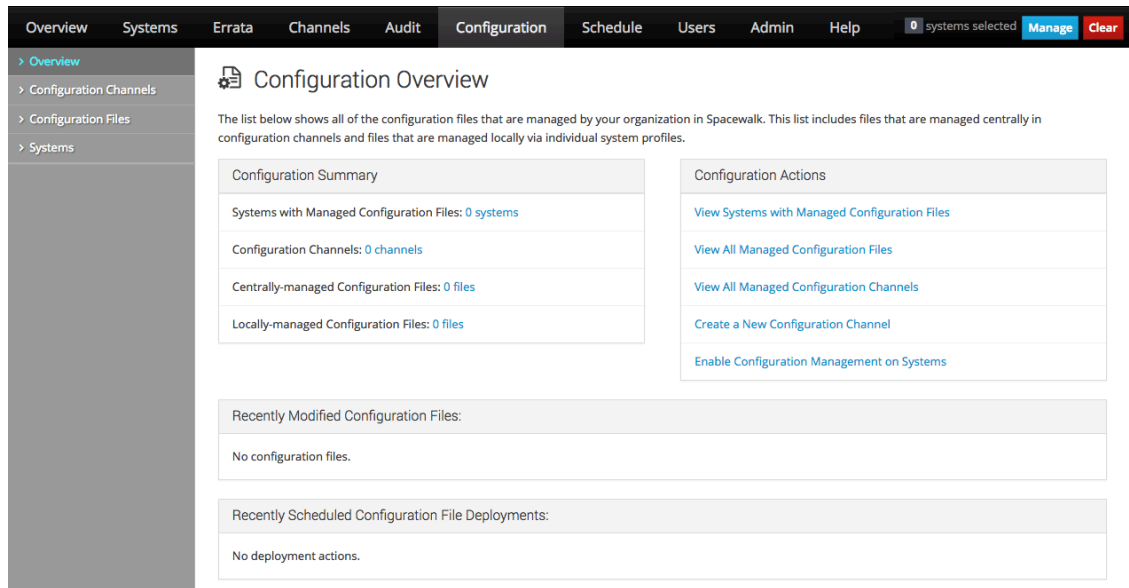


The OpenSCAP Scans page displays a summary of any scans that you have performed on your systems.

Configuration Tab

The Configuration tab opens the Configuration Overview page.

Figure 1-6 Configuration Overview Page

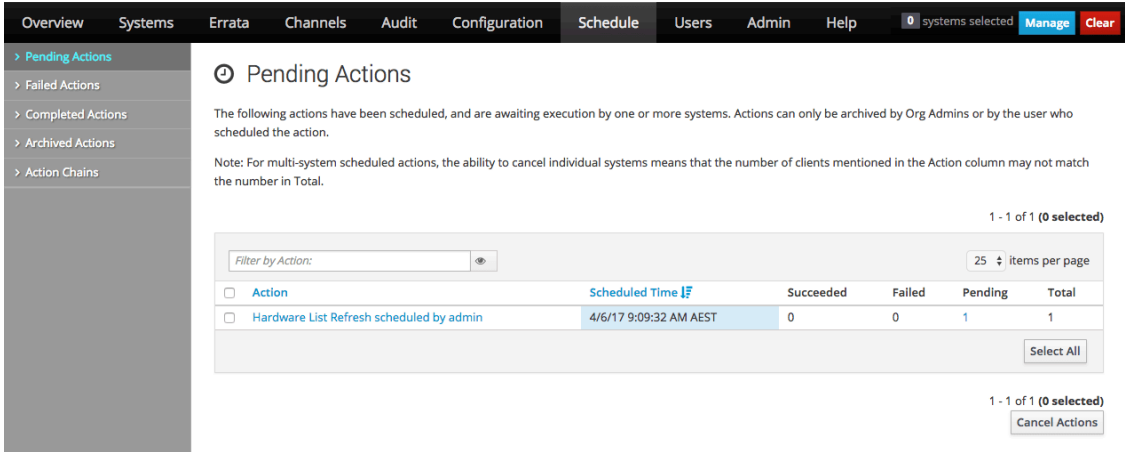


The Configuration Overview page displays a summary of the configuration files that are known to Oracle Linux Manager, links to actions you can perform with configuration files, and scheduled deployments of configuration files.

Schedule Tab

The Schedule tab opens the Pending Actions page.

Figure 1-7 Pending Actions Page

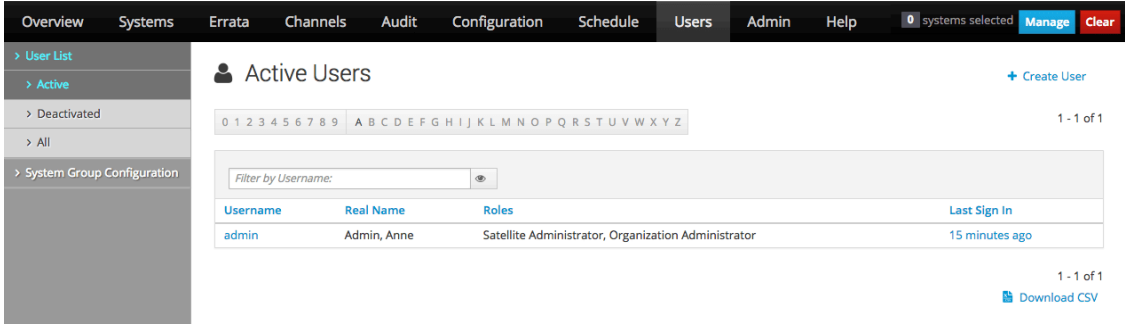


The Pending Actions page displays a list of actions that are scheduled to be performed.

Users Tab

The Users tab opens the Active Users page.

Figure 1-8 Active Users Page



The Active Users page displays a list of administrators or other users and their allocated roles. The Oracle Linux Manager Administrator role permits a user to perform all actions in Oracle Linux Manager.

You can configure the Organization Administrator role to grant one or more of the following roles to a user who has administrative access to one or more organizations:

- System Group Administrator
- Channel Administrator

- Activation Key Administrator
- Configuration Administrator

A read-only API user has limited access to the XML/RPC API but cannot access the web interface.

Admin Tab

The Admin tab opens the Organizations page.

Figure 1-9 Organizations Page

Organization	Systems	Users	Trusts
Demo Inc *	1	1	0

*Tip: This organization is your Spacewalk's default organization.

The Organizations page displays the organizations for which you have configured Oracle Linux Manager server to administer and the number of systems, Oracle Linux Manager administrators, and trusts that are configured for the system. If a trust is enabled, you can share content and move systems between organizations.

About the spacecmd Command

The `spacecmd` command enables you to perform most of the functions that are available in the web interface. You can run the `spacecmd` command as an interactive shell by typing the command by itself without arguments or options. In this guide, the `spacecmd` examples show the use of the utility in the interactive shell mode.

You can issue `spacecmd` commands either locally on Oracle Linux Manager server or remotely. To run the command remotely, identify the server by its IP address or its FQDN, for example:

```
sudo spacecmd -s swksvr.mydom.com
```

You must enter your Oracle Linux Manager user name and password as prompted to gain full access to the interactive mode. The Oracle Linux Manager credentials were created together with the initial organization.

To access `spacecmd` securely while bypassing typing user credentials, do the following steps:

1. Create a file such as `~/.spacecmd/config` that contains the credentials, for example:

```
[spacecmd]
server=swksvr.mydom.com
username=swadmin
password=password
```

2. Restrict the file's permissions.

```
sudo chmod 400 ~/.spacecmd/config
```

To run the command non-interactively, use the following syntax:

```
sudo spacecmd [options] -- arguments
```

arguments consists of the subcommand, subcommand options, and parameter values.

For example, to create channels in the non-interactive mode, you might type the following:

```
sudo spacecmd -- softwarechannel_create -l oraclelinux7-x86_64-ksplICE \
-n "Oracle Linux 7 x86_64 KsplICE Channel" -p oraclelinux7-x86_64 -a x86_64
```

```
INFO: Connected to https://swksvr.mydom.com/rpc/api as swadmin
```

The `spacecmd` options are different from subcommand options. You specify these options before entering the subcommand and its options. For example, `spacecmd`'s `-q` option suppresses informational messages while the `-y` option skips all confirmation prompts.

The following example illustrates the use of `spacecmd` options.

```
sudo spacecmd -q -- softwarechannel_list
```

```
oraclelinux7-x86_64
oraclelinux7-x86_64-addons
oraclelinux7-x86_64-ksplICE
oraclelinux7-x86_64-optional
oraclelinux7-x86_64-patch
oraclelinux7-x86_64-spacewalk26-client
oraclelinux7-x86_64-spacewalk26-server
oraclelinux7-x86_64-uek-r3
oraclelinux7-x86_64-uek-r4
```

To skip informational messages and confirmation prompts while deleting a channel, you would type, for example:

```
sudo spacecmd -q -y -- softwarechannel_delete oraclelinux7-x86_64-ksplICE
```

```
Channels
-----
oraclelinux7-x86_64-ksplICE
```

The `softwarechannel_delete` subcommand displays the channel to be deleted. When you run the `softwarechannel_list` subcommand again, the deleted channel is no longer included in the list.

For more information, see the `spacecmd(1)` man page.

 **Note:**

Other commands to administer Oracle Linux Manager are also available through Oracle Linux Manager packages downloaded at installation. These commands are run directly from the command line prompt rather than in a `spacecmd` shell. They require that you have been granted permission in the `/etc/sudoers` file.

2

Creating Software Channels and Repositories

WARNING:

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

This chapter describes how to create software channels in Oracle Linux Manager. Client systems subscribe to channels to obtain packages and errata. Each channel is associated with at least one repository that defines the source of the packages and errata.

The examples in this chapter use the Unbreakable Linux Network (ULN) and the Oracle Linux yum server. However, you can also use Oracle Linux Manager to obtain software packages from other internal or external sources.

Caution:

If you are using Oracle Linux 8 or Oracle Linux 9, note the limitations of channel manipulation if channels contain modules. See Known Issues in Oracle Linux Manager in [Oracle Linux Manager: Release Notes for Release 2.10](#) .

About Software Channel Configuration

ULN has more than 100 unique channels for supported architectures. To access ULN, go to <https://linux.oracle.com/>.

ULN provides an `olN_arch_[baseos]_latest` label for the repository that includes all of the latest versions of the packages in a distribution, including any errata that are also provided in the patch channel.

Note:

The inclusion of `baseos` in the label applies only to Oracle Linux 8 and Oracle Linux 9.

If no vulnerabilities have been found in a package, the package version might be the same as the version that was included in the original distribution. For other packages, the version is the same as the version that is provided in the patch channel for the highest update level. For example, for Oracle Linux 8 using the `x86_64` architecture, the `ol8_x86_64_baseos_latest`

channel contains a combination of the `ol8_x86_64_u1_baseos_base` as well as the `ol8_x86_64u1_baseos_patch` channels.

 **Important:**

Do **not** register an Oracle Linux Manager server or client with ULN. Instead, register an Oracle Linux Manager server as a client of itself to receive updates.

The Oracle Linux yum server provides the `olN_latest` label for the repository that includes all of the packages for an entire Oracle Linux release, in addition to the `olN_uN_base` repository for each update.

Some channels, such as the `patch` channel, and those for DTrace userspace, Ksplice, and OFED are available on ULN, but not the Oracle Linux yum server.

ULN also provides older versions of packages that are added to a parent channel. The `_archive` suffix is usually added to the channel for which it hosts archive packages. For example, the `_latest` channels have equivalent `_latest_archive` channels to host older versions of packages that have been updated in the `_latest` channels.

Packages are moved to an archive channel when newer versions of the same packages are added to the parent channel. The addition of archive channels helps keep the metadata for the parent channel manageable and also keeps the overall size of the channel to a minimum. If you require an earlier version of a package, you can subscribe to the equivalent `_archive` channel to obtain it. When performing an installation or downgrade, you must specify the version of the package that you want to install.

 **Note:**

Avoid installing packages from an `_archive` channel. Doing so might result in your system running software that has since been patched for security related issues. Your system becomes exposed to vulnerabilities that could be exploited for malicious purposes.

ULN also provides `base` and `patch` channels for each update of an Oracle Linux release. Depending on the Oracle Linux release, other channels might provide the latest packages for additional features such as DTrace user-space, Ksplice, and the OpenFabric Enterprise Edition (OFED) feature. The Oracle Linux yum server, however, does not provide `patch` channels for updates.

Other channels might also be available, such as `_beta` channels for beta versions of packages.

As each new, major version or minor update of Oracle Linux becomes available, Oracle creates new `base` and `patch` channels for each supported architecture to distribute new packages. The existing `base` and `patch` channels for the previous versions or updates remain available and do not include the new packages. The `_latest` channel distributes the latest possible version of any package, and tracks the top of the development tree independently of the update level.

Oracle recommends that you design a channel configuration that is based on your particular work flow. For example, if you intend to use Oracle Linux Manager's channel cloning feature to

promote systems from development through testing to production, you could configure a `base` channel and child `patch` channel together with other child channels.

If you duplicate child channels, you do not need to duplicate their repositories. These channels can use the same repositories as the channel from which they were cloned. For example, each cloned base channel might have a unique `addons` child channel, but each of the child channels would use the same repository. The packages are not duplicated, even though they are referenced in multiple channels.

To make archive channel content available to local clients, you must first create repositories for each ULN archive channel that you want to synchronize, then map these repositories to the software channels that you have created in Oracle Linux Manager.

If necessary, you can maintain the `latest` channels separately, without subscribing any systems to these channels. If the need arises, you can copy errata packages from the `latest` channels to the `patch` channels to make the latest fixes available.

 **Note:**

You do not need to associate a software channel with a repository if you want to create custom channels that obtain their packages by methods such as `rhnpush` or uploading by using the web interface. These methods enable you to serve locally developed and packaged software that has no upstream repository.

For a complete description of the main channels that are available on ULN, see the chapter on ULN registration in [Oracle Linux: Unbreakable Linux Network User's Guide for Oracle Linux 6 and Oracle Linux 7](#).

Configuring Software Channels for ULN

Oracle Linux Manager servers and clients must **not** be registered with ULN. Instead, register an Oracle Linux Manager server as a client of itself to receive updates. Oracle Linux Manager contains a ULN plug-in for the `spacewalk-repo-sync` tool. The plug-in enables you to synchronize software channels without having to register Oracle Linux Manager server with ULN.

To configure the plug-in:

1. Grant read-write permissions to `/etc/rhn/spacewalk-repo-sync/uln.conf`.
2. Edit the file by adding your SSO login user name and authentication token for ULN:

```
[main]
username=ULN_SSO_username
password=ULN_Authentication_Token
```

3. Revert the file's permissions to read-only.
4. (Optional) Verify that the read-only permission has been restored to the file.

```
sudo ls -l /etc/rhn/spacewalk-repo-sync/uln.conf
```

Ensure that this file has read-only permissions by `root` to protect your ULN credentials.

After you have configured the ULN plug-in, you can use either the Oracle Linux Manager web interface, the `spacecmd` command, or the `spacewalk-common-channels` command to create Oracle Linux Manager software channels, repositories, and activation keys.

 **Tip:**

Although the `spacewalk-common-channels` configures software channels to access the Oracle Linux yum server, you can configure the repository entries to access ULN instead. For example, if you want to use the Oracle Linux `base` and `patch` channels for an Oracle Linux release update on ULN, reconfigure the `base` software channel to access the `base` channel and create an additional child channel and associated repository entry for the `patch` channel.

Obtaining Packages From the Oracle Linux Yum Server

Use the `spacewalk-common-channels` command to configure software channels that use the Oracle Linux yum server. This command also enables you to configure repositories, GPG keys, and activation keys for Oracle Linux releases.

 **Note:**

The `spacewalk-common-channels` currently cannot yet create Oracle Linux 9 channels.

To list the available channels, use the `--list` option:

```
sudo spacewalk-common-channels --list | grep "oracle"
```

 **Note:**

Unlike ULN, the Oracle Linux yum server does not provide `patch` channels for each update of an Oracle Linux release. Instead, the `spacewalk-common-channels` command configures the `base` (parent) software channel to use the `olN_latest` repository, which includes all of the packages for the entire release.

Some ULN channels, such as those for DTrace userspace, Ksplice, and OFED, are not available on the Oracle Linux yum server.

For example, you would create the software channels for Oracle Linux 8 (x86_64) as follows:

```
sudo spacewalk-common-channels -v -u swadm -p swadm_passwd -a x86_64 -k unlimited  
'oraclelinux8*'
```

The `-k unlimited` option specifies that the command should create an activation key with no limit on the number of servers with which you can use it. Otherwise, no activation key is activated.

See the following for more information and instructions:

- [Configuring Oracle Linux Manager Repositories](#)
- [Configuring Software Channels](#)
- [Working With Activation Keys in Oracle Linux Manager](#)

After you have set up the software channels and repositories, download the packages by synchronizing the software channels with the Oracle Linux yum server. See [Synchronizing Software Channels](#).

Configuring Oracle Linux Manager Repositories

Oracle Linux Manager repositories define where to obtain packages from ULN or the Oracle Linux yum server.

For ULN, an Oracle Linux Manager repository specifies the URL of a ULN channel in the following format:

```
uln:///ULN_channel_label
```

To obtain a list of available ULN channel labels, do the following:

1. Log in to ULN (<https://linux.oracle.com>) and then select the **Channels** tab.
2. On the Channels page, select the release and architecture from their respective pull-down menus.

The resulting page displays the channels, their labels, and their descriptions. The URL for Oracle Linux 8 Application Stream Packages (x86_64) channel, for example, would be `uln:///ol8_x86_64_appstream`.

For the Oracle Linux yum server, an Oracle Linux Manager repository specifies the URL of a repository in the following format:

```
https://yum.oracle.com/repository_path
```

To obtain the correct URL for a repository, do the following:

1. At <https://yum.oracle.com/>, select your Oracle Linux release from the list of repositories, for example, Oracle Linux 7 .
2. From the list of latest packages, click the link to the packages that corresponds to your system's architecture, for example, x86_64.

A list of the RPM packages is displayed. The page also includes the Direct YUM Repository URL of the repository you are viewing.

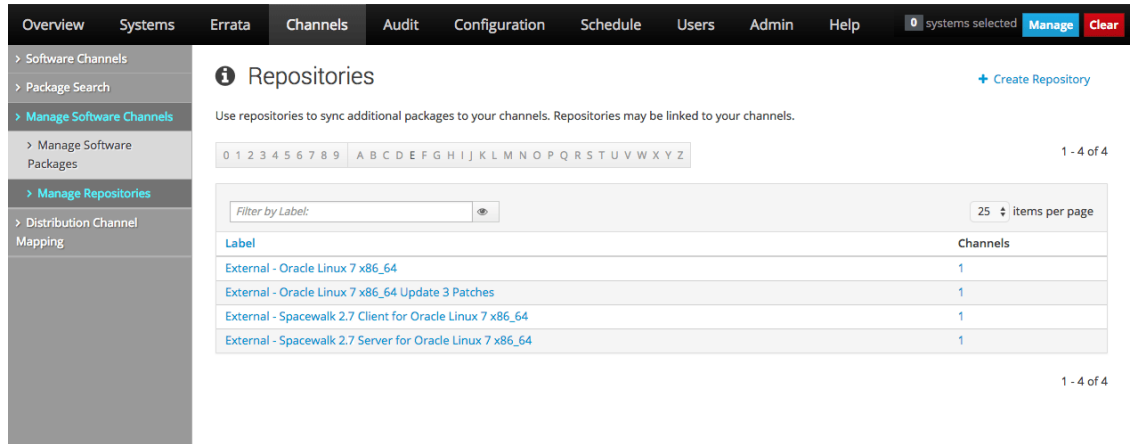
Note:

For Oracle Linux 8 repository access, you must install the appropriate `release-el8` packages and enable any repositories to which you require access prior to configuring yum repositories.

For more detailed information about DNF and Yum repository configuration in Oracle Linux 8, see [Oracle Linux: Managing Software on Oracle Linux](#). See also the `dnf(5)` manual page.

Using the Oracle Linux Manager Web Interface

Figure 2-1 Repositories Page



Go to **Channels**, select **Manage Software Channels**, and then select **Manage Repositories**:

- To create a repository:
 - Click **+ Create Repository**.
 - On the Create New Repository page, enter the following repository settings:

Repository Label

Name for the repository, for example, Oracle Linux 8 (x86_64).

Repository URL

URL of the source for the repository's packages, for example, `uln:///ol8_x86_64_baseos_latest` or `https://yum.oracle.com/repo/OracleLinux/OL8/baseos/latest/x86_64`.

Repository Type

Source of the repositories depending on the **Repository URL** setting, which is either `uln` or `yum`.

Leave the remaining fields unset for Oracle Linux.

- Click **Create Repository** to create the repository.
- To view a repository, select its entry to display its details.
 - To modify a repository:
 - Select the repository that you want to edit.
 - On the Repository Details page, modify the repository settings and click **Update Repository** to save your changes.
 - To delete a repository:
 - Go to **Channels**, select **Manage Software Channels**, and then select **Manage Repositories**.

2. Select the repository that you want to delete.
3. On the Repository Details page, click **Delete repository**, and then confirm the deletion.

To associate a software channel with a repository, see [Using the Oracle Linux Manager Web Interface](#).

Note About Using Special Characters

Both Oracle Linux Manager web interface and `spacecmd` subcommands enable you to configure Oracle Linux Manager repositories, channels, and so on. However, they differ when handling special characters.

With the web interface, special characters such as parentheses are automatically dropped when you click the Create button. Thus, `(x86_64)`, for example, becomes stored as `x86_64`.

However, the `spacecmd` command preserves these characters. Thereafter, if you continue to use the command line on channels or repositories whose names contain special characters, you must use the backslash (`\`) escape character for these special characteres to be parsed correctly. See the example for `repo_details` in [Using the repo_create Command](#).

Tip:

To facilitate your work, do not use special characters when you create channels and repositories with the command line. Thus, you would avoid unexpected results, especially if you regularly switch between the web interface and command line when administering Oracle Linux Manager.

Using the repo_create Command

Create a repository by using the `repo_create` command as follows. The information you need to provide is in bold:

```
spacecmd {SSM:0}> repo_create

Name: Oracle Linux 8 x86_64
URL: uln:///ol8_x86_64_baseos_latest
Type: uln
...
```

To list all repositories, use the `repo_list` command:

```
spacecmd {SSM:0}> repo_list
```

To list the details of a repository, use the `repo_details` command:

```
spacecmd {SSM:0}> repo_details "repository"
```

For example, listing the details of the Oracle Linux 8 x86_64 repository would display the following:

```
spacecmd {SSM:0}> repo_details "Oracle Linux 8 x86_64"

Repository Label:          Oracle Linux 8 x86_64
Repository URL:           uln:///ol8_x86_64_baseos_latest
Repository Type:          uln
```

```
Repository SSL Ca Certificate:    None
Repository SSL Client Certificate: None
Repository SSL Client Key:       None
```

 **Note:**

If you used the command line and created repositories with special characters, for example, Oracle Linux 8 (x86_64), then you would type:

```
spacecmd {SSM:0}> repo_details "Oracle Linux 8 \ (x86_64\)"
```

The example shows how you need to use the backslash character for the command to correctly parse the parentheses in the repository name. Preferably, avoid using special characters in repository or channel names. See [Note About Using Special Characters](#).

To delete a repository, use the `repo_delete` command. Ensure that you confirm the command as prompted.

```
spacecmd {SSM:0}> repo_delete "repository"
```

To associate a software channel with a repository, see [Using the softwarechannel_create Command](#).

Configuring Software Channels

The main software channel for an Oracle Linux release is referred to as the *base* or *parent* software channel. A base channel has associated child software channels. Each child software channel provides packages that are not available with the base software channel. If multiple versions of a package exist in different subscribed channels, `yum` versioning and dependency resolution ensure that the most up-to-date version of a package is installed.

You can subscribe a client to a single base channel and one or more of its child channels. For channels that are not specific to an update, such as `addons`, create an `addons` child channel for each update-level base channel and associate this child channel with the same `addons` repository.

If you set up Oracle Linux Manager to obtain Oracle Linux packages from ULN, Oracle recommends that you configure a separate `olN-un_arch-base` base software channel and `olN-un_arch-patch` child software channel for each Oracle Linux update, as it becomes available. This practice ensures that software channels stay small and helps to speed up channel cloning. Client systems are not upgraded across update levels unless you either change the source channel used for channel cloning or reconfigure the channels to which a client system subscribes.

The following example illustrates a typical configuration of the base and child software channels for Oracle Linux, where the base and patch channels are synchronized with ULN:

```
oraclelinux7-x86_64-base
|-- oraclelinux7-x86_64-addons
|-- oraclelinux7-x86_64-kssplice
|-- oraclelinux7-x86_64-optional
|-- oraclelinux7-x86_64-patch
|-- oraclelinux7-x86_64-spacewalk27-client
|-- oraclelinux7-x86_64-uek-r4
```

**Note:**

Software channels other than the base and patch software channels do not have to be associated with ULN.

On the Oracle Linux yum server, if the base channel for an update release is synchronized with the `_latest` channel, a configuration similar to the following is obtained:

```
oraclelinux7-u4-x86_64-base
|-- oraclelinux7-u4-x86_64-addons
|-- oraclelinux7-u4-x86_64-ksplICE
|-- oraclelinux7-u4-x86_64-optional
|-- oraclelinux7-u4-x86_64-patch
|-- oraclelinux7-u4-x86_64-spacewalk27-client
|-- oraclelinux7-u4-x86_64-uek-r4
```

Using the Oracle Linux Manager Web Interface

Figure 2-2 Software Channel Management Page

The screenshot shows the Oracle Linux Manager web interface. The top navigation bar includes 'Overview', 'Systems', 'Errata', 'Channels', 'Audit', 'Configuration', 'Schedule', 'Users', 'Admin', and 'Help'. A '0 systems selected' indicator and 'Manage' and 'Clear' buttons are also present. The left sidebar has a tree view with 'Manage Software Channels' highlighted. The main content area is titled 'Software Channel Management' and includes a '+ Create Channel' and 'Clone Channel' link. Below this, there is a message: 'The following software channels are owned by your organization. Modify an existing software channel by selecting it from the list below, or create a new software channel.' A table lists the channels with columns for 'Channel Name' and 'Packages'. The table shows four entries: 'Oracle Linux 7 (x86_64)' with 4768 packages, 'Oracle Linux 7 (x86_64) Update 3 Patches' with 773 packages, 'Spacewalk 2.7 Client for Oracle Linux 7 (x86_64)' with 26 packages, and 'Spacewalk 2.7 Server for Oracle Linux 7 (x86_64)' with 203 packages. A search filter and '25 items per page' dropdown are also visible.

Channel Name	Packages
Oracle Linux 7 (x86_64)	4768
Oracle Linux 7 (x86_64) Update 3 Patches	773
Spacewalk 2.7 Client for Oracle Linux 7 (x86_64)	26
Spacewalk 2.7 Server for Oracle Linux 7 (x86_64)	203

Go to **Channels** and select **Manage Software Channels**:

- To create a software channel:
 - Click **+ Create Channel**.
 - On the Create Software Channel page, enter channel settings in the following fields, which are the most important for the initial configuration of a channel:

Channel Name

Descriptive short name for the channel, for example, `Oracle Linux 8 (x86_64) Base`.

Channel Label

Unique label for the channel that is used by the software, for example, `oraclelinux8-x86_64`.

 **Note:**

Channel labels *must* be unique across the entire Oracle Linux Manager installation. If you have multiple child channels that use the same upstream repository, for example, if you have multiple parents, then each child channel must have its own unique label. A recommended practice is to specify the update level for each hierarchy in the label, such as `oraclelinux8-u1-x86_64`.

Parent Channel

Channel of which the channel you are configuring is a child. If this is a base channel, select **None**.

Architecture

Architecture of the systems the channel serves. If necessary, select the architecture from the pull-down list. For i386 repositories, the option is **IA-32**.

Yum Repository Checksum Type

For Oracle Linux, select **sha256**.

Channel Summary

Short, descriptive summary of the channel, for example the channel name.

This field *cannot* be left blank.

Channel Description

Long description of the channel or leave the field blank.

GPG key URL

URL of the local GPG key.

For releases of Oracle Linux before Oracle Linux 9, the URL is `file:///etc/pki/rpm-gpg/RPM-GPG-KEY`.

For Oracle Linux 9, the URL is `file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle`.

The key *must* be stored locally. Thus, for third-party repositories, you must import the GPG key into the local file system and deploy it by using provisioning or another method that is appropriate for your site.

GPG key ID**GPG key Fingerprint**

Key ID and fingerprint for the Oracle Linux release.

Consult the following table:

Release	Key ID	Key Fingerprint
Oracle Linux 7	EC551F03	4214 4123 FECF C55B 9086 313D 72F9 7B74 EC55 1F03
Oracle Linux 8	AD986DA3	76FD 3DB1 3AB6 7410 B89D B10E 8256 2EA9 AD98 6DA3

Release	Key ID	Key Fingerprint
Oracle Linux 9	8D8B756F	3E6D 826D 3FBA B389 C2F3 8E34 BC4D 06A0 8D8B 756F

3. Click **Create Channel**.

- To associate a software channel with a repository:
 - Select the channel that you want to associate with a repository.
 - On the Basic Channel Details page, select **Repositories**, select the check box of the repository, and click **Update Repositories**.
- To view a software channel, select its entry to display its details.

Alternatively, go to **Channels**, select **Software Channels** and click a channel name to view its details. You can also select a specific channel and click **Show All Child Channels** to display its child channels.

- To modify a software channel:
 - Select the channel that you want to edit.
 - On the Basic Channel Details page, modify the channel settings and click **Update Channel** to save your changes.

Note:

You cannot change the channel label after you have created the channel.

- To delete a software channel:
 - Select the channel that you want to delete.
 - On the Basic Channel Details page, click **delete software channel** and then click **Delete Channel** to confirm.

Using the `softwarechannel_create` Command

Create a software channel by using the following interactive command:

```
softwarechannel_create
```

In the interactive mode, manual input of specific information is required, as shown in bold in the following example:

```
spacecmd {SSM:0}> softwarechannel_create

Channel Name: Ksplice for Oracle Linux 7
Channel Label: oraclelinux7-x86_64-ksplice
Base Channels
-----
oraclelinux7-x86_64
oraclelinux8-x86_64

Select Parent [blank to create a base channel]: oraclelinux7-x86_64

Architecture
-----
i386-sun-solaris
```

```

ia32
ia64
ppc
sparc-sun-solaris
x86_64

Select: x86_64

Checksum type
-----
sha256
sha384
sha512

Select: sha256

GPG URL
-----

GPG URL: file:///etc/pki/rpm-gpg/RPM-GPG-KEY

GPG ID
-----

GPG ID: EC551F03

GPG Fingerprint
-----

GPG Fingerprint: 4214 4123 FECF C55B 9086 313D 72F9 7B74 EC55 1F03

```

 **Note:**

The GPG URL might vary depending on the Oracle Linux release that you're configuring.

For releases of Oracle Linux before Oracle Linux 9, the URL is `file:///etc/pki/rpm-gpg/RPM-GPG-KEY`.

For Oracle Linux 9, the URL is `file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle`.

To associate a software channel with a repository, use the `softwarechannel_addrepo` command, as shown in the following example:

```
spacecmd {SSM:0}> softwarechannel_addrepo ksplice-ol7-x86_64 "Ksplice for Oracle Linux 7"
```

To list all software channels, use the `softwarechannel_list` command.

```
spacecmd {SSM:0}> softwarechannel_list oraclelinux7*
```

To list all base (parent) software channels, use the `softwarechannel_listbasechannels` command.

```
spacecmd {SSM:0}> softwarechannel_listbasechannels
```

To list the children of a base software channel, use the `softwarechannel_listchildchannels` command.

```
spacecmd {SSM:0}> softwarechannel_listchildchannels base-channel
```

To list the systems that subscribe to a software channel, use the `softwarechannel_listsystems` command:

```
spacecmd {SSM:0}> softwarechannel_listsystems base-channel
```

To display the details of a software channel, use the `softwarechannel_details` command.

```
spacecmd {SSM:0}> softwarechannel_details channel-name
```

The following output is displayed:

The following example shows how to list the details of the `ol8_x86_64_baseos_latest` channel:

```
softwarechannel_details ol8_x86_64_baseos_latest

Label:                ol8_x86_64_baseos_latest
Name:                 Oracle Linux 8 (x86_64)
Architecture:        x86_64
Parent:
Systems Subscribed:  0
Number of Packages:  0

Summary
-----
Oracle Linux 8 (x86_64)

GPG Key:              AD986DA3
GPG Fingerprint:     76FD 3DB1 3AB6 7410 B89D B10E 8256 2EA9 AD98 6DA3
GPG URL:              file:///etc/pki/rpm-gpg/RPM-GPG-KEY

Repos
-----
Oracle Linux 8 x86_64 BaseOS Latest
```

To delete a software channel, use the `softwarechannel_delete` command. Ensure that you confirm the command as prompted.

```
spacecmd {SSM:0}> softwarechannel_delete channel-name
```

Synchronizing Software Channels

After you have configured the software channels and the associated repositories, synchronize the software, either by performing an immediate manual synchronization or by scheduling a recurring synchronization job.

At a minimum, Oracle recommends that you update the Oracle Linux latest channels daily.

The initial synchronization of the Oracle Linux channels can take several days to complete. Oracle recommends that you perform an initial manual synchronization to populate the channels, and then configure a recurring job to keep them updated.

Memory Considerations When Building Repositories

When building repository metadata, Oracle Linux Manager can fail with insufficient memory issues. This failure is caused by the default Java memory settings for the Taskomatic daemon, which are set to 512 MB of minimum RAM and 1024 MB of maximum RAM. These settings are defined in the global Oracle Linux Manager configuration file `/etc/rhn/rhn.conf`.

If this type of failure occurs, increase the maximum memory to 4096 MB by setting the following property in the configuration file:

```
taskomatic.java.maxmemory=4096
```

However, to assign a value greater than 4096 MB, follow these steps:

1. Disable the automatic memory property settings.
2. Manually add the minimum and maximum Java memory values (`-Xms` and `-Xmx`, respectively) to the `taskomatic.java.additional.1` and `taskomatic.java.additional.2` properties.

The configuration file would resemble the following example:

```
taskomatic.java.initmemory=0
taskomatic.java.maxmemory=0
taskomatic.java.additional.1=-Xms1024m
taskomatic.java.additional.2=-Xmx8192m
```

Similar memory issues can also occur in the web interface if you have big data sets, such as a large number of servers or packages. To resolve these issues, increase the Tomcat memory limits in the `/etc/sysconfig/tomcat` file. Specifically, edit the `JAVA_OPTS` environment variable and increase `-Xmx`, which is the parameter for the maximum amount of memory.

For example, `-Xmx1024m` increases the maximum size of the heap to 1 GB.

After making changes to the memory settings, restart the Oracle Linux Manager services.

```
sudo /usr/sbin/spacewalk-service restart
```

Using the Oracle Linux Manager Web Interface

Figure 2-3 Channel Repositories Page

The screenshot displays the Oracle Linux Manager web interface. The top navigation bar includes tabs for Overview, Systems, Errata, Channels, Audit, Configuration, Schedule, Users, Admin, and Help. The 'Channels' tab is selected, and the 'Manage Software Channels' sidebar option is active. The main content area shows the 'Oracle Linux 7 (x86_64)' channel details, with the 'Repositories' sub-tab selected. The 'Channel Repositories' section shows a table with one repository: 'External - Oracle Linux 7 x86_64'. Below the table, there are four checkboxes: 'Do not sync errata', 'Sync only latest packages', 'Create kickstartable tree', and 'Terminate upon any error'. A 'Sync Now' button is located at the bottom right of the repository configuration area.

To synchronize software channels:

1. Go to **Channels**, select **Manage Software Channels** and then select the required channel.
2. On the Channel Details page, select **Repositories**.
3. On the Channel Repositories page, select the *Sync* tab.
 - a. Select the following check boxes as required:

Do not sync errata

Select if you do not want to synchronize any errata that are available for the channel.

Sync only latest packages

Select if you want to synchronizes only the latest packages from the repositories.

 **Caution:**

DO NOT select this check box when synchronizing module-enabled channels such as `ol8_AppStream`. The mechanism that underlies this option is not module-aware and if used, will skip required packages.

Create kickstartable tree

Select if you want to associate a kickstart profile with the channel.

However, note that ULN and the Oracle Linux yum server do not host the boot image files that you require to create a kickstartable tree. Instead, you can obtain the files from a Oracle Linux Media Pack DVD image and make them available on a local file system. See [Provisioning Client Systems](#).

Terminate upon any error

Select to stop synchronization if an error occurs.

b. Synchronize the software channel:

- To perform an immediate manual synchronization, click **Sync Now**.
- To schedule a recurring synchronization job, select the preferred schedule and times, and then click **Schedule**.

You can specify a schedule by using Quartz format. For example, `0 30 22 ? * *`, would specify that Oracle Linux Manager should resynchronize the channel every day at 10:30 PM. Using Quartz format is the only way to schedule a synchronization several times a day. For example, `0 0 0,2,22 ? * *`, would specify that synchronization should take place at 10 PM, midnight, and 2 AM. For more information, see <https://www.quartz-scheduler.org/overview/>.

Using the `softwarechannel_syncrepos` Command

The `softwarechannel_syncrepos` command is used within a `spacecmd` session as follows:

```
spacecmd {SSM:0}> softwarechannel_syncrepos oraclelinux7-x86_64-ksplice
```

 **Caution:**

DO NOT use the `--latest` option when synchronizing module-enabled channels such as `ol8_AppStream`. The mechanism that underlies this option is not module-aware and if used, will skip required packages.

Note that the command returns immediately and does not show the status of the synchronization.

Use the `tail -f` command to view the log file, `/var/logs/rhn/reposync/channel_label.log`.

To set up a schedule for channel synchronization, use the `softwarechannel_setsyncschedule` command, for example:

```
spacecmd {SSM:0}> softwarechannel_setsyncschedule oraclelinux7-x86_64-ksplice 0 30 2 ? *
*
```

The previous command configures the `oraclelinux7-x86_64-ksplice` channel to be resynchronized once every day at 2:30 AM. Specify the schedule in Quartz format. For more information, see <https://www.quartz-scheduler.org/overview/>.

To list scheduled channel synchronizations, use the `softwarechannel_listsyncschedule`.

```
spacecmd {SSM:0}> softwarechannel_listsyncschedule
```

To remove a scheduled channel synchronization, use the `softwarechannel_removesyncschedule` command, for example:

```
spacecmd {SSM:0}> softwarechannel_removesyncschedule oraclelinux7-x86_64-uek-r3
```

▲ Caution:

DO NOT use the `--latest` option when synchronizing module-enabled channels such as `ol8_AppStream`. The mechanism that underlies this option is not module-aware and if used, will skip required packages.

Using the `spacewalk-repo-sync` Command

The `spacewalk-repo-sync` command is issued outside of a `spacecmd` session.

The command can be run manually or in a cron job. If you run the command in a cron job, include the `-q` or `--quiet` options to prevent large email messages from being sent to `root`.

To display the channel label and the URL of the repository, use the `spacewalk-repo-sync -l` together with `grep`.

```
sudo spacewalk-repo-sync -l | grep repo-keyword
```

This example applies to the `Ksplice` repository.

```
sudo spacewalk-repo-sync -l | grep ksplice

ksplice-ol7-x86_64 | uln:///ol7_x86_64_ksplice
ksplice-ol6-i386 | uln:///ol6_i386_ksplice
ksplice-ol6-x86_64 | uln:///ol6_x86_64_ksplice
```

This example applies to the `addons` repository:

```
sudo spacewalk-repo-sync -l | grep addons

oraclelinux7-x86_64-addons | https://yum.oracle.com/repo/OracleLinux/OL7/addons/x86_64/
ol8_x86_64_addons          | https://yum.oracle.com/repo/OracleLinux/OL8/addons/x86_64/
```

To synchronize a channel with either a Yum or a ULN repository, use the `-c` option.

```
sudo spacewalk-repo-sync -c channel
```

For the `Ksplice` channel, you would type:

```
sudo spacewalk-repo-sync -c ol7_x86_64_ksplice
```

To synchronize a parent channel and all of its children in one operation, Use the `-p` option.

For the Oracle Linux 7 x86_64 base channel, you would type:

```
sudo spacewalk-repo-sync -p oraclelinux7-x86_64 [--latest]
```

If you include `--latest`, the server synchronizes only the latest packages that are available:

The `--latest` option downloads the latest packages that are available at the time of synchronization. Running the command does not remove older packages from the channel. If the time interval between synchronizations is large, you might miss a particular version of a package, which can have implications for errata handling because errata are associated with specific package versions. If errata consistency is important to you, Oracle recommends that you do *not* use `--latest`, except when synchronizing a Ksplice channel, where packages are always cumulative.

 **Caution:**

DO NOT use the `--latest` option when synchronizing module-enabled channels such as `ol8_AppStream`. The mechanism that underlies this option is not module-aware and if used, will skip required packages.

For more information, see the `spacewalk-repo-sync(8)` manual page.

To reduce the overall space consumed by Ksplice Offline packages, Oracle strongly recommends using repository filters to limit downloaded packages to only those required by your client systems. You can apply a filter either on the Repository configuration in the Manage Repositories page in the web UI or by providing the `-i` or `--include` parameter with the `spacewalk-repo-sync` command line, for example:

```
sudo spacewalk-repo-sync --channel ol6_x86_64_ksplice -t uln -i uptrack-updates-  
installed kernel base version,...
```

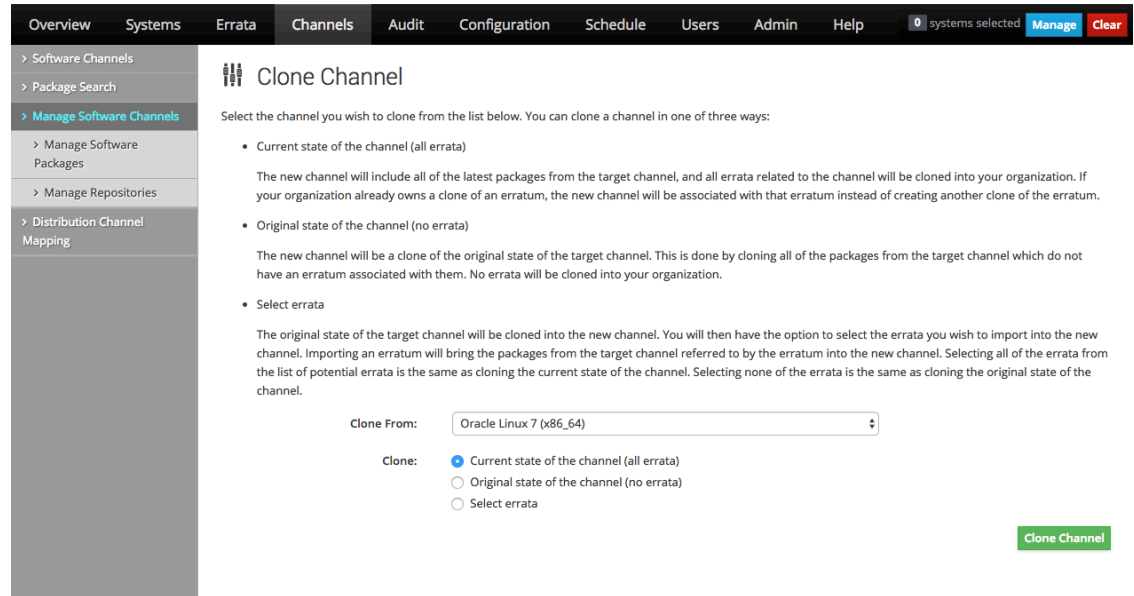
Cloning Software Channels

Clone a software channel to capture the state of its packages and errata at a given point. Cloned channels are useful for providing a stable reference base when developing and testing server systems before deployment. Cloning channels are not recommended for deployed systems, as they might be exposed to security vulnerabilities. When a channel with defined modules and streams is cloned, the software clones the `modules.yaml` file and thus ensures that the clone channel's file matches the state of the channel at the point of cloning.

Using the Oracle Linux Manager Web Interface

Use the web interface especially to clone one channel at a time.

Figure 2-4 Clone Channel Page



Clone a software channel as follows:

1. Go to **Channels** and select **Manage Software Channels**.
2. Click **Clone Channel**.
3. On the Clone Channel page, select the source channel that you want to clone from the pull-down menu, and select the clone type:

Current state of the channel (all errata)

The cloned channel includes all packages and errata from the source channel.

Original state of the channel (no errata)

The cloned channel includes all of the packages that were originally in the source channel, but no associated errata.

Select errata

The cloned channel includes all of the packages that were originally in the source channel and any errata that you select. Selecting all errata is equivalent to cloning the current state of the channel. Selecting no errata is equivalent to cloning the original state of the channel.

4. Click **Clone Channel**.
5. On the Edit Software Channel page, you can change the channel details. The default label is the source channel label prefixed with `clone-`.
6. Click **Clone Channel** again.
7. If you specified **Select errata** as the clone type, the Clone Errata page displays the available errata.

The following choices are available for each erratum:

- Merge it with the source erratum.

This default option means that the source erratum is used instead of creating a cloned copy.

- Use it to create a separate cloned erratum.
- Exclude the erratum.

Click **Clone Errata** when you have finished cloning errata.

8. On the Details page for the channel, you can also edit the channel details other than the channel label.

If you select the **Errata** tab, you can add errata from other channels or clone errata from the source channel.

If you select the **Packages** tab, you can add or remove packages from the channel.

Using the `softwarechannel_clone` Command

In a `spacecmd` session, the `softwarechannel_clone` command clones a base channel and all of its child channels at the same time. Use the command as follows:

```
spacecmd {SSM:0}> softwarechannel_clone -s ol8u1-x86_64 -x "s/$/-clone/" -o
spacecmd {SSM:0}> softwarechannel_details ol8u1-x86_64-clone
```

The `-x` option appends `-clone` to the new channel's name and label. The `-o` option excludes all errata from the cloned channel.

To compare the differences between the package contents of two channels, use the `softwarechannel_diff` command as follows:

```
spacecmd {SSM:0}> softwarechannel_diff ol8u1-x86_64-clone ol8u1-x86_64
```

To clone a base channel and all of its child channels, use the `softwarechannel_clonetree` command as follows:

```
spacecmd {SSM:0}> softwarechannel_clonetree -s ol8u1-x86_64 -p "clone-"
```

Using the `spacewalk-clone-by-date` Command

You use the `spacewalk-clone-by-date` command to clone Oracle Linux channels for a given date, which preserves the state of the channel's errata and their associated packages from its original release up to and including that date. If required, you can blacklist or remove packages, and choose which types of errata to include or exclude. The command likewise clones a base channel and all of its child channels at the same time.

The command is used outside of a `spacecmd` session.

For example, the following command clones only security errata from the `ol7-x86_64-latest` channel up to November 30, 2019 to `ol7-x86_64-latest-sec-20191130`:

```
sudo spacewalk-clone-by-date --username=swadmin --password=swpasswd --to_date=2019-11-30
--channels=ol7-x86_64-latest ol7-x86_64-latest-sec-20191130 --security_only --background
--assumeeyes
```

The command runs uninterrupted in the background. The specified spacewalk user must have Organizational Administrator or Channel Administrator privileges.

The following example clones both a base channel and a patch child channel up to November 30, 2019, excluding all versions of the `ntp` package and packages that start with `fuse`:

```
sudo spacewalk-clone-by-date --username=swadmin --password=swpasswd --channels=ol7-
x86_64-base ol7-x86_64-base-20191130 --channels=ol7-x86_64-patch ol7-x86_64-patch-20191130
--to_date=2019-11-30 --blacklist=ntp,fuse*
```

You can run `spacewalk-clone-by-date` remotely by using the `-s` option to specify the Oracle Linux Manager server's URL for XML/RPC API connections, for example `-s https://olmsvr_FQDN/rpc/api`.

A common use case is to run `spacewalk-clone-by-date` at regular intervals to keep cloned channels up to date. To generate a sample configuration file, use the following command:

```
sudo spacewalk-clone-by-date --sample-config
```

For more information, see the `spacewalk-clone-by-date(8)` manual page.

Adding Packages to Channels With Defined Modules



Note:

This functionality can only be performed by using the web interface. No command line equivalent exists.

Adding packages to modularized channels can currently be performed only by using the web interface. In this process, the software updates the target channel's `modules.yaml` file with the `modules:streams` that correspond to the packages that are being added.

Additionally, if packages are listed in each `module:stream` that are not present in the target channel, then these packages are also automatically added to the target channel. Note however, that package dependencies are not included in this automatic addition.

Source channels and the target channels must already be modular, which means that these channels must have existing `modules.yaml` files, because the software does not create `modules.yaml` by default.

1. Go to **Channels** and select **Manage Software Channels**.
2. Select the modularized channel, for example, the Applications Stream channel.
3. Click the Packages tab and then select Add Packages.
4. From the displayed list, select the packages you want to add.
5. Select Add Packages and then confirm the action.

At the end of the process, a summary of the packages that were successfully added is displayed. The list includes both the packages you manually selected and other required packages that were automatically added.

Managing Channel Life Cycles

The `spacewalk-manage-channel-lifecycle` command enables you to manage channel life cycles. It supports archiving and performing roll-backs, and is also designed for repetitive use. Thus, it is a preferred command compared to the `spacewalk-clone-by-date` command described in [Using the spacewalk-clone-by-date Command](#).

The following examples explain the processes that occur when you initialize, promote, archive, or perform roll backs of channels.

- Create a development channel `dev-ol7-x86_64-appsvr` based on the latest available packages in `ol7-x86_64-appsvr`.

```
sudo spacewalk-manage-channel-lifecycle -c ol7-x86_64-appsvr --init
```

When you run this command on channels that have defined modules on channels, the source channel's `modules.yaml` file is copied to the newly initialized channel.

- Promote the packages from the development channel to the test channel, or from the test channel to the production channel, as shown in the following two sample commands:

```
sudo spacewalk-manage-channel-lifecycle -c dev-ol7-x86_64-appsvr --promote
```

```
sudo spacewalk-manage-channel-lifecycle -c test-ol7-x86_64-appsvr --promote
```

In cases where a channel has modules and streams defined, `modulemd` stanzas from the `modules.yaml` file in the source channel are copied to the `modules.yaml` file in the target channel. However, if you use the `--clear` option when promoting a channel, then source channel's `modules.yaml` overwrites the same file in the target channel.

- Save the state of a channel by creating an archive channel called `archive-date-channel`.

```
sudo spacewalk-manage-channel-lifecycle -c prod-ol7-x86_64-appsvr --archive
```

For channels with defined modules, the `modules.yaml` file is copied from the specified channel to the new archive channel.

- Restore the state of an archived channel by performing a roll back.

```
sudo spacewalk-manage-channel-lifecycle -c archive-20191130-test-ol7-x86_64-appsvr --rollback
```

In cases where a channel has defined modules, the `modules.yaml` file is copied from the archive channel to the target channel.

To list channels, use the `-l` option.

```
sudo spacewalk-manage-channel-lifecycle -l
```

In the management of life cycles of modularized channels, you might need to increase the `XMLRPC API timeout` setting due to the intense processes involving the `modules.yaml` file. Follow these steps:

Note:

The configuration options might not be present in the following files. If they do not exist, then manually add them with the recommended settings.

- In `/etc/httpd/conf/httpd.conf`, set a new directive for `Timeout`, for example:

```
Timeout 500
```

- In `/var/lib/tomcat6/webapps/rhn/WEB-INF/web.xml`, increase Tomcat's `session_timeout` field, for example:

```
<session-config>
  <session-timeout>90</session-timeout>
</session-config>
```

- In `/etc/httpd/conf.d/zz-spacewalk-www.conf`, revise the `AJP timeout`, for example:

```
ProxyTimeout 1800
```

After making these changes, restart Oracle Linux Manager services.

```
sudo /usr/sbin/spacewalk-service start
```

3

Working With Activation Keys in Oracle Linux Manager

WARNING:

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

This chapter describes how to create and manage activation keys in Oracle Linux Manager.

About Oracle Linux Manager Activation Keys

An *activation key* enables a client system to register with Oracle Linux Manager without having to provide a user name and password. After you have configured and synchronized a base software channel and any child channels, create an activation key so that client systems can register with Oracle Linux Manager.

With the `spacewalk-common-channels -k` command syntax, you can simultaneously set up software channels and repositories and create an activation key to associate with the channels.

If required, you can also use an activation key to define the default parent and child software channel subscriptions and any configuration channels. Oracle Linux Manager subscribes a client to these channels during registration. You can change the channels any time later without changing the activation key.

Note:

Do not oversubscribe client systems to channels. Oracle recommends that you configure activation keys to subscribe a client to a minimal number of channels.

If required, you can create an activation key for each combination of base channel, system architecture, and server type. For example, you would create separate activation keys for web, mail, or application servers running on all Oracle Linux x86_64 releases.

Setting Up Activation Keys

When creating a key, use a meaningful label that is easy to understand, for example, based on the version number and architecture (`oraclelinux8-x86_64`), or based on the server type (`webserver` or `appserver`).

Oracle Linux Manager automatically prefixes the organization ID to the activation key label. For example, if you select `oraclelinux-x86_64` as the label, Oracle Linux Manager creates a key named `1-oraclelinux-x86_64`. You can create multiple activation keys for the same base channel, each with different configuration options. The name that you use is presented during Oracle Linux Manager client registration. Creating your own key labels helps you to select the correct key.

Using the Oracle Linux Manager Web Interface

Figure 3-1 Activation Keys Page

The screenshot shows the Oracle Linux Manager web interface. The top navigation bar includes tabs for Overview, Systems, Errata, Channels, Audit, Configuration, Schedule, Users, Admin, and Help. The left sidebar shows a navigation menu with options like Overview, Systems, System Groups, System Set Manager, Advanced Search, Activation Keys (highlighted), Stored Profiles, Custom System Info, Kickstart, and Software Crashes. The main content area is titled "Activation Keys" and includes a search icon and a "+ Create Key" button. Below the title, there is a section for "Universal Default" with a warning message: "You do not currently have a universal default activation key set. To set a key as the universal default, please visit the details page of that key and check off the 'Universal Default?' checkbox." Below this is a table of "All Activation Keys" with one key listed: "Oracle Linux 7 x86_64" with key "1-oraclelinux7-x86_64" and usage "1/(unlimited)". The key is selected, and there are "Select All" and "Unselect All" buttons. A tip at the bottom states: "*Tip: This key is your organization's universal default activation key." and an "Update Activation Keys" button.

Go to **Systems** and select **Activation Keys**:

- To create an activation key:
 - Click **+ Create Key**.
 - On the Create Activation Key page, enter the key settings in the following fields:

Description

Enter a description for the key, for example, `Oracle Linux 8 (x86_64)`.

Key

Enter a meaningful label for the activation key, for example, `oraclelinux8-x86_64`.

Usage

Leave this field blank to allow unlimited use by clients.

Base Channels

Select the base channel with which the key is associated, for example, Oracle Linux 8 (x86_64) Base.

Add-on Entitlements

If you are running the KVM hypervisor, enable the **Virtualization** entitlement. This entitlement installs additional packages on the target host so that Oracle Linux Manager is able to identify and manage virtual guests on that host.

Universal Default

Select if the key should be used as the default activation key for all newly-registered systems.

Note:

Oracle strongly recommends that you do not associate any channel with a universal default key. Oracle Linux Manager uses the universal default key if a key is not specified so it might be used by any version of any operating system.

3. Click **Create Activation Key** to create the activation key.
- To view an activation key, select its entry to display its details.
- To modify an activation key:
 1. Select the activation key whose settings you want to edit.
 2. On the Activation Key Details page, modify the key settings.
 3. Click **Update Activation Key** to save your changes.
- To delete an activation key:
 1. Select the activation key that you want to delete.
 2. On the Activation Key Details page, click **delete key** and then click **Delete Activation Key** to confirm.

Using the `activationkey_create` Command

Note:

Oracle recommends that you do not use automatic key generation.

When you issue the `activationkey_create` command, an interactive output is generated where you need to provide specific information, as shown in bold in the following example:

```
spacecmd {SSM:0}> activationkey_create

Name (blank to autogenerate): oraclelinux8-x86_64
Description [None]: Oracle Linux 8 (x86_64)

Base Channels
-----
oraclelinux6-x86_64
```

```
oraclelinux7-x86_64
oraclelinux8-x86_64
```

```
Base Channel (blank for default): oraclelinux8-x86_64
```

```
virtualization_host Entitlement [y/N]: N
```

```
Universal Default [y/N]: N
```

```
INFO: Created activation key 1-oraclelinux8-x86_64
```

To list all activation keys, use the `activationkey_list` command:

To display the details of an activation key, use this syntax:

```
activationkey_details key-name
```

The following example shows how the command to list activation key details and the command's output:

```
spacecmd {SSM:0}> activationkey_details 1-oraclelinux8-x86_64
```

```
Key:                1-oraclelinux8-x86_64
Description:        Oracle Linux 8 x86_64
Universal Default:  False
Usage Limit:        0
Deploy Config Channels: False
```

```
Software Channels
```

```
-----
```

```
oraclelinux8-x86_64
|-- ol8_x86_64_addons
|-- ol8_x86_64-developer_UK6R6
|-- ol8_x86_64-appstream
|-- ol8_x86_64-ksplite
```

```
Configuration Channels
```

```
-----
```

```
Entitlements
```

```
-----
```

```
System Groups
```

```
-----
```

```
Packages
```

```
-----
```

To delete an activation key, use the `activationkey_delete` command. Ensure that you confirm the command as prompted.

```
spacecmd {SSM:0}> activationkey_delete 1-oraclelinux6-x86_64
```

4

Provisioning Client Systems

WARNING:

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

This chapter describes how to provision client systems by using Kickstart profiles. With Kickstart, Oracle Linux Manager automatically installs Oracle Linux Manager Client software on any Oracle Linux server that it provisions and then registers the server as an Oracle Linux Manager client.

About Kickstart Trees, Distributions, and Profiles

To provision bare-metal systems and virtual machines by using Kickstart, create a distribution in Oracle Linux Manager for each combination of Oracle Linux release and system architecture that you want to install.

First, set up a local directory `/var/distro-trees` on your Oracle Linux Manager server that serves as the root tree for all Kickstart distributions. This directory must be readable and accessible by the `httpd` and `tomcat` services. The Kickstart tree does not need to include any packages. The packages will be eventually provided by Oracle Linux Manager.

Note:

In all examples, this document assumes that the Kickstart root tree is `/var/distro-trees`.

Store the installation kernel and initial ram-disk images in the `/var/distro-trees/kickstart-tree/images/pxeboot` subdirectory. For example, if the distribution for Oracle Linux 8 installations is `ol8-x86_64-server`, then the directory for the kernel and ram-disk images is `/var/distro-trees/ol8-x86_64-server/images/pxeboot`.

Create a Kickstart distribution by associating a Kickstart tree with existing channels. A client boots by using the Kickstart tree but installs its software packages from the existing channels.

 **Note:**

Currently, you cannot use the `spacewalk-repo-sync --sync-kickstart` command to create a Kickstart distribution from the channels that are available on the Oracle Linux yum server or on ULN.

After you have created a Kickstart distribution, you can use it with Kickstart profiles. Typically, each profile provisions a different type of server. You can configure a profile to generate a Kickstart file, or you can use an existing Kickstart file. You can associate as many profiles with a single distribution as you need to provision servers that share the same combination of Oracle Linux release and system architecture.

Setting Up Kickstart Trees

To set up a Kickstart tree on Oracle Linux Manager server, do the following:

1. Create a Kickstart tree for a distribution, for example, for Oracle Linux 8 installations.

```
sudo mkdir -p /var/distro-trees/ol8u1-x86_64-server
```

2. If SELinux is enabled in enforcing mode on your system, set the SELinux file type appropriately so that `httpd` and `tomcat` can make the files available.

- a. Define the default file type of the Kickstart tree as `httpd_sys_content_t`.

```
sudo /usr/sbin/semange fcontext -a -t httpd_sys_content_t "/var/distro-trees(/.*)?"
```

- b. Apply the file type to the entire directory hierarchy.

```
sudo /sbin/restorecon -R -v /var/distro-trees
```

 **Note:**

The `semanage` and `restorecon` commands are provided by the `policycoreutils-python` and `policycoreutils` packages.

3. Download to the Kickstart tree the full Oracle Linux Media Pack DVD image for the specific Oracle Linux release and system architecture.

The image is downloadable from the Oracle Software Delivery Cloud at <https://edelivery.oracle.com/linux>.

4. Mount the image on a suitable mount point

For example, based on the directory you created in Step 1, you would type:

```
sudo mount -o loop /var/ISOs/DVDimage.iso /var/distro-trees/ol8u1-x86_64-server
```

5. Create an entry in the `/etc/fstab` file so that the system always mounts the DVD image after a reboot.

```
/var/ISOs/V100082-01.iso /var/distro-trees/OL8u1-x86_64-server iso9660 loop,ro 0 0
```

6. If you want to associate a Kickstart tree with a software channel or to be able to boot iPXE clients, create a symbolic link from `/var/www/html` to `/var/distro-trees`.

```
sudo ln -s /var/distro-trees /var/www/html/distro-trees
```

To view the installation images, use your browser to navigate to either your Oracle Linux Manager server's or proxy's URL. For the current example, the URL would be `https://olmsvr_FQDN/distro-trees/ol8u1-x86_64-server/images`.

If you cannot see the files, do the following:

1. Open the `/etc/httpd/conf/httpd.conf` file.
2. Under the file's `<Directory "/var/www/html">` section, specify the line `Options Indexes FollowSymLinks`.

This setting enables support for directory indexing and symbolic links.

3. Reload the `httpd` service.

```
sudo service httpd reload
```

Creating Kickstart Distributions

Kickstart distributions define the path and components for specific Kickstart profiles.

Using the Oracle Linux Manager Web Interface

Figure 4-1 Kickstartable Distributions Page

The screenshot shows the Oracle Linux Manager web interface. The top navigation bar includes tabs for Overview, Systems, Errata, Channels, Audit, Configuration, Schedule, Users, Admin, and Help. The left sidebar shows a navigation menu with options like Overview, Systems, System Groups, System Set Manager, Advanced Search, Activation Keys, Stored Profiles, Custom System Info, Kickstart, Profiles, Bare Metal, GPG and SSL Keys, Distributions, and Kickstart Snippets. The main content area displays a notification 'Kickstart Distribution Created' and a section titled 'Kickstartable Distributions' with a '+ Create Distribution' button. Below this, a table lists available distributions:

Label	Base Channel	Valid*
ol7_x86_64-server	Oracle Linux 7 (x86_64)	✓

A tip at the bottom states: '*-Kickstart distributions may be invalid due wrong or non existent base path to the kernel or initrd image. To make your custom distribution valid please select the distribution and modify its base path appropriately.'

Go to **Systems**, select **Kickstart** and then **Distributions**.

- To create a distribution:
 1. Click **+ Create Distribution**.
 2. On the Create Activation Key page, enter the key settings in the following fields:

Distribution Label

Label for the distribution, for example, `ol8u1-x86_64-server`.

Tree Path

Path of the Kickstart tree for the distribution, for example, `/var/distro-trees/ol8u1-x86_64-server`.

Base Channel

Base channel with which the distribution is associated, for example, Oracle Linux 8 (x86_64) Base.

Installer Generation

Operating system release that provided the installer, for example, Red Hat Enterprise Linux 8.

Kernel Options

Options to be used when booting the installation kernel, for example, `noapic` or `text`.

Post Kernel Options

Options to be used when booting the installed system's kernel, for example, `3` or `selinux=0`.

3. Click **Create Kickstart Distribution** to create the distribution.
- To view a distribution, select its entry to display its details.
- To modify a distribution:
 1. Select the distribution whose settings you want to edit.
 2. On the Edit Kickstart Distribution page, modify the settings as required.
 3. For every Kickstart variable you want to create, modify, or delete:
 - a. Select the **Variables** tab.
 - b. On the Kickstart Variables page, define new variables or edit or delete existing variable entries.
 - c. Click **Update Variables** to save your changes.
 - d. Select the Edit tab to return to the Edit Kickstart Distribution page.
 4. Click **Update Kickstart Distribution** to save your changes.
- To delete a distribution:
 1. Select the distribution that you want to delete.
 2. On the Edit Kickstart Distribution page, click **delete distribution** and then click **Delete Distribution** to confirm.

Using the `distribution_create` Command

In a `spacecmd` session, create a Kickstart distribution as shown in the following example. Note that you would need to provide specific information on the command's interactive output, as shown in bold.

```
spacecmd {SSM:0}> distribution_create

Name: ol8u1-x86_64-server
Path to Kickstart Tree: /var/distro-trees/ol8u1-x86_64-server

Base Channels
-----
oraclelinux7-x86_64
```

```
oraclelinux8-x86_64

Base Channel: oraclelinux8-x86_64

Install Types
-----
fedora
generic_rpm
rhel_2.1
rhel_3
rhel_4
rhel_5
rhel_6
rhel_7
rhel_8
suse

Install Type: rhel_8
```

To list all activation keys, use the `distribution_list` command.

To display the details of an activation key, use the `distribution_details key` command, for example:

```
spacecmd {SSM:0}> distribution_details ol8u1-x86_64-server
```

To delete an activation key, use the `distribution_delete key` command. Remember to confirm the command's action as prompted.

Configuring Kickstart Profiles

A Kickstart configuration file contains all of the information that Kickstart requires to perform an automated installation of a server. Every Oracle Linux installation creates a Kickstart file, which is `/root/anaconda-ks.cfg`. You can use this file to repeat an installation, or you can customize the settings in this file for different system configurations. The file is also useful for troubleshooting a boot-time problem with an installed system.

Oracle Linux Manager can create a Kickstart profile that generates a Kickstart file. However, you can create a profile yourself that contains a Kickstart file that you have uploaded or copied into Oracle Linux Manager.

You can create a Kickstart profile in Oracle Linux Manager which would automatically generate a Kickstart file. However, you can also use an existing Kickstart file that you upload or copy into Spatewalk.

Use either the Oracle Linux Manager web interface or the command line to configure Kickstart profiles.

Using the Oracle Linux Manager Web Interface

Figure 4-2 Kickstart Profiles Page

The following kickstart profiles have been created for use by your organization:

Label	Active	Distribution	Spacewalk Managed?*
ol7-x86_64-minimal	✓	ol7-x86_64-server	✓

Tip: * - Kickstart profiles that are not managed by Spacewalk can not be edited in the Spacewalk UI. To modify these profiles, please log onto the Spacewalk server and run the 'cobbler profile edit' command. For more information, consult the Spacewalk Reference Guide.

Adding GPG Keys and SSL Certificates

Oracle Linux ships with the GPG key that is required for each release version. However, you must manually add GPG keys for any third-party vendors.

The Oracle GPG key that is appropriate for the installed version is stored by default at `file:///etc/pki/rpm-gpg-RPM-GPG-KEY` on every Oracle Linux server.

1. Go to **Systems**, select **Kickstart** and then **GPG and SSL Keys**.
2. Click **Create Stored Key/Cert** to display the Create GPG/SSL Key page.
3. Enter a text description of the key or certificate in the Description field.
4. Select **GPG** or **SSL**, as appropriate, from the **Type** pull-down menu.
5. Either click **Browse** and select the key or certificate file to upload or paste the file contents into the Key contents field.

Note:

GPG keys must be in ASCII, not binary, format.

6. Click **Create Key**.

Creating Profiles

Go to **Systems**, select **Kickstart** and then **Profiles**.

To create a profile that contains a Kickstart file generated by Oracle Linux Manager:

1. Click **+ Create Kickstart Profile**.
2. On Step 1, Create Kickstart Profile page, enter the profile settings in the following fields:

Label

Label for the profile, for example, `ol8u1-x86_64-minimal`.

Base Channel

Base channel with which the distribution is associated, for example, Oracle Linux 8 (x86_64) Base.

Kickstartable Tree

Kickstart distribution with which the profile is associated, for example, `ol8u1-x86_64-server`.

Virtualization Type

Type of virtualization.

For Oracle Linux installations on virtual machines that are hosted by Oracle VM or Oracle VM VirtualBox, select **None**.

For Oracle Linux 8 and Oracle Linux 7 as a Kernel-based Virtual Machine (KVM) guest, select **KVM Virtualized Guest**.

Click **Next**.

3. On Step 2, Distribution File Location page, click **Next** to accept the default download location that Oracle Linux Manager creates from the Kickstart tree.
4. On Step 3, Root Password page, enter and verify the root password for newly installed systems, and click **Finish** to create the profile.

You can now configure the Kickstart profile itself. The following steps describe the changes that are usually required to create a useable profile.

 **Note:**

At any stage, you can select **Kickstart File** to view the Kickstart file that Oracle Linux Manager would generate from the profile by using the saved configuration settings.

To create a profile that contains a Kickstart file that you upload or copy into Oracle Linux Manager, do the following:

1. Click **Upload new kickstart file**.
2. On the Kickstart Details page, enter the key settings in the following fields:

Label

Enter a label for the profile, for example, `ol6-x86_64-custom`.

Kickstartable Tree

Kickstart distribution with which the profile is associated, for example, `ol6-x86_64-server`.

Virtualization Type

Type of virtualization type. For Oracle Linux installations on virtual machines that are hosted by Oracle VM or Oracle VM VirtualBox, select **None**.

3. Do one of the following:
 - Copy and paste the contents of a Kickstart file into the **File Contents** text box.

- Click **Browse...**, then select the path of a Kickstart file and click **Upload file** to upload it to the **File Contents** text box.
4. If necessary, edit the Kickstart file contents in the **File Contents** text box.
 5. Click **Update**.

To view a profile, select its entry to display its details.

To delete a profile, do the following:

1. Select the distribution that you want to delete.
2. On the Kickstart Details page, click **delete kickstart** and then click **Delete Kickstart** to confirm.

Configuring and Modifying Profiles

1. Select a profile then select **Kickstart Details**.
 - a. On the Details page, you can do the following:
 - Edit the Kickstart label.
 - Change the virtualization type.
 - Activate or de-activate the profile.
 - Configure custom post and pre script logging.
 - Choose whether to save a copy of the Kickstart configuration to `/root` on an installed system.
 - Select an organization default profile.
 - Specify installation and post-installation kernel options.
 - Add a description of the profile.Click **Update Kickstart** to save your changes.
 - b. Select **Operating System** and select the check boxes for the child channels that you want to associate with the profile.

Note:

To enable Oracle Linux Manager to register the system automatically, select the Oracle Linux Manager Client channel.

The **Software URL** path is the virtual location where Oracle Linux Manager hosts the installation packages. It is not a real path in the file system.

Click **Update Kickstart** to save your changes.

- c. (Optional) Select **Variables** to define any Kickstart variables that you require, then click **Update Variables** to save your changes.
- d. Select **Advanced Options** to modify the Kickstart options, then click **Update Kickstart** to save your changes.

For more information about the available Kickstart options, see [Kickstart Options](#).

- e. If you intend to install bare-metal systems, select **Bare Metal Kickstart** and follow the instructions on the Bare Metal Kickstart page. This page lists the URL of the Kickstart

file that you can use to install bare-metal systems and enables you to define the IP address ranges that are associated with the profile.

2. Select **System Details** to display the Details page:

a. On the Details page, you can do the following:

- Choose the default SELinux mode for the installed system.
- Enable or disable Oracle Linux Manager configuration file management by selecting or deselecting the **Enable Oracle Linux Manager Configuration Management** check box.

For more details, see [Configuring Client Systems for Remote Management](#).

- Enable or disable Oracle Linux Manager remote commands by selecting or deselecting the **Enable Oracle Linux Manager Remote Commands** check box.

For more details, see [Configuring Client Systems for Remote Management](#).

- Choose whether to reuse an existing profile, replace the existing profile, or create a new profile but retain the existing profile.
- Change the root password for installed systems.

 **Note:**

If you make any other changes on this page, you must re-enter and verify the root password.

Click **Update System Details** to save your changes.

- b. Select **Locale**, then select the default time zone for installed systems and whether the hardware clock uses UTC, and click **Update Locale Preferences** to save your changes.

- c. Select **Partitioning** to define the partitions to be created during installation.

If you select the automatic-partitioning option `autopart` on the Advanced Options page, then clear the partitioning configuration .

Then click **Update Partitions** to save your changes.

- d. Select **GPG & SSL** to display a list of the GPG keys and SSL certificates that are known to Oracle Linux Manager. Select the keys and certificates that should be imported into the `%post` section of the Kickstart profile, then click **Update Keys** to save your selection.

For information about adding a GPG key or SSL certificate to Oracle Linux Manager, see [Adding GPG Keys and SSL Certificates](#).

3. Select **Software** to display the Package Groups page.

a. Edit the list of packages to be installed:

- For sample lists of packages, see [Sample Minimum Package Lists](#).
- The `@ Base` entry installs a minimal group of packages that are required to install a system.

If you want to specify the list of base packages explicitly, select the **Don't install @Base package group** check box.

- If you do not want the installation to halt if it cannot locate a package, select the **Ignore missing packages** check box.

- If you have associated the Oracle Linux Manager Client channel with the profile, Oracle Linux Manager installs Oracle Linux Manager Client packages automatically. You do not need to specify them in this list.
 - To apply updates and actions to a client system immediately from Oracle Linux Manager server, include the `osad` package.
- b. Click **Update Packages** to save your changes.
4. Select **Activation Keys**, then select the activation key to associate with the profile, and click **Update Activation Keys** to save your changes.

 **Note:**

A Oracle Linux Manager server activates the channels that are associated with an activation key when it registers the Oracle Linux Manager client at the end of the provisioning process. Enabling the Oracle Linux Manager Client channel by specifying the activation key is not sufficient to install the Oracle Linux Manager client software during the Kickstart process. Instead, you must specify the packages in the Kickstart profile. The channels that are available to an Oracle Linux Manager client during a Kickstart installation and the channels that are available after installation are independent. You can use channels during a Kickstart installation that are not available after installation if the activation key does not enable them.

5. Select **Scripts** to define commands that you want to run on the system before or after installation.

You can configure a preinstallation or post-installation script by using the following fields:

Scripting Language

(Optional) The path name of the script language interpreter, such as `/usr/bin/python`.

Leave blank if you want to run `bash` shell commands.

Script Name

Enter a name for the script.

Script Contents

Select the script type from the pull-down list: **Shell**, **XML**, **Ruby**, **Python**, or **perl**, and enter the script in the text area.

Script Execution Time

Select the time at which the script is executed from the pull-down list: **Pre Script** for before installation or **Post Script** for after installation.

nochroot

(Optional) Select whether the script should run outside a `chroot` jail.

erroronfail

(Optional) Select to stop the installation if an error occurs when the script runs.

Template

(Optional) Select to enable Cobbler templating for the script.

**Note:**

To apply updates and actions to a client system immediately, refer to [Configuring Client Systems for Immediate Updates](#) for configuration instructions.

Using the `kickstart_create` Command

The following example shows how to create a Kickstart profile in a `spacecmd` session.

```
spacecmd {SSM:0}> kickstart_create

Name: ol8u1-x86_64-minimal
Virtualization Types
-----
none
para_host
qemu
xenfv
xenpv

Virtualization Type [none]: none

Distributions
-----
ol8u1-x86_64-server

Select: ol8u1-x86_64-server

Root Password:
Repeat Password:
```

You can list all of the Kickstart profiles by using the `kickstart_list` command.

To display the details of a Kickstart profile, use the `kickstart_details` *kickstart-profile* command.

The following example illustrates the command use and the ensuing output.

```
spacecmd {SSM:0}> kickstart_details ol8u1-x86_64-minimal

Name:          ol8u1-x86_64-minimal
Label:         ol8u1-x86_64-minimal
Tree:          ol8u1-x86_64-server
Active:        True
Advanced:      False
Org Default:   False

Configuration Management: False
Remote Commands:          False

Software Channels
-----
ol8u1-x86_64

Advanced Options
-----
auth --enablshadow --passalgo=sha256
bootloader --location mbr
```

```
clearpart --all
firewall --disabled
keyboard us
lang en_US
network --bootproto dhcp
rootpw $5$ZdYXHxbNqu76Q5dG$.KWiOPyrGk8V5q/FEqYbWpCZdD5St387sn7jOyPH400
selinux --permissive
timezone America/New_York
url --url /var/distro-trees/ol6-x86_64-server
```

```
Software
```

```
-----
```

```
@ Base
```

```
Crypto Keys
```

```
-----
```

```
RHN-ORG-TRUSTED-SSL-CERT
```

```
Variables
```

```
-----
```

```
org = 1
```

To display the contents of the Kickstart file that a profile generates, use the `kickstart_getcontents os-profile` command.

The following example shows how to obtain the details of the Kickstart file generated by the `ol8u1-x86_64-minimal` profile:

```
spacecmd {SSM:0}> kickstart_getcontents ol8u1-x86_64-minimal

# Kickstart config file generated by Oracle Linux Manager Config Management
# Profile Label : ol8u1-x86_64-minimal
# Date Created : 2019-06-11 11:34:15.157666

install
text
network --bootproto dhcp
url --url http://olmsvr.mydom.com/ks/dist/ol8u1-x86_64-server
lang en_US
keyboard us
zerombr
clearpart --all
bootloader --location mbr
timezone America/New_York
auth --enableshadow --passalgo=sha256
rootpw --iscrypted $5$ZdYXHxbNqu76Q5dG$.KWiOPyrGk8V5q/FEqYbWpCZdD5St387sn7jOyPH400
selinux --permissive
reboot
firewall --disabled
skipx
autopart
...
```

Installing Client Systems by Using Kickstart

To install a client system from a generated Kickstart file, you can do one of the following:

- Boot the system from a real or virtual CD-ROM drive by using a boot ISO image or a full DVD image that you have downloaded from the Oracle Software Delivery Cloud at <https://edelivery.oracle.com/linux>, specifying the network location of the Kickstart file as a boot option.

This installation method is suitable for installing virtual machines or if you need to install only a small number of bare-metal systems at a local site.

- After configuring DHCP to support network booting of PXE clients and Cobbler to support the requirements of individual clients, boot the system from the network.

This installation method is suitable for installing virtual machines or if you need to install bare-metal systems at both local and remote sites.

Configuring Cobbler and DHCP to Support Network Booting

The following procedure assumes that you have configured a DHCP server on the same system as Oracle Linux Manager server. If DHCP is already provided on your network, you will need to work with your Network Administrator to configure network (PXE) boot support.

To configure Cobbler and DHCP to support the booting of client systems over the network, follow these steps:

1. Install the `cobbler-loaders` and `dhcp` packages.

```
sudo yum install cobbler-loaders dhcp
```

2. To configure Cobbler to manage the DHCP service, edit `/etc/cobbler/settings` and modify the `manage_dhcp` setting.

```
manage_dhcp: 1
```

Note:

Ensure that the file contains valid FQDNs. Names such as `localhost.localdomain` are not accepted.

3. Edit the DHCP server configuration template file (`/etc/cobbler/dhcp.template`) and change the subnet configuration for your local configuration.

The following example show how to select either the `pxelinux` boot loader for BIOS-based PXE clients or the `GRUB` boot loader for UEFI-based PXE clients:

```
# *****
# Cobbler managed dhcpd.conf file
#
# generated from cobbler dhcp.conf template ($date)
# Do NOT make changes to /etc/dhcpd.conf. Instead, make your changes
# in /etc/cobbler/dhcp.template, as /etc/dhcpd.conf will be
# overwritten.
#
# *****

ddns-update-style interim;

allow booting;
allow bootp;

ignore client-updates;
set vendorclass = option vendor-class-identifier;
option pxe-system-type code 93 = unsigned integer 16;
set pxetype = option pxe-system-type;

option domain-name "mydom.com";
```

```

subnet 192.168.1.0 netmask 255.255.255.0 {
    option domain-name-servers 192.168.1.1;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.254;
    default-lease-time 14400;
    max-lease-time 28800;
    pool {
        range 192.168.1.101 192.168.1.200;
    }
}

#for dhcp_tag in $dhcp_tags.keys():
    ## group could be subnet if your dhcp tags line up with your subnets
    ## or really any valid dhcpd.conf construct ... if you only use the
    ## default dhcp tag in Cobbler, the group block can be deleted for a
    ## flat configuration
# group for Cobbler DHCP tag: $dhcp_tag
group {
    #for mac in $dhcp_tags[$dhcp_tag].keys():
        #set iface = $dhcp_tags[$dhcp_tag][$mac]
    host $iface.name {
        hardware ethernet $mac;
        #if $iface.ip_address:
            fixed-address $iface.ip_address;
        #end if
        #if $iface.hostname:
            option host-name "$iface.hostname";
        #end if
        #if $iface.netmask:
            option subnet-mask $iface.netmask;
        #end if
        #if $iface.gateway:
            option routers $iface.gateway;
        #end if
        if substring(vendorclass, 0, 9)="PXEClient" {
            if pxetype=00:06 or pxetype=00:07 {
                filename "/grub/grub.efi";
            } else {
                filename "/pxelinux.0";
            }
        }
        ## Cobbler defaults to $next_server, but some users
        ## may like to use $iface.system.server for proxied setups
        next-server $next_server;
        ## next-server $iface.next_server;
    }
}
#end for
}
#end for

```

The previous example also configures a pool of generally available IP addresses in the range 192.168.1.101 through 192.168.1.200 on the 192.168.1/24 subnet. Systems in this pool do not boot by using PXE.

Oracle Linux Manager configures Cobbler to use TFTP to serve the boot-loader configuration files from the `/var/lib/tftpboot` directory. For more information about the format of these files, see [About Boot Loader Configuration Files](#).

If you want DHCP to support network booting of iPXE clients, see [Configuring DHCP to Support iPXE Clients](#).

4. If SELinux is enabled in enforcing mode on your system, configure SELinux for Cobbler operation as follows:

- a. Permit the `httpd` service to act as a proxy for Cobbler.

```
sudo setsebool -P httpd_can_network_connect=1
```

- b. Set the `public_content_t` file type on the `/var/lib/tftpboot` file and `/var/www/cobbler/images` directory hierarchies as follows:

```
sudo /usr/sbin/semange fcontext -a -t public_content_t "/var/lib/tftpboot/.*"
sudo /usr/sbin/semange fcontext -a -t public_content_t "/var/www/cobbler/
images/.*"
```

 **Note:**

The `semange` command is provided by the `polycoreutils-python` package.

5. Restart the `cobblerd` service:

```
sudo service cobblerd restart
```

6. Start the `httpd` service and configure it to start after a reboot.

```
sudo service httpd start
sudo chkconfig httpd on
```

 **Note:**

If you make any changes to `/etc/cobbler/dhcp.template`, run the `cobbler sync` command.

If you make any changes to `/etc/cobbler/settings`, restart the `cobblerd` service and then run the `cobbler sync` command.

7. To support booting of UEFI-based PXE clients, copy the `/boot/efi/EFI/redhat/grub.efi` file to `/var/lib/tftpboot/grub`.

```
sudo cp /boot/efi/EFI/redhat/grub.efi /var/lib/tftpboot/grub
```

8. Configure the firewall to allow access through DHCP requests.

```
sudo firewall-cmd --permanent --zone=public --remove-interface=enp0s3
sudo firewall-cmd --permanent --zone=internal --add-interface=enp0s3
sudo firewall-cmd --permanent --zone=internal --add-port=67/udp
sudo firewall-cmd --permanent --zone=internal --add-port=68/udp
sudo systemctl reload firewalld
```

In the previous example, the server expects to receive requests on interface `enp0s3` in the internal zone.

Adding a PXE Client To Be Provisioned by Oracle Linux Manager

Add a PXE client to be provisioned by Oracle Linux Manager as follows:

1. List the Kickstart profiles in Oracle Linux Manager that are usable by Cobbler.

```
sudo cobbler profile list
```

2. Use the `cobbler system add` command to define the host name, MAC address, and IP address of the target PXE client and the profile that you want to install, as shown in the following example:

```
sudo cobbler system add --name=svr1.mydom.com --hostname=svr1.mydom.com --
mac=08:00:27:c6:a1:16 --ip=92.168.1.253 --profile=ol7u7-x86_64-
server:1:OracleLinuxManagerDefaultOrganization
```

If you are provisioning a client that uses an IP address from a DHCP address pool, you might use a command such as the following:

```
sudo cobbler system add --name=devsys2 --hostname=devsys2 --profile=ol7u7-x86_64-
devsys:1:OracleLinuxManagerDefaultOrganization --kopts="ksdevice=eth0"
```

The `--kopts` option enables you to specify options to be added to the kernel boot line. In this example, `ksdevice=eth0` specifies the network interface that Kickstart should use for installation, which prevents the installation pausing to prompt you to choose which network interface to use.

3. By default, GRUB displays a boot menu for UEFI-based clients and prompts you to choose an entry. To prevent GRUB from displaying this menu, edit the `/etc/cobbler/pxe/grubsystem.template` file and add `default=0`, `hiddenmenu`, and `timeout=0` entries, as shown in the following example:

```
default=0
hiddenmenu
timeout=0

title $profile_name
    root (nd)
    kernel $kernel_path $kernel_options
    initrd $initrd_path
```

4. Run the `cobbler sync` command:

The following example shows how the command is used and the output that ensues:

```
sudo cobbler sync

task started: YYYY-MM-DD_hhmmss_sync
task started (id=Sync, time=date)
...
generating PXE configuration files
generating: /var/lib/tftpboot/pxelinux.cfg/01-08-00-27-c6-a1-16
generating: /var/lib/tftpboot/grub/01-08-00-27-c6-a1-16
rendering DHCP files
generating /etc/dhcp/dhcpd.conf
...
*** TASK COMPLETE ***
```

Cobbler creates `pxelinux` and GRUB boot configuration files for the client in the `/var/lib/tftpboot/pxelinux.cfg` and `/var/lib/tftpboot/grub` files. These files are named for the client's MAC address prefixed by `01-`, which represents the ARP hardware type for Ethernet, and use dashes to separate each byte value instead of colons. These client-specific files are based on the `/etc/cobbler/pxe/pxesystem.template` and `/etc/cobbler/pxe/grubsystem.template` files.

Cobbler also creates generic `pxelinux.cfg/default` and `grub/efidefault` boot configuration files from `/etc/cobbler/pxe/pxeprofile.template` and `/etc/cobbler/pxe/grubprofile.template`.

Cobbler adds an entry for the client to the `/etc/dhcp/dhcpd.conf` file, which is based on `/etc/cobbler/dhcp.template`, as shown in the following example:

```
$ group for Cobbler DHCP tag: default
group {
    host generic1 {
        hardware ethernet 08:00:27:c6:a1:16;
        fixed-address 192.168.1.253;
        option host-name "svr1.mydom.com";
        if substring(vendorclass, 0, 9)="PXEClient" {
            if pxetype=00:06 or pxetype=00:07 {
                filename "/grub/grub.efi";
            } else {
                filename "/pxelinux.0";
            }
        }
        next-server olmsvr.mydom.com;
    }
}
```

5. Restart the `dhcpd` service.

```
sudo systemctl restart dhcpd
```

6. Display the PXE systems that are known to Cobbler.

```
sudo cobbler system list
```

About Boot Loader Configuration Files

A boot-loader configuration file for BIOS-based PXE clients uses `pxelinux` configuration settings, as shown in the following Oracle Linux 7 example:

```
default ol7u7
prompt 0
timeout 1
label ol7u7
    kernel /images/ol7u7-x86_64:1:OracleLinuxManagerDefaultOrganization/vmlinuz
    ipappend 2
    append initrd=/images/ol7u7-x86_64:1:OracleLinuxManagerDefaultOrganization/
initrd.img \
    ksdevice=bootif lang=en_US kssendmac text \
    ks=http://192.168.1.3/cblr/svc/op/ks/system/svr1.mydom.com
```

! Important:

The `append` command and all of its arguments must be on the same line. Do *not* use the `\` line-continuation character. This character is used in the example to denote that the line has been broken for printing.

To enable the `boot: prompt` to be displayed, change the value of `prompt` to `1`. To display the prompt, press `Shift` or `Alt` at the console.

The `default` directive identifies the default boot entry by its `label` value, `ol6u6`.

Pxelinux boots the client by using the default boot entry after `timeout/10` seconds.

The `kernel` directive defines the name of the kernel executable and the `append` directive defines any parameters that should be appended when loading the kernel, such as the name of the ram-disk image and the location of the Kickstart file.

The `ipappend 2` directive specifies that the Installer should use the same network interface as the system used to boot.

For `pxelinux`, the kernel and ram-disk image file paths are relative to `/var/lib/tftpboot`. The default boot loader configuration file for `pxelinux` is `/var/lib/tftpboot/pxelinux.cfg/default`

A boot-loader configuration file for UEFI-based PXE clients uses GRUB configuration settings:

```
default=0
hiddenmenu
timeout=0

title ol7u7-x86_64-server:1:OracleLinuxManagerDefaultOrganization
    root (nd)
    kernel /images/ol7-x86_64:1:OracleLinuxManagerDefaultOrganization/vmlinuz \
        ksdevice=bootif lang=en_US kssendmac text \
        ks=http://192.168.1.3/cblr/svc/op/ks/system/svr1.mydom.com
    initrd /images/ol7-x86_64:1:OracleLinuxManagerDefaultOrganization/initrd.img
```

! Important:

The `kernel` command and all of its arguments must be on the same line. Do *not* use the `\` line-continuation character. This character is used in the example to denote that the line has been broken for printing.

Note the following additional important points about boot loader configuration:

- The `timeout=0` and `hiddenmenu` directives cause the default kernel to boot immediately without allowing you to press a key to display a menu or modify the configuration of a boot entry. The default kernel is defined as the first entry (0), which is the only entry listed in this file.
- The `root` directive defines that the kernel and initial ram-disk image files are available on the network device (`nd`), indicating that the files are available by using TFTP.
- The `kernel` directive defines the name of the kernel executable and any parameters that should be appended when loading the kernel, such as the location of the installation packages, and how to access these packages. The `initrd` directive specifies the initial ram-disk image file.
- For GRUB, the kernel and ram-disk image file paths are relative to `/var/lib/tftpboot/grub`. The default boot loader configuration file for GRUB is `/var/lib/tftpboot/grub/efidefault`.

To support different types of clients, a configuration file can be named for the following:

- Client's UUID (for example, `a8943708-c6f6-51b9-611e-74e6ac80b93d`)
- Client's MAC address prefixed by `01-`, which represents the ARP hardware type for Ethernet, and by using dashes to separate each byte value instead of colons (for example, `01-80-00-27-c6-a1-16`)

- Client's IP address expressed in hexadecimal without any leading 0x (for example, C0A801FD represents the IP address 192.168.1.253)

Cobbler writes client boot configuration files to both `/var/lib/tftpboot/grub` and `/var/lib/tftpboot/pxelinux.cfg` to handle both UEFI or BIOS-based PXE clients.

The boot loader looks for a configuration file in the following order until it finds a matching file name:

- `UUID` (for example, a8943708-c6f6-51b9-611e-74e6ac80b93d)
- `01-MAC_address` (for example, 01-80-00-27-c6-a1-16)
- Full 32 bits of the IP address (for example, C0A801FD)
- Most significant 28 bits of the IP address (for example, C0A801F)
- Most significant 24 bits of the IP address (for example, C0A801)
- Most significant 20 bits of the IP address (for example, C0A80)
- Most significant 16 bits of the IP address (for example, C0A8)
- Most significant 12 bits of the IP address (for example, C0A)
- Most significant 8 bits of the IP address (for example, C0)
- Most significant 4 bits of the IP address (for example, C)
- `default` (BIOS) or `efidefault` (EFI)

For more information about GRUB, type the `info grub` command to access the GRUB manual.

For more information about pxelinux, see <https://wiki.syslinux.org/wiki/index.php?title=PXELINUX>.

Configuring DHCP to Support iPXE Clients

iPXE extends the capabilities of PXE in many ways, including the following:

- iPXE clients can boot by using HTTP, iSCSI, AoE, and FCoE
- The boot process can be controlled by using scripts
- DNS lookup is available
- Booting across wide area networks or the Internet is possible

The `gpxelinux.0` boot loader provides some iPXE features, such as DNS lookup and HTTP file transfer, and is available in the `syslinux` package. It does not support iPXE commands or scripts.

You can use `gpxelinux.0` with BIOS-based PXE clients and with UEFI-based PXE clients in legacy mode but not in UEFI mode.

To configure the DHCP service to support iPXE clients, follow these steps:

1. Edit the DHCP server configuration template file `/etc/cobbler/dhcp.template` as follows:
 - a. Add the following lines to define the iPXE options for DHCP:

```
option space ipxe;
option ipxe-encap-opts code 175 = encapsulate ipxe;
option ipxe.priority code 1 = signed integer 8;
option ipxe.keep-san code 8 = unsigned integer 8;
```

```

option ipxe.skip-san-boot code 9 = unsigned integer 8;
option ipxe.syslogs code 85 = string;
option ipxe.cert code 91 = string;
option ipxe.privkey code 92 = string;
option ipxe.crosscert code 93 = string;
option ipxe.no-pxedhcp code 176 = unsigned integer 8;
option ipxe.bus-id code 177 = string;
option ipxe.bios-drive code 189 = unsigned integer 8;
option ipxe.username code 190 = string;
option ipxe.password code 191 = string;
option ipxe.reverse-username code 192 = string;
option ipxe.reverse-password code 193 = string;
option ipxe.version code 235 = string;
option iscsi-initiator-iqn code 203 = string;
option ipxe.pxeext code 16 = unsigned integer 8;
option ipxe.iscsi code 17 = unsigned integer 8;
option ipxe.aoe code 18 = unsigned integer 8;
option ipxe.http code 19 = unsigned integer 8;
option ipxe.https code 20 = unsigned integer 8;
option ipxe.tftp code 21 = unsigned integer 8;
option ipxe.ftp code 22 = unsigned integer 8;
option ipxe.dns code 23 = unsigned integer 8;
option ipxe.bzimage code 24 = unsigned integer 8;
option ipxe.multiboot code 25 = unsigned integer 8;
option ipxe.slam code 26 = unsigned integer 8;
option ipxe.srp code 27 = unsigned integer 8;
option ipxe.nbi code 32 = unsigned integer 8;
option ipxe.pxe code 33 = unsigned integer 8;
option ipxe.elf code 34 = unsigned integer 8;
option ipxe.comboot code 35 = unsigned integer 8;
option ipxe.efi code 36 = unsigned integer 8;
option ipxe.fcoe code 37 = unsigned integer 8;
option ipxe.vlan code 38 = unsigned integer 8;
option ipxe.menu code 39 = unsigned integer 8;
option ipxe.sdi code 40 = unsigned integer 8;
option ipxe.nfs code 41 = unsigned integer 8;

```

- b.** If you do not use a proxy DHCP server, specify the following line to speed up negotiation with the DHCP server:

```
option ipxe.no-pxedhcp 1;
```

- c.** Add the following line to define the user-class option:

```
option user-class code 77 = string;
```

- d.** Configure the DHCP server to provide the IP addresses of name servers that iPXE clients can use to resolve domain names to IP addresses:

```
option domain-name-servers 192.168.1.1, 192.168.1.4, 192.168.1.8;
```

- e.** Configure DHCP to specify the `gpxelinux.0` boot loader for non-iPXE clients and the URI of a boot script for iPXE clients:

```

if exists user-class and option user-class = "iPXE" {
    filename "http://web.mydom.com/pxeboot.ipxe";
} else {
    filename "gpxelinux.0";
}

```

In the previous example, pure iPXE clients run the HTTP-served boot script `pxeboot.ipxe`.

The following is an example of a boot script for an iPXE client:

```
#!ipxe

dhcp

kernel http://olmsvr.mydom.com/distro-trees/ol6u9-x86_64-server/images/pxeboot/
vmlinuz
initrd http://olmsvr.mydom.com/distro-trees/ol6u9-x86_64-server/images/pxeboot/
initrd.img
boot vmlinuz initrd=initrd.img ksdevice=bootif lang=en_US kssendmac text \
ks=http://192.168.1.3/cblr/svc/op/ks/profile/ol6-x86_64-
minimal:1:OracleLinuxManagerDefaultOrganization
```

`dhcp` configures the client's network interfaces.

`kernel` downloads the installation kernel.

`initrd` downloads the initial ram-disk image file.

`boot` boots the downloaded installation kernel. Boot line parameters, such as the name of the initial ram-disk file and the location of the Kickstart file, are specified as additional arguments.

! Important:

The `boot` command and all of its arguments must be on the same line. Do *not* use the `\` line-continuation character. This character is used in the example to denote that the line has been broken for printing.

For more information, see <https://ipxe.org/scripting> and <https://ipxe.org/cmd>.

Non-iPXE clients boot by using `gpxlinux.0`. A configuration file for `gpxlinux.0` is named in the same way as for `pxlinux.0` as described in [About Boot Loader Configuration Files](#). Unlike `pxlinux.0`, you can use HTTP to access the installation kernel and initial ram-disk image files.

The following is an example of a configuration file for `gpxlinux.0`:

```
prompt 0
default ol6u6
timeout 0

label ol6u9
kernel http://olmsvr.mydom.com/distro-trees/ol6u9-x86_64-server/images/pxeboot/
vmlinuz
append initrd=http://olmsvr.mydom.com/distro-trees/ol6u9-x86_64-server/images/
pxeboot/initrd.img \
ksdevice=bootif lang=en_US kssendmac text \
ks=http://192.168.1.3/cblr/svc/op/ks/profile/ol6-x86_64-
minimal:1:OracleLinuxManagerDefaultOrganization
ipappend 2
```

! Important:

The `append` command and all of its arguments must be on the same line. Do *not* use the `\` line-continuation character. This character is used in the example to denote that the line has been broken for printing.

2. Run the `cobbler sync` command:

```
sudo cobbler sync
task started: YYYY-MM-DD_hhmmss_sync
task started (id=Sync, time=date)
...
rendering DHCP files
generating /etc/dhcp/dhcpd.conf
...
*** TASK COMPLETE ***
```

The Cobbler service regenerates the `/etc/dhcp/dhcpd.conf` file and reloads the `dhcpd` service.

If you make any further changes to `/etc/cobbler/dhcp.template`, run the `cobbler sync` command. You do not need to run this command if you change the content of the boot loader configuration files.

Creating a Kickstart Profile in Cobbler

You can also create Kickstart profiles in Cobbler outside of Oracle Linux Manager. As for Oracle Linux Manager, a profile defines how to configure an installation if the target client has to perform a certain role. For example, you might want to configure a system as a web or database server. To create a profile in Cobbler, you associate a Kickstart file with a distribution.

**Note:**

Cobbler-only profiles are not visible from within Oracle Linux Manager.

You can use the `cobbler profile list` command as follows to list the profiles that are known to Cobbler:

```
sudo cobbler profile list
```

To find out which Kickstart file a profile uses, run the `cobbler profile report` command:

```
sudo cobbler profile report ol8ul-x86_64 | grep Kickstart
```

Do not use sample Kickstart files. Instead, create a Kickstart file appropriate for the profile you want to use.

To create a new profile for a distribution, follow these steps:

1. Create the Kickstart file to associate with a distribution.

For example, the following file, named `ol8ul_basic_server.ks`, contains a Kickstart definition for a basic Oracle Linux 8 server:

```
# Oracle Linux 8 Basic Server
# Use text-based installation
text
# Install using HTTP from a URL provided by Cobbler
url --url=$tree
# Define localized settings
lang en_US.UTF-8
keyboard us
timezone --utc America/New_York
# Configure network interface settings
```

```

network --onboot yes --device eth0 --bootproto dhcp --noipv6
# root password is an SHA-512 hash provided by Cobbler
rootpw --iscrypted $default_password_crypted
authconfig --enablesshadow --passalgo=sha512
# Allow only SSH connections
firewall --service=ssh
# Configure SELinux enforcing mode
selinux --enforcing
# Perform a new installation, removing all existing partitions
# before configuring the new boot loader and disk partitions
install
zerombr
clearpart --drives=sda --all --initlabel
bootloader --location=mbr --driveorder=sda --append="crashkernel=auto rhgb quiet"
autopart
# Shutdown and power off the system after installation is finished
# to allow you to change the boot order or make other changes.
poweroff
# Alternatives are halt (default), reboot, and shutdown,
# which might not be suitable for unattended installations

# Package groups and packages to be installed
%packages
...
%end

```

For sample package lists, see [Sample Minimum Package Lists](#).

Note:

This example requires that you configure an SHA-512 password hash for the `default_password_crypted` setting in `/etc/cobbler/settings`.

The previous example does not take advantage of the power of Kickstart templating and snippets for managing large numbers of profiles and systems in Cobbler. For more information, see the `cobbler(1)` manual page and <https://www.cobblerd.org/>.

2. If SELinux is enabled in `enforcing` mode on your system and you create the Kickstart file in a directory other than `/var/lib/cobbler/kickstarts`, for example `/var/kickstart`, do the following:

- a. Type the `semanage` command to define the default file type of the directory hierarchy as `cobbler_var_lib_t`:

```
sudo /usr/sbin/semanage fcontext -a -t cobbler_var_lib_t "/var/kickstart(/.*)?"
```

- b. Type the `restorecon` command to apply the file type to the entire directory hierarchy.

```
sudo /sbin/restorecon -R -v /var/kickstart
```

- c. For each Kickstart file in the directory, type the `chcon` command to set the SELinux user to `system_u`.

```
sudo chcon -u system_u /var/kickstart/*.ks
```

If SELinux is enabled in `enforcing` mode on your system, and you create the Kickstart file in `/var/lib/cobbler/kickstarts` or in a directory on which you have defined the default file type as `cobbler_var_lib_t`, type the `chcon` command to set the SELinux user of the file to `system_u`:

```
sudo chcon -u system_u ol8ul_basic_server.ks
```

You can use the `ls -Z` command to display the context:

```
sudo ls -Z ol8ul_basic_server.ks
```

The correct SELinux context for a Kickstart file used by Cobbler is `system_u:object_r:cobbler_var_lib_t:s0`.

3. Create the profile by running the `cobbler profile add` command:

```
sudo cobbler profile add --name=ol8ul_basic_server --distro=ol8ul-x86_64 \  
--kickstart=/var/lib/cobbler/kickstarts/ol8ul_basic_server.ks
```

 **Note:**

If the command returns the error `kickstart not found` for a file that does exist at the specified path, the file's SELinux context is incorrect. See the previous step for details of how to set the correct SELinux context on a file.

4. Display the profiles that are now known to Cobbler by running the `cobbler profile list` command.

```
sudo cobbler profile list
```

To remove a profile, such as `ol7-x86_64` in the example, use the `cobbler profile remove` command, as shown in the following example.

```
sudo cobbler profile remove --name=ol7-x86_64
```

You can verify the result by running `cobbler profile list`.

 **Note:**

Removing a profile also removes any client `system` definitions that you have created from that profile.

You can now define the PXE clients that Cobbler can provision based on the profile that you created. See [Adding a PXE Client To Be Provisioned by Cobbler](#).

Adding a PXE Client To Be Provisioned by Cobbler

To add a PXE client to be provisioned by Cobbler, follow these steps:

1. Define the host name, MAC address, and IP address of the target PXE client and the profile that you want to install by using the `cobbler system add` command:

```
sudo cobbler system add --name=svr1 --hostname=svr1 --mac=08:00:27:c6:a1:16 --  
ip=10.0.0.253 --profile=ol9ul_basic_server
```

If you are provisioning a desktop client that uses an IP address from a DHCP address pool, you might use a command such as the following:

```
sudo cobbler system add --name=devsys2 --hostname=devsys2 --profile=ol8ul_devsys --  
kopts="ksdevice=eth0"
```

The `--kopts` option enables you to specify options to be added to the kernel boot line. In this example, `ksdevice=eth0` specifies the network interface that Kickstart should use for installation, which prevents the installation pausing to prompt you to choose which network interface to use.

2. By default, GRUB displays a boot menu for UEFI-based clients and prompts you to choose an entry. To prevent GRUB from displaying this menu, edit `/etc/cobbler/pxe/grubsystem.template` and add `default=0`, `hiddenmenu`, and `timeout=0` entries, as shown in the following example:

```
default=0
hiddenmenu
timeout=0

title $profile_name
    root (nd)
    kernel $kernel_path $kernel_options
    initrd $initrd_path
```

3. Run the `cobbler sync` command.

```
sudo cobbler sync
```

Cobbler creates `pxelinux` and GRUB boot configuration files for the client in `/var/lib/tftpboot/pxelinux.cfg` and `/var/lib/tftpboot/grub`. These files are named for the client's MAC address prefixed by `01-`, which represents the ARP hardware type for Ethernet, and use dashes to separate each byte value instead of colons. These client-specific files are based on `/etc/cobbler/pxe/pxesystem.template` and `/etc/cobbler/pxe/grubsystem.template`.

Cobbler also creates generic `pxelinux.cfg/default` and `grub/efidefault` boot configuration files from `/etc/cobbler/pxe/pxeprofile.template` and `/etc/cobbler/pxe/grubprofile.template`.

Cobbler adds an entry for the client to `/etc/dhcp/dhcpd.conf`, which is based on `/etc/cobbler/dhcp.template`, as shown in the following example:

```
# group for Cobbler DHCP tag: default
group {
    host generic1 {
        hardware ethernet 08:00:27:c6:a1:16;
        fixed-address 10.0.0.253;
        option host-name "svr1";
        if substring(vendorclass, 0, 9)="PXEClient" {
            if pxetype=00:06 or pxetype=00:07 {
                filename "/grub/grub.efi";
            } else {
                filename "/pxelinux.0";
            }
        }
        next-server 10.0.0.6;
    }
}
```

4. Display the systems that are known to Cobbler by running the `cobbler system list` command:

```
sudo cobbler system list
```

Removing a PXE Client Definition From Cobbler

To remove a PXE Client definition from Cobbler, follow these steps:

1. Display the systems that are known to Cobbler by typing the `cobbler system list` command:

```
sudo cobbler system list
```

2. Specify the name of the system that you want to remove by typing the `cobbler system remove`, as shown in the following example where `svr2` is the system that you want to remove:

```
sudo cobbler system remove --name=svr2
```

3. Run the `cobbler sync` to update the Cobbler configuration.

```
sudo cobbler sync
```

4. Verify that `svr2` has been removed as follows:

```
sudo cobbler system list
```

Configuring Kernel Based Virtual Machines on Oracle Linux Manager

On Oracle Linux Manager, you can provision Kernel-based Virtual Machines (KVMs) either as hosts or guests.

Provisioning KVM Hosts

The following procedure describes how to use Oracle Linux Manager to provision a KVM host. Before performing this procedure, ensure that you have done the following:

- You are familiar with how to set up and use activation keys and Kickstart profiles in Oracle Linux Manager and how to configure Cobbler, DHCP, and boot loaders to support network installation of client systems.
See [Working With Activation Keys in Oracle Linux Manager](#), [Configuring Kickstart Profiles](#), and [Installing Client Systems by Using Kickstart](#).
- You have set up a base channel and kickstartable tree for the Oracle Linux distribution that you want to install on the KVM host.
- The system that you configure as a KVM host must have VT-x acceleration enabled in the BIOS or UEFI firmware and be able to forward this capability to any KVM guests. Suitable systems are bare-metal systems with VT-x enabled and Oracle VM virtual machines that have been configured with this capability. Oracle VirtualBox virtual machines do not support this functionality and are not suitable.

To provision a KVM host, follow these steps:

1. In Oracle Linux Manager, create an activation key that is specific to KVM hosts on the desired release and platform.

Enter the key settings as follows:

Description

Enter a description for the key, for example, Oracle Linux 7 (x86_64) KVM host.

Key

Enter a meaningful label for the activation key, for example, kvmhost-oraclelinux7-x86_64.

Usage

Leave blank to enable unlimited use by clients.

Base Channels

Select the base channel with which the key is associated: Oracle Linux 7 (x86_64) Base.

Add-on Entitlements

If you are running the KVM hypervisor on Oracle Linux 7, you should enable the **Virtualization** entitlement. This entitlement installs additional packages on the target host so that Oracle Linux Manager is able to identify and manage virtual guests on that host.

Universal Default

Select if the key should be used as the default activation key for all newly-registered systems.

Note:

Oracle strongly recommends that you do not associate any channels with a universal default key. Oracle Linux Manager uses the universal default key if a key is not specified so it might be used by any version of any operating system.

2. In Oracle Linux Manager, create a Kickstart profile for KVM host systems on the desired platform and do the following:
 - a. Associate the activation key that you created in Step 1 with the profile.
 - b. Enter the profile settings as follows:

Label

Enter a label for the profile. for example, kvmhost-ol7u5-x86_64.

Base Channel

Select the base channel with which the distribution is associated, for example, Oracle Linux 7 Update 5 (x86_64) Base.

Kickstartable Tree

Select the Kickstart distribution with which the profile is associated, for example, ol7u5-x86_64-server.

Virtualization Type

Select the virtualization type as **None**.

- c. Configure the following software packages, which Kickstart should install on the host, in addition to the @Base and @Core packages:

Virtualization packages (required for a KVM host):

- @virtualization-hypervisor
- @virtualization-tools

Virtualization packages (recommended):

- qemu-kvm-tools (provides debugging and diagnostic utilities.)
- virt-manager (provides a graphical virtual machine manager that you can use with KVM.)
- virt-viewer (provides a graphical console client for connecting to virtual machines)

Graphical desktop packages (required to use the Virtual Machine Manager):

- @^graphical-server-environment (provides a full graphical server environment)
- @fonts
- @gnome-desktop (select alternate desktop environment such as KDE if preferred)
- @x11

Oracle Linux Manager client packages (recommended):

- rhncfg
- rhncfg-actions
- rhncfg-client

Suggested optional packages

- @input-methods (Only include if the graphical-server-environment is enabled. Otherwise, not recommended.)
 - @internet-browser (Only include if the graphical-server-environment is enabled. Otherwise, not recommended.)
 - @multimedia (Only include if the graphical-server-environment is enabled. Otherwise, not recommended.)
 - kexec-tools
 - osad (enables you to apply updates and actions to a client system immediately from Oracle Linux Manager server)
- d. In the Kickstart profile, configure any Kickstart advanced options that you require, such as keyboard, lang, or network.
- e. Set up the %pre or %post sections for any pre or post-installation configuration that you want Kickstart to perform.

For example, you can enable configuration file management and remote commands by including the rhncfg, rhncfg-actions, and rhncfg-client packages and configuring rhn-actions-control to run in the post-installation shell:

```
rhn-actions-control --enable-all
```

3. Configure Cobbler or DHCP to provide IP and TFTP settings so that the guest being installed can access the appropriate boot loader to continue the provisioning process.

4. Configure the boot-loader configuration file that the boot loader uses to locate the installation kernel, the ram-disk image, and the Kickstart file that is served by Oracle Linux Manager.
5. After setting up Oracle Linux Manager Kickstart profile, Cobbler, DHCP, and boot-loader configuration, boot the target host system from the network, which starts the installation process.

Provisioning KVM Guests

The following procedure describes how to use Oracle Linux Manager to provision KVM guests.

Before you begin, ensure that you have done the following:

- Familiarize yourself with how to set up and use activation keys and Kickstart profiles in Oracle Linux Manager and how to configure Cobbler, DHCP, and boot loaders to support network installation of client systems.

See [Working With Activation Keys in Oracle Linux Manager](#), [Configuring Kickstart Profiles](#), and [Installing Client Systems by Using Kickstart](#).

- Familiarize yourself with how to use KVM to configure a KVM guest, for example by using the graphical Virtual Machine Manager.

Note:

To use KVM to configure a KVM guest, you must install the `spacewalk-koan` package on the KVM host.

- You have set up a base channel and kickstartable tree for the Oracle Linux distribution that you want to install on the KVM guest.

To provision a KVM guest, follow these steps:

1. In Oracle Linux Manager, create an activation key that is specific to KVM hosts on the desired platform, such as Oracle Linux 7 (x86_64). Enter the key settings as follows:

Description

Enter a description for the key for example, `Oracle Linux 7 (x86_64) KVM guest`.

Key

Enter a meaningful label for the activation key, for example, `kvmguest-oraclelinux7-x86_64`.

Usage

Leave blank to enable unlimited use by clients.

Base Channels

Select the base channel with which the key is associated, for example, `Oracle Linux 7 (x86_64) Base`.

Add-on Entitlements

No add-on entitlements are required for a virtual guest. The Virtualization entitlement is only required for virtual hosts. Do not select it when creating a virtual guest Kickstart profile.

Universal Default

Select this option if the key should be used as the default activation key for all newly-registered systems.

Oracle strongly recommends that you **do not** associate any channels with a universal default key. Oracle Linux Manager uses the universal default key if a key is not specified so it might be used by any version of any operating system.

2. In Oracle Linux Manager, create a Kickstart profile for KVM host systems on the desired platform:
 - a. Associate the activation key that you created with the profile.
 - b. Enter the profile settings as follows:

Label

Enter a label for the profile, for example, `kvmguest-ol7u5-x86_64`.

Base Channel

Select the base channel with which the distribution is associated, for example, Oracle Linux 7 Update 5 (x86_64) Base.

Kickstartable Tree

Select the Kickstart distribution with which the profile is associated, for example, `ol7u5-x86_64-server`.

Virtualization Type

For a KVM guest, select the virtualization type as **KVM Virtualized Guest**. KVM supports only HVM guests.

- c. Configure the software packages that Kickstart should install on the host in addition to the `@Base` package. The intended function of the guest system determines the set of packages. However, Oracle recommends the following additional packages for a KVM guest that is also an Oracle Linux Manager client:
 - `@guest-agents` (agents used when running under a hypervisor)
 - `@guest-desktop-agents` (agents used when running as a virtualized desktop)
 - `acpid` (enables you to control the power state of the guest from the host)
 - `osad` (enables you to apply updates and actions to a client system immediately from Oracle Linux Manager server)
 - `rhncfg`
 - `rhncfg-actions`
 - `rhncfg-client`
- d. In the Kickstart profile, configure any Kickstart advanced options that you require, such as `keyboard`, `lang`, or `network`.
- e. Set up the `%pre` or `%post` sections for any pre or post-installation configuration that you want Kickstart to perform.

For example, you can enable configuration file management and remote commands by including the `rhncfg`, `rhncfg-actions`, and `rhncfg-client` packages and configuring `rhn-actions-control` to run in the post-installation shell:

```
rhn-actions-control --enable-all
```

3. If you want to install the guest by using PXE network booting and Kickstart, do the following:

- a. Configure Cobbler or DHCP to provide IP and TFTP settings so that the guest being installed can access the appropriate boot loader to continue the provisioning process.
- b. Configure the configuration file that the boot loader uses to locate the installation kernel, the ram-disk image, and the Kickstart file served by Oracle Linux Manager.
- c. After setting up Oracle Linux Manager Kickstart profile, Cobbler, DHCP, and boot-loader configuration, boot the target guest system from the network to start the installation process.

If you want to install the guest by using a network installation, use a boot image that is made available over HTTP by Oracle Linux Manager server. You can use a full ISO image, a UEK boot image, or a RHCK boot image in conjunction with Kickstart, depending on your requirements.

5

Registering Client Systems With Oracle Linux Manager

WARNING:

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

This chapter describes how to register client systems with Oracle Linux Manager. The procedures here are performed on the client system.

When registering a client system with Oracle Linux Manager, you should also create an activation key to use with the client, as described in [Working With Activation Keys in Oracle Linux Manager](#). Do not use a default activation key, also called the universal default key. Otherwise, Oracle Linux Manager does not perform channel subscription or package installation on the registered client. Oracle also recommends that, instead of a default activation key, use an activation key that is specific to the Oracle Linux release and system architecture of the client.

Oracle Linux Manager's provisioning service typically performs all Oracle Linux Manager registration. For existing or manually installed systems, use the `rhncreg_ks` command to register them with Oracle Linux Manager.

Do not register an Oracle Linux Manager server or client with ULN. Oracle recommends that you register Oracle Linux Manager server as a client of itself after you have set up the software channels. Include the Oracle Linux Manager server channel in the list of software channels to which the server is subscribed.

Note:

Starting with systems running Oracle Linux 7 Update 1 and Oracle Linux 6 Update 9, you can register these systems without first installing Oracle Linux Manager client software. Likewise, Oracle Linux 8 Update 2 hosts that are updated with the latest available packages from ULN or `yum.oracle.com` can also be registered to Oracle Linux Manager without first installing Oracle Linux Manager client software. See [Registering a Client System Only](#).

Registering a Client System by Using Kickstart

If you install a system by using a kickstart file that is generated from an Oracle Linux Manager profile, Oracle Linux Manager automatically registers the system as an Oracle Linux Manager client if the following conditions are met:

- The Oracle Linux Manager client channel must be selected on the Modify Operating System page under the profile's **Kickstart Details** tab.
- An activation key is associated with the profile on the Kickstart Details page under the profile's **Activation Keys** tab.

For Oracle Linux 8 hosts to successfully register with Oracle Linux Manager using Kickstart, ensure that the profile has access to the `ol8_AppStream` software channel.

Note:

For more information about enabling remote configuration and management of client systems, see [Configuring Client Systems for Remote Management](#).

Installing Oracle Linux Manager Client Software and Registering the Client System

Note:

To register Red Hat Enterprise Linux clients, administrators are recommended to contact Oracle Linux consulting services to ensure proper registration of the clients.

Installing on Oracle Linux 7 Hosts

1. Ensure that only the following repositories are enabled on the current host:

- UEKR5 or UEKR6
- `ol7_latest`
- `ol7_optional_latest`
- `ol7_oraclelinuxmanager210_client`

You can verify enabled repositories by running the following command:

```
sudo yum repolist
```

2. Enable access to the repositories that contain Oracle Linux Manager client and any dependent packages on the Oracle Linux yum server at <https://yum.oracle.com>.
 - a. Ensure that the system is up to date and that you have transitioned to use the modular yum repository configuration:

```
sudo yum install oraclelinux-release-el7  
sudo /usr/bin/ol_yum_configure.sh
```

- b. Install the client release package to install appropriate yum repository configuration.

```
sudo yum install oracle-linux-manager-client-release-el7
```

3. Install additional packages as follows:

```
sudo yum install rhn-client-tools rhn-check rhn-setup rhnsd m2crypto yum-rhn-plugin
```

This command replaces the existing packages and deletes any previous registration with ULN.

4. Download the CA certificate file to the client server using one of the following methods:

- Using a browser:

Download http://olmsvr_FQDN/pub/RHN-ORG-TRUSTED-SSL-CERT to the `/usr/share/rhn` directory.

- Using a command line, for example `wget`.

```
sudo wget -q -O /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT http://olmsvr_FQDN/pub/RHN-ORG-TRUSTED-SSL-CERT
```

- Install the automatically generated package when you install SSL certificates, for example:

```
sudo yum install http://olmsvr_FQDN/pub/rhn-org-trusted-ssl-cert-1.0-1.noarch.rpm
```

You might need to specify a different URL if you replaced the SSL certificates after installing and configuring Oracle Linux Manager server software.

5. Register the system with Oracle Linux Manager with the appropriate certificate.

```
sudo rhnreg_ks --sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT --serverUrl=https://olmsvr_FQDN/XMLRPC --activationkey=activation_key
```

If you need to re-register an Oracle Linux Manager client with a Oracle Linux Manager server, or if the client was previously registered to ULN, specify `--force` as an additional option.

6. Disable access to Oracle Linux Manager client repository in the Oracle Linux yum server repository configuration file.

For example, you can type the following command:

```
sudo yum remove oracle-linux-manager-client-release-el7
```

Installing on Oracle Linux 8 Hosts

This procedure upgrades Oracle Linux 8 hosts from the default `satellite-5-client` to Oracle Linux Manager 2.10 client packages

1. Ensure that only the following repositories are enabled on the current host:

- `ol8_baseos_latest`
- `ol8_appstream`

You can verify enabled repositories by running the following command:

```
sudo dnf repolist
```

2. Enable access to the repositories that contain Oracle Linux Manager client and any dependent packages on the Oracle Linux yum server at <https://yum.oracle.com>.

- a. Ensure that your system is up to date.

- b.** Install the client release package to install appropriate yum repository configuration.

```
sudo dnf install oracle-linux-manager-client-release-el8
```

- 3.** Run the following commands .

```
sudo dnf -y module disable rhn-tools satellite-5-client
sudo dnf -y --allowerase install python3-rhnlb python3-spacewalk-usix rhn-client-
tools rhn-check rhn-setup rhnsd dnf-plugin-spacewalk
sudo dnf -y remove rhnlb
```

- 4.** Download the CA certificate file to the client server using one of the following methods:

- Using a browser:

Download http://olmsvr_FQDN/pub/RHN-ORG-TRUSTED-SSL-CERT to the `/usr/share/rhn` directory.

- Using a command line, for example `wget`.

```
sudo wget -q -O /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT http://olmsvr_FQDN/pub/
RHN-ORG-TRUSTED-SSL-CERT
```

- Install the automatically generated package when you install SSL certificates, for example:

```
sudo dnf install http://olmsvr_FQDN/pub/rhn-org-trusted-ssl-cert-1.0-1.noarch.rpm
```

You might need to specify a different URL if you replaced the SSL certificates after installing and configuring Oracle Linux Manager server software.

- 5.** Register the system with Oracle Linux Manager with the appropriate certificate.

```
sudo rhnreg_ks --sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT --
serverUrl=https://olmsvr_FQDN/XMLRPC --activationkey=activation_key
```

If you need to re-register an Oracle Linux Manager client with a Oracle Linux Manager server, or if the client was previously registered to ULN, specify `--force` as an additional option.

- 6.** Disable access to Oracle Linux Manager client repository in the Oracle Linux yum server repository configuration file.

For example, you can type the following command:

```
sudo dnf remove oracle-linux-manager-client-release-el8
```

Installing on Oracle Linux 9 Hosts

This procedure upgrades Oracle Linux Manager clients on Oracle Linux 8 hosts to work on Oracle Linux 9.

- 1.** Ensure that only the following repositories are enabled on the current Oracle Linux 9 host:

- `ol9_baseos_latest`
- `ol9_appstream`
- `ol9_UEKR7`

To check which repositories are enabled on the system, type:

```
sudo dnf repolist
```

- 2.** Enable access to the repositories that contain Oracle Linux Manager client packages and any dependent packages in the Oracle Linux yum server at <https://yum.oracle.com>.

- a. Ensure that your system is up to date.

```
sudo dnf update
```

- b. Install the client release package to install the appropriate yum repository configuration.

```
sudo dnf install oracle-linux-manager-client-release-el9
```

3. Run the following command to install additional required packages.

```
sudo dnf -y --allowerase install python3-rhnlb python3-spacewalk-usix rhn-client-tools rhn-check rhn-setup rhnsd dnf-plugin-spacewalk
```

4. Download the CA certificate file to the client server using one of the following methods:

- Using a browser:

Download `http://olmsvr_FQDN/pub/RHN-ORG-TRUSTED-SSL-CERT` to the `/usr/share/rhn` directory.

- Using a command line, for example `wget`.

```
sudo wget -q -O /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT http://olmsvr_FQDN/pub/RHN-ORG-TRUSTED-SSL-CERT
```

- Install the automatically generated package when you install SSL certificates, for example:

```
sudo dnf install http://olmsvr_FQDN/pub/rhn-org-trusted-ssl-cert-1.0-1.noarch.rpm
```

You might need to specify a different URL if you replaced the SSL certificates after installing and configuring Oracle Linux Manager server software.

5. Register the system with Oracle Linux Manager with the appropriate certificate.

```
sudo rhnreg_ks --sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT --serverUrl=https://olmsvr_FQDN/XMLRPC --activationkey=activation_key
```

If you need to re-register an Oracle Linux Manager client with a Oracle Linux Manager server, or if the client was previously registered to ULN, specify `--force` as an additional option.

Installing on CentOS 7 Hosts

Prerequisites

Before performing this procedure, ensure that the following channels are available on the Oracle Linux Manager and that these channels have been recently synchronized:

- `ol7_latest`
- `ol7_UEKR6`
- `ol7_oraclelinuxmanager210_client`

Likewise, ensure that an Oracle Linux 7 Activation Key has been defined that includes these channels.

1. Import the GPG key.

```
sudo wget https://yum.oracle.com/RPM-GPG-KEY-oracle-ol7 -O /etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
sudo gpg --import /etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
```

2. Create a temporary yum repository file `/etc/yum.repos.d/ol7-temp.repo` with the following minimum required entries:

```
[ol7_latest]
name=Oracle Linux $releasever Latest ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

3. Install the `oraclelinux-release-el7` package.

```
sudo yum install oraclelinux-release-el7
```

4. Rename CentOS repository files that conflict with Oracle Linux repositories.

```
mv CentOS-Base.repo CentOS-Linux-Base.repo.disabled
```

Likewise, rename other existing CentOS repository files as needed.

5. Rename the temporary repository file.

```
mv ol7-temp.repo ol7-temp.repo.disabled
```

6. Install Oracle Linux Manager Client on the CentOS 7 host.

```
sudo yum install oracle-linux-manager-client-release-el7
sudo yum install rhn-client-tools rhn-check rhn-setup rhnsd m2crypto yum-rhn-plugin
```

7. Download the CA certificate file to the client.

Choose from one of the following methods:

- Using a browser

Download http://olmsvr_fqdn/pub/RHN-ORG-trusted-SSL-CERT to the `/usr/share/rhn` directory.

- Using a command line utility, such as `wget`.

```
sudo wget -q -O /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT http://olmsvr_FQDN/pub/RHN-ORG-TRUSTED-SSL-CERT
```

- Installing the automatically generated package when the SSL certificates were installed.

```
sudo yum install http://olmsvr_FQDN/pub/rhn-org-trusted-ssl-cert-1.0-1.noarch.rpm
```

8. Register the system.

```
sudo rhnreg_ks --sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT --
serverUrl=https://olmsvr_FQDN/XMLRPC --activationkey=activation_key
```

If you need to re-register an Oracle Linux Manager client with an Oracle Linux Manager server, or if the client was previously registered to ULN, specify `--force` as an added option to the command.

Installing on CentOS 8 Hosts

! Important:

The information in the current procedure serves only a historical purpose. CentOS Linux 8 is no longer an active community project. CentOS Linux 8 was discontinued by CentOS in December 2021 and therefore, is no longer supported.

Further, CentOS Streams 8 is not supported by Oracle.

Prerequisites

Before performing this procedure, you must ensure that the following channels are available for the Oracle Linux Manager and that these channels have been recently synchronized:

- Oracle Linux 8 BaseOS Latest
- Oracle Linux 8 Appstream
- Oracle Linux Manager 2.10 Client for Oracle Linux 8

Ensure further that the Oracle Linux 8 Activation Key has been defined that includes these channels.

1. Import the GPG key.

```
sudo wget https://yum.oracle.com/RPM-GPG-KEY-oracle-ol8 -O /etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
sudo gpg --import --import-options show-only /etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
```

2. Create a temporary yum repository file `/etc/yum.repos.d/ol8-temp.repo` with the following minimum required entries:

```
[ol8_baseos]
name=Oracle Linux $releasever Latest ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL8/baseos/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

3. Install the `oraclelinux-release-el8` package.

```
sudo dnf install oraclelinux-release-el8
```

4. Rename CentOS repository files that conflict with Oracle Linux repositories.

```
mv CentOS-Linux_Appstream.repo CentOS-Linux-AppStream.repo.disabled
```

Likewise, rename other existing CentOS repository files as needed.

5. Rename the temporary repository file.

```
mv ol8-temp.repo ol8-temp.repo.disabled
```

6. Install Oracle Linux Manager Client on the CentOS host.

```
sudo dnf install oracle-linux-manager-client-release-el8
sudo dnf remove subscription-manager
sudo dnf module disable rhn-tools satellite5-client
sudo dnf --allowerase install python3-rhnlib python3-spacewalk-usix rhn-client-
```

```
tools rhn-check rhn-setup rhnsd dnf-plugin-spacewalk osad
sudo dnf remove rhnlib
```

7. Download the CA certificate file.

```
sudo wget -q -O /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT http://olmsvr_FQDN/pub/RHN-
ORG-TRUSTED-SSL-CERT
```

8. Register the system.

```
sudo rhnreg_ks --sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT --
serverUrl=https://olmsvr_FQDN/XMLRPC --activationkey=activation_key
```

9. Enable and start the `osad` service.

```
sudo systemctl enable osad
sudo systemctl start osad
```

Registering a Client System Only

Beginning from Oracle Linux 7 Update 1, and on Oracle Linux 8 Update 3 updated with the latest packages from either ULN or `yum.oracle.com`, you can register a client prior to installing the client software. Previously, client software installation was a prerequisite to registration.

To register an Oracle Linux server as an Oracle Linux Manager client:

1. Download the CA certificate file to the client server using one of the following methods:

- Using a browser:

Download `http://olmsvr_FQDN/pub/RHN-ORG-TRUSTED-SSL-CERT` to the `/usr/share/rhn` directory.

- Using a command line, for example `wget`.

```
sudo wget -q -O /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT http://olmsvr_FQDN/pub/
RHN-ORG-TRUSTED-SSL-CERT
```

2. Register the client with Oracle Linux Manager with the appropriate certificate.

```
sudo rhnreg_ks --sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT --
serverUrl=https://olmsvr_FQDN/XMLRPC --activationkey=activation_key
```

If you need to re-register an Oracle Linux Manager client with an Oracle Linux Manager server, additionally specify the `--force` option.

To subsequently install client software, subscribe the server to the appropriate Oracle Linux Manager client software channel. Then perform the procedures in [Installing Oracle Linux Manager Client Software and Registering the Client System](#) depending on the Oracle Linux version that your system is running.

Oracle recommends that you install Oracle Linux Manager client software after registration to support all of the features provided by Oracle Linux Manager, which include provisioning and auditing.

Upgrading the Spacewalk Client to Oracle Linux Manager 2.10

After you have upgraded to the Oracle Linux Manager server, the Spacewalk client remains registered to that server and continues to have access to the server's channels. However, the Oracle Linux Manager Client channels for the Oracle Linux versions that are in use need to be added and synchronized for the client. Follow these steps:

1. On the Oracle Linux Manager server, configure the Oracle Linux Manager Client channels.

Choose one of the following methods:

Using a browser

- a. Follow the steps in [Using the Oracle Linux Manager Web Interface](#) to configure repositories. For the URL of the source repository package, specify `https://yum.oracle.com/repo/OracleLinux/OL7/oraclelinuxmanager210/client/x86_64`.
- b. Follow the steps in [Using the Oracle Linux Manager Web Interface](#) to configure software channels. For the channel name, specify `Oracle Linux 7 (x86_64) Oracle Linux Manager 2.10 Client`.

Using the `repo_create` and `softwarechannel_create` commands

- a. Follow the steps in [Using the repo_create Command](#) to configure repositories. For the URL of the source repository package, specify `https://yum.oracle.com/repo/OracleLinux/OL7/oraclelinuxmanager210/client/x86_64`.
- b. Follow the steps in [Using the softwarechannel_create Command](#) to configure software channels. For the channel name, specify `Oracle Linux 7 (x86_64) Oracle Linux Manager 2.10 Client`.

Using the `spacewalk_common_channels` command

Type the following command:

```
sudo spacewalk-common-channels -v -u swadm -p swadm_passwd -a x86_64 -k unlimited 'oraclelinuxmanager210-client'
```

For more information about this command, see [Obtaining Packages From the Oracle Linux Yum Server](#).

2. Synchronize the newly created Oracle Linux Manager Client Channel with the source repository.

For the different methods of synchronizing channels, see [Synchronizing Software Channels](#).

3. Remove the current client access to the Spacewalk 2.7 Client channel and add access to the Oracle Linux Manager 2.10 Client channel.
 - a. On the Oracle Linux Manager web interface, select **Systems**, and then **System Groups**.
 - b. Create a group and add all of your clients to the group.

If necessary, create multiple groups to which you assign specific collections or groups of clients.

- c. Use the **System Set Manager** option to do the following:
 - Unsubscribe the groups from the Spacewalk 2.7 Client channel.
 - Subscribe the groups to the Oracle Linux Manager 2.10 Client channel.
- d. Click **Alter Subscriptions**.
- e. Click **Change Subscriptions**.

Depending on the number of clients, the subscription changes might take a while to complete. Ensure that you allow enough time for the changes to be processed.

For more detailed instructions on working with groups, see [Setting Up System Groups](#).

4. Update the client and `osad` packages.

For this step, you would run the following commands:

```
sudo systemctl stop osad
sudo yum upgrade rhn-client-tools rhn-check rhn-setup rhnsd m2crypto yum-rhn-plugin
osad
sudo systemctl start osad
```

Choose one of the following methods to execute these commands:

- Run the commands in each client's terminal.
- Use Oracle Linux Manager's **Remote Management** feature to run these commands in a remote script.

For instructions, see [Running Command Scripts on Remote Clients](#).

6

Configuring Client Systems for Immediate Updates

WARNING:

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

This chapter describes how you can configure client systems to ensure that updates as well as scheduled operations are applied to the clients immediately.

By default, the `rhnsd` daemon on a client system connects to Oracle Linux Manager server every four hours to perform updates and other actions that you have scheduled. By installing the OSA daemon, you can apply updates and actions to client systems immediately from Oracle Linux Manager server.

Enabling the OSA Daemon in Oracle Linux Manager

To set up client systems to receive updates immediately, use kickstart features such as a kickstart profile or a kickstart file. Configuring either a profile or a file as instructed in the following procedures automatically starts the `osad` service at the default run level when the client system reboots after a kickstart installation is completed.

Configuring a Kickstart Profile

Use this procedure if you want Oracle Linux Manager to automatically generate the kickstart file. The steps configure the kickstart profile to install and enable the OSA daemon on a client system.

1. Log in to Oracle Linux Manager server at `https://olmsvr-FQDN`.
2. Go to **Systems**, select **Kickstart** and then **Profiles**.
3. Select the appropriate profile, and then select **Software**.
4. On the Package Groups page, include `osad` in the list of packages to install.
5. Select **System Details**.
6. On the Details page, select **Advanced Options**.
7. Enable the `services` option and add the following entry:

```
--enabled=osad
```

This entry causes Oracle Linux Manager to add the following line to the kickstart file that it generates:

```
services --enabled=osad
```

8. Click **Update Kickstart Distribution** to save your changes.

Configuring a Kickstart File

Use this procedure if you prefer to upload an existing kickstart file instead of using an autogenerated file from Oracle Linux Manager. In this case you manually configure the kickstart file as follows:

- Include the `osad` service among the packages to be installed.
- Add the following entry that enables the `osad` service:

```
services --enabled=osad
```

Enabling the OSA Daemon Manually

Use this procedure if you want to install and enable the OSA daemon independently of Oracle Linux Manager and the use of kickstart. Instead, you issue appropriate commands.

1. Manually install the `osad` package:

- On an Oracle Linux 7 host:

```
sudo yum install -y osad
```

- On an Oracle Linux 8 or Oracle Linux 9 host:

```
sudo dnf install -y osad
```

2. Enable and start the `osad` service:

```
sudo systemctl enable osad  
sudo systemctl start osad
```

3. Check that the `osa` daemon is online.

```
sudo systemctl status osa
```

If the `osad` service does not start but instead displays the error `SSLDisabledError`, do the following:

1. Check the client's `/etc/sysconfig/rhn/up2date` file and verify the following for the `serverURL` entry:

- URL uses secure HTTP protocol (`https:`)
- URL uses the fully qualified domain name of Oracle Linux Manager server or proxy, as shown in the following example:

```
serverURL=https://olmsvr.mydom.com/XMLRPC
```

2. If you made any corrections, re-register the client.

See [Registering a Client System Only](#) for instructions.

 **Note:**

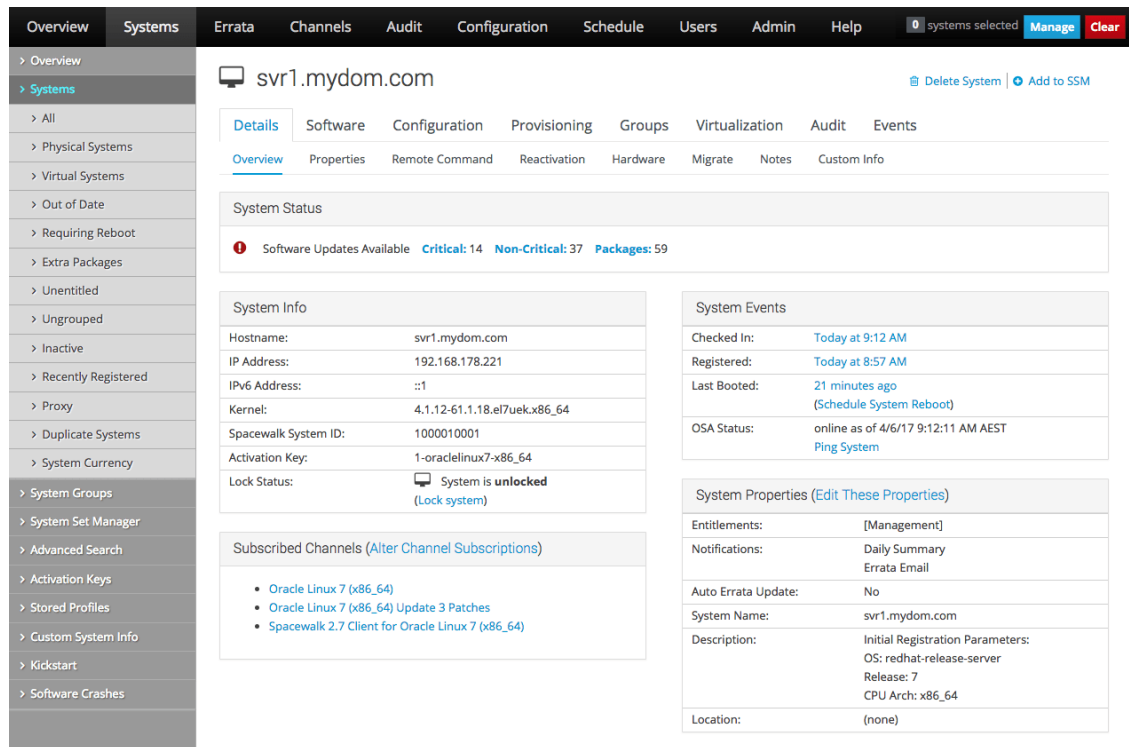
If you encounter errors that are related to `jabberd`, check if you have not yet switched to using SQLite as the default database. See [Preparing to Upgrade in Oracle Linux Manager: Installation Guide for Release 2.10](#).

Querying Status of Client Systems

Check the status of client systems by using either the Oracle Linux Manager web interface or the `spacecmd` command.

Using the Oracle Linux Manager Web Interface

Figure 6-1 System Status Page



The screenshot displays the Oracle Linux Manager web interface for a client system named `svr1.mydom.com`. The interface includes a navigation menu on the left, a top navigation bar with tabs like Overview, Systems, Errata, Channels, Audit, Configuration, Schedule, Users, Admin, and Help. The main content area shows the system status for `svr1.mydom.com`, including a warning for software updates available (14 Critical, 37 Non-Critical, 59 Packages). Below this, there are sections for System Info (hostname, IP addresses, kernel, spacewalk system ID, activation key, lock status), System Events (checked in, registered, last booted, OSA status), and System Properties (entitlements, notifications, auto errata update, system name, description, location).

To verify the status of an active client system:

1. Go to **Systems** and select the client system from the list.

The Overview page displays the following information for the client system:

- The System Status pane shows how many critical errata updates, non-critical errata updates, and packages are available to install on the client system. Select **Critical**, **Non-Critical**, or **Packages** to view and optionally install the available errata and packages.
- The System Info pane shows the host name, IP addresses, kernel version, Oracle Linux Manager system ID, activation key, and whether the system is locked.

- The Subscribed Channels pane shows the base and child channels to which the client system is subscribed.
 - The System Events pane shows when the client system last checked in, when it was registered, when it last booted, and when the OSA daemon last started.
 - The System Properties pane shows the entitlements, notifications, automatic errata update status, system name, summary of the installed operating system, and location.
2. If the OSA status is shown as `online` as of `unknown`, select **Ping System**.
 3. Wait a few seconds and then reload the page. The status should update and display when the OSA daemon was last started.

Using the `system_details` Command

Verify the status of a client system in a `spacecmd` session as follows:

```
spacecmd {SSM:0}> system_details olmsvr.mydom.com
```

7

Configuring Client Systems for Remote Management

WARNING:

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

This chapter describes how to install packages and configure clients that enable you to manage these clients remotely.

Using the Oracle Linux Manager Web Interface

In the Oracle Linux Manager web interface, you can configure remote management either by using Kickstart features or by configuring channels.

Configuring Kickstart Features

Choose the Kickstart feature you want to configure for remote management of clients.

Enabling Remote Management in a Kickstart Profile

Use this procedure if you want Oracle Linux Manager to generate the Kickstart file that would install and configure the software for remote management.

1. Log in to Oracle Linux Manager server at `https://olmsvr-FQDN`.
2. Go to **Systems**, select **Kickstart** and then **Profiles**.
3. Select the appropriate profile, and then select **Software**.
4. On the Package Groups page, include `rhncfg`, `rhncfg-actions`, and `rhncfg-client` in the list of packages to install.
5. Select **System Details**.
6. On the Details page, select the **Enable Oracle Linux Manager Configuration Management** and **Enable Oracle Linux Manager Remote Commands** check boxes.
7. Click **Update Kickstart Distribution** to save your changes.

Enabling Remote Management in a Kickstart File

Use this procedure if you want to use an existing Kickstart file instead of a file autogenerated by Oracle Linux Manager. In this case, you must manually configure the Kickstart file as follows:

- Include `rhncfg`, `rhncfg-actions`, and `rhncfg-client` among the packages to be installed.
- Add the following entry that directs `rhn-actions-control` to automatically run on the client system as part of the post-installation shell:

```
%post
rhn-actions-control --enable-all
%end
```

For more information, see the `rhn-actions-control(8)` man page.

Configuring Channels

This procedure applies to existing non-managed client systems. It uses software channels in Oracle Linux Manager server to automatically install the remote configuration packages

1. Log in to Oracle Linux Manager server at `https://olmsvr-FQDN`.
2. Ensure that the channel that contains the remote management packages includes the string `rhn-tools` on its label, for example `ol8-spacewalk210-client-rhn-tools`.
3. Enable the software channel that contains the `rhncfg`, `rhncfg-actions`, and `rhncfg-client` packages for the client. See [Updating Client Systems](#).
4. Go to **Configuration, Systems**, and then **Target Systems**.
5. On the Target Systems page, select the client systems from the systems that are listed and click **Enable Spacewalk Configuration Management**.
6. Configure configuration management by running the following commands on each client system:

- a. Check for any queued pending actions.

```
sudo rhn_check
```

- b. Enable all remote actions.

```
sudo rhn-actions-control --enable-all
```

- c. Optionally, display the list of remote actions that have been enabled.

```
sudo rhn-actions-control --report
```

The following example shows output the command might generate:

```
sudo rhn-actions-control --report
```

```
deploy is enabled
diff is enabled
upload is enabled
mtime_upload is enabled
run is enabled
```

For more information, see the `rhn-actions-control(8)` man page.

Using the `rhn-actions-control` Command

This procedure applies if you want to install and enable the OSA daemon independently of Oracle Linux Manager. Perform the steps on the target client system

1. Manually install the `rhncfg`, `rhncfg-actions`, and `rhncfg-client` packages:

- On an Oracle Linux 7 host:

```
sudo yum install rhncfg rhncfg-actions rhncfg-client
```

- On an Oracle Linux 8 host:

```
sudo dnf install rhncfg rhncfg-actions rhncfg-client
```

2. Enable the remote actions that can be performed on the client.

```
sudo rhn-actions-control --enable-all
```

3. Optionally, display the list of remote actions that have been enabled.

```
sudo rhn-actions-control --report
```

For more information, see the `rhn-action-control(8)` man page.

8

Configuring System Groups to Manage Client Systems

WARNING:

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

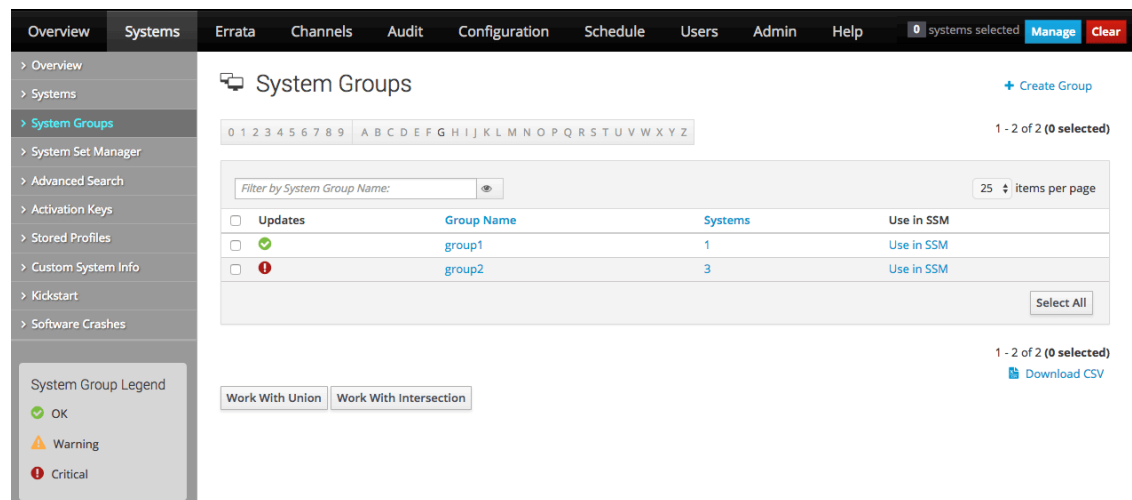
This chapter describes how to create system groups to perform the same actions on multiple client systems. Typically, a system group consists of systems that have a common installation base, architecture, and profile such as an Oracle Linux 7 (x86_64) server.

Setting Up System Groups

If you manage large numbers of systems, creating system groups, also called *system sets*, is an effective way of managing these systems. Through system groups, you apply errata, install or upgrade packages, change channel subscriptions, deploy configuration files, and reconfigure kickstart provisioning with a minimum of effort.

Using the Oracle Linux Manager Web Interface

Figure 8-1 System Groups Page



Select **Systems** and then **System Groups**:

- To create a system group:
 1. Click **Create Group**.
 2. On the Create System Group page, enter a name and description for the system group.
 3. Click **Create Group**.
- To add client systems to a system group:
 1. Click the system group name.
 2. Select the **Target Systems** tab.
 3. On the Target Systems page, select the check boxes for the systems that you want to add to the group and click **Add Systems**.
- To work with a system group:
 1. Click the system group name.
 2. On the Details page, click **Work With Group**.

Oracle Linux Manager loads the group into the **System Set Manager**.

Under **System Set Manager**, the **Selected Systems List** page displays the member systems of the system group. Any actions that you take on the tabs under System Set Manager apply only to these systems.

The **System Set Manager** option enables you to maintain a current, working system group, including adding or removing systems and system groups and so on.
- To work with the union or intersection of two or more system groups:
 1. Select the check boxes next to the system groups.
 2. Click either **Work With Union** or **Work With Intersection**.
 - **Work With Union** creates a union group that includes all member systems of the selected groups.
 - **Work With Intersection** creates an intersection group that includes only systems that are members of all of the selected groups. If no systems are members of all of the groups, the intersection group does not have any members.
 - The Selected Systems List page under System Set Manager displays the member systems of the union or intersection group. Any actions that you take on the tabs under System Set Manager apply only to these systems.
 - To save a union or intersection group as a new system group, select the Groups tab, click **+ Create New Group**, enter a name and description for the system group, and click **Create Group**.
- To remove client systems from a system group:
 1. Click the system group name.
 2. Select the **Systems** tab.
 3. On the Systems page, select the check boxes of the systems that you want to remove from the group and click **Remove Systems**.
- To delete a system group:
 1. Click the system group name.
 2. Click **delete group** and then click **Confirm Deletion**.

Using the `group_create` Command

In a `spacecmd` session, create a system group as follows:

```
spacecmd {SSM:0}> group_create group3 "Example system group 3"
```

The `group_addsystems` command enables you to perform different actions on groups, such as the following:

- Add a client system

```
spacecmd {SSM:0}> group_addsystems group3 svr1.mydom.com
```

- Add a channel

```
spacecmd {SSM:0}> group_addsystems group3 channel:ol6-x86_64
```

- Assign an IP address

```
spacecmd {SSM:0}> group_addsystems group3 ip:192.168.1
```

- Assign multiple groups to a single group

```
spacecmd {SSM:0}> group_create group4 "Example system group 4"  
spacecmd {SSM:0}> group_addsystems group4 group:group1 group:group2
```

To list system groups, use the `group_list` command.

To display the details of a system group, use the `group_details group-name` command. This command lists the members of the specified group.

To work with a system group, specify it by using `group:group_name` to a `spacecmd` command.

The following example shows how to list the system errata in `group2` and the output that might be displayed:

```
spacecmd {SSM:0}> system_listerrata group:group2
```

```
System: svr1.mydom.com
```

```
Security Errata
```

```
-----
```

```
ELSA-2017-1095 Important: bind security update 4/19/17  
ELSA-2017-0907 Moderate: util-linux security and bug fix update 4/12/17  
ELSA-2017-0906 Moderate: httpd security and bug fix update 4/12/17  
ELSA-2017-0933 Important: kernel security, bug fix, and 4/12/17  
...
```

To create an intersection of two or more system groups, you would perform the following steps:

1. Clear the contents of the system set.
2. With the `ssm_intersect` command, create the intersection as the new system set
3. Create an empty group
4. With the `group_addsystems`, specify the system set as `ssm`.

The commands to issue are as follows:

```
spacecmd {SSM:0}> ssm_clear  
spacecmd {SSM:0}> ssm_intersect group:group1 group:group2  
spacecmd {SSM:2}> group_create group5 "Example system group 5"  
spacecmd {SSM:2}> group_addsystems group5 ssm  
spacecmd {SSM:2}> ssm_clear
```

**Note:**

`{SSM:N}` shows the number of systems that are members of the system set.

To remove client systems from a system group, use the `group_removesystems system-name` command. Ensure that you confirm the command action as prompted.

To delete a system group, use the `group_delete group-name` command. Likewise, ensure that you confirm the command action as prompted.

Searching for Systems by Using the `system_search` Command

In a `spacecmd` session, search for systems by using the following syntax:

```
spacecmd {SSM:0}> system_search criterion:value
```

You can search on the following criteria:

device

System device name, for example, "xen platform device".

driver

System driver name, for example, `ata_piix`.

hostname

FQDN of the system, for example, `svr1.mydom.com`.

id

System ID in Oracle Linux Manager, for example, 1000010100.

ip

IP address, for example, 192.168.1.

name

System name in Oracle Linux Manager, for example, `svr1.mydom.com`.

uuid

System UUID, for example, 0004fb0000060000a4d43e4f737f4f5d.

vendor

System vendor name, for example, `GenuineIntel`.

For example, you would search for systems that have an IP address that contains 192.168.1 as follows:

```
spacecmd {SSM:0}> system_search ip:192.168.1
```

You can also use a search query instead of a system name with `spacecmd` commands, as shown in this example:

```
spacecmd {SSM:0}> group_addsystems group3 search:ip:192.168.1
```

To search for systems that subscribe to a software channel, use the `softwarechannel_listsystems` command:

```
spacecmd {SSM:0}> softwarechannel_listsystems ol6-x86_64
```

9

Updating Client Systems

WARNING:

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

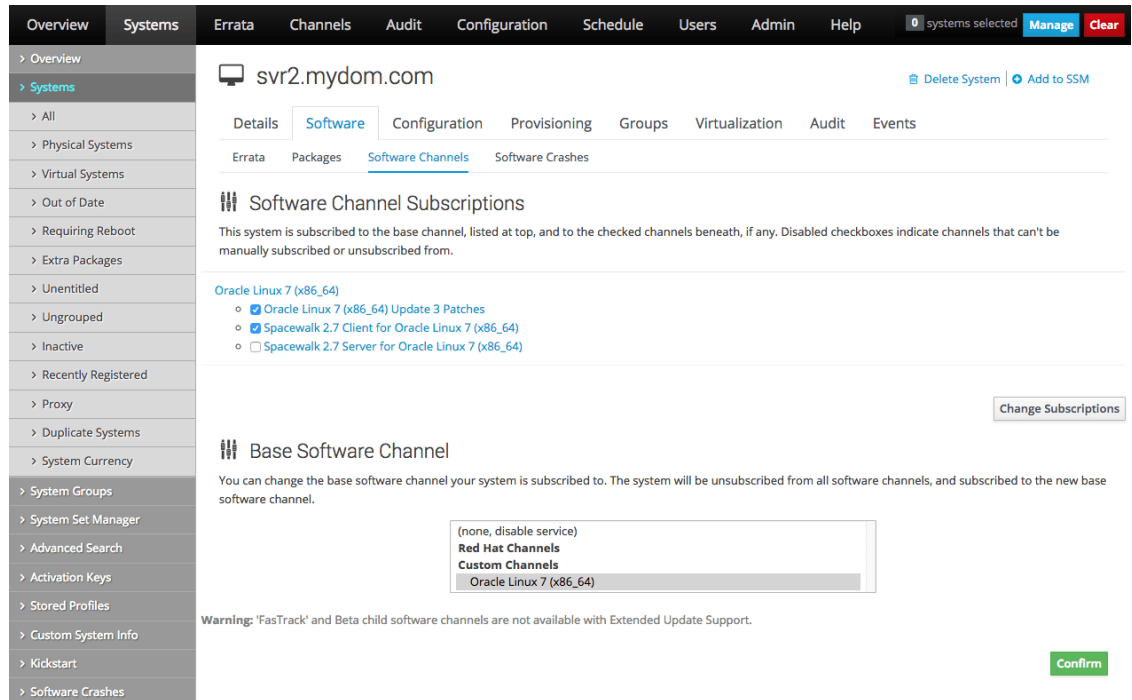
This chapter describes how to configure client systems to receive software through Oracle Linux Manager server.

Subscribing Client Systems to Software Channels

Client systems must subscribe to software channels to receive software update.

Using the Oracle Linux Manager Web Interface

Figure 9-1 Software Channel Subscriptions Page



The screenshot displays the Oracle Linux Manager web interface. The top navigation bar includes tabs for Overview, Systems, Errata, Channels, Audit, Configuration, Schedule, Users, Admin, and Help. A dropdown menu for 'Systems' is open on the left, listing various system categories. The main content area shows the 'Software Channel Subscriptions' page for the system 'svr2.mydom.com'. The page has tabs for Details, Software, Configuration, Provisioning, Groups, Virtualization, Audit, and Events. Under the 'Software Channels' tab, there is a section for 'Software Channel Subscriptions' with a description and a list of channels: 'Oracle Linux 7 (x86_64)' with sub-items 'Oracle Linux 7 (x86_64) Update 3 Patches' (checked), 'Spacewalk 2.7 Client for Oracle Linux 7 (x86_64)' (checked), and 'Spacewalk 2.7 Server for Oracle Linux 7 (x86_64)' (unchecked). Below this is a 'Base Software Channel' section with a description and a dropdown menu showing options: '(none, disable service)', 'Red Hat Channels', 'Custom Channels', and 'Oracle Linux 7 (x86_64)'. A warning message at the bottom states: 'Warning: 'FasTrack' and Beta child software channels are not available with Extended Update Support.' A 'Confirm' button is located at the bottom right.

To subscribe systems to software channels:

1. Go to **Systems** and click the system name.
2. Select **Software** and then select the **Software Channels** tab.
3. Change the child or base software channels to which a system is subscribed.
 - To change the child software channels:
 - a. In the Software Channel Subscriptions section, select or deselect the check boxes next to the child software channels to which you want to want to subscribe or unsubscribe the client.
 - b. Click **Change Subscriptions**.
 - To change the base software channel:
 - a. In the Base Software Channel section, select the new base software channel.
 - b. Click **Confirm**.
 - c. On the Confirm Base Software Channel page, click **Modify Base Software Change**.

 **Note:**

Changing the base software channel unsubscribes a system from all other software channels.

Using Commands in a spacecmd Session

To list the base and child software channels to which a system is subscribed, use the `system_listbasechannel system-name` and `system_listchildchannels system-name` commands.

To add or remove child channels, use the `system_addchildchannels` and `system_removechildchannels` commands. For both commands, a confirmation of the command action is required, as shown in bold in the following example:

```
spacecmd {SSM:0}> system_removechildchannels svr1.mydom.com ol7_x86_64_addons

Systems
-----
svr1.mydom.com

Removing Channels
-----
ol7_x86_64_addons

Is this ok [y/N]: y

spacecmd {SSM:0}> system_addchildchannels svr2.mydom.com ol7_x86_64_playground
Systems
-----
svr2.mydom.com

Adding Channels
-----
ol7_x86_64_playground
```

Is this ok [y/N]: **y**

To list the available base channels, use the `softwarechannel_listbasechannels` command.

To list the available child channels of a base channel, use the `softwarechannel_listchildchannels base-channel-name` command, for example:

```
spacecmd {SSM:0}> softwarechannel_listchildchannels oraclelinux7-x86_64
```

To change the base channel to which a system is subscribed, use the `system_setbasechannel system-name new-base-channel-name` command. A confirmation of the command action is required. The following example shows the information that is displayed when you run the command.

```
spacecmd {SSM:0}> system_setbasechannel svr5.mydom.com oraclelinux7u4-x86_64
```

```
System:          svr5.mydom.com
Old Base Channel: oraclelinux7u3-x86_64
New Base Channel: oraclelinux7u4-x86_64
```

Is this ok [y/N]: **y**



Note:

Changing the base software channel unsubscribes a system from all other software channels.

You can change the subscribed channels for multiple systems by specifying the following arguments in place of a system name. Confirmation of the command actions is required.

- `channel:channel_name` : Matches systems that are subscribed to the specified software channel.
- `group:group_name` : Specifies the systems in the named system group, for example

```
spacecmd {SSM:0}> system_removechildchannels group:group3 ol7_x86_64_playground
```

```
Systems
-----
svr1.mydom.com
svr2.mydom.com

Removing Channels
-----
ol7_x86_64_playground
```

Is this ok [y/N]: **y**

- `search:criterion:value` : Matches systems that fulfill a search criterion. See [Searching for Systems by Using the system_search Command](#).
- `ssm`: Specifies the systems that are currently in the system set.

Using the `spacewalk-channel` Command

Use the `spacewalk-channel` command to subscribe an individual client system to specific software channels. You do not need to be logged into an Oracle Linux Manager server to use this command. The same command is also used to unsubscribe individual client systems from specific software channels.

The command supports the following options:

-c *channel*, --channel=*channel*

Specify channel to which to subscribe or unsubscribe.

-a, --add

Subscribe to channel.

-r, --remove

Unsubscribe from channel.

-l, --list

List subscribed channels.

-L, --available-channels

List available channels.

-v, --verbose

Specify verbose output.

-u *user_name*, --user=*user_name*

Specify your user name.

-p *your_password*, --password=*your_password*

Specify your password.

The following examples show different usages of the `spacewalk-channel` command:

- To subscribe a client system to a specific software channel:

```
sudo spacewalk-channel --add -c ol7_x86_64_kspace -u user_name -p your_password
```

- To subscribe a client system to multiple software channels:

```
sudo spacewalk-channel --add -c ol7_x86_64_kspace -c ol6_x86_64_kspace -u  
user_name -p your_password
```

- To unsubscribe a client system from a specific software channel:

```
sudo spacewalk-channel --remove -c ol7_x86_64_kspace -u user_name -p your_password
```

- To unsubscribe a client system from multiple software channels:

```
sudo spacewalk-channel --remove -c ol7_x86_64_kspace -c ol6_x86_64_kspace -u  
user_name -p your_password
```

Applying Available Security Updates and Errata

You can make queries on available updates and errata that can be installed on client systems.

When you apply errata on client channels that have modules and streams defined, the software enforces a filtering mechanism. The filtering action ensures that only errata that are

relevant to the `modules:streams` combination are installed on the client. Oracle Linux Manager correctly installs the errata for individual systems, groups of systems, as well as system sets.

Using the Oracle Linux Manager Web Interface

Figure 9-2 Relevant Errata Page

The screenshot shows the Oracle Linux Manager web interface. The top navigation bar includes tabs for Overview, Systems, Errata, Channels, Audit, Configuration, Schedule, Users, Admin, and Help. The main content area is titled "Relevant Errata" for system "svr3.mydom.com". Below the title, there is a filter dropdown set to "All" and a "Show" button. A table of 25 errata is displayed, with columns for Type, Advisory, Synopsis, Status, and Updated. The table lists various advisories and security updates, including ELBA-2017-0839, ELBA-2017-0823, ELSA-2017-0386, and others.

Type	Advisory	Synopsis	Status	Updated
<input type="checkbox"/>	ELBA-2017-0839	tzdata enhancement update	N/A	3/23/17
<input type="checkbox"/>	ELBA-2017-0823	selinux-policy bug fix update	N/A	3/22/17
<input type="checkbox"/>	ELEA-2017-0460	nspr, nss-util, and nss bug fix and enhancement update	N/A	3/8/17
<input type="checkbox"/>	ELBA-2017-0472	tzdata bug fix and enhancement update	N/A	3/8/17
<input type="checkbox"/>	ELBA-2017-0394	lvm2 bug fix update	N/A	3/2/17
<input type="checkbox"/>	ELBA-2017-0392	polkit bug fix update	N/A	3/2/17
<input type="checkbox"/>	ELBA-2017-0371	systemd bug fix update	N/A	3/2/17
<input type="checkbox"/>	ELSA-2017-0386	Important: kernel security, bug fix, and enhancement update	N/A	3/2/17
<input type="checkbox"/>	ELBA-2017-0373	wpa_supplicant bug fix update	N/A	3/2/17
<input type="checkbox"/>	ELBA-2017-0377	NetworkManager bug fix update	N/A	3/2/17
<input type="checkbox"/>	ELBA-2017-0400	firewalld bug fix update	N/A	3/2/17
<input type="checkbox"/>	ELBA-2017-0375	selinux-policy bug fix update	N/A	3/2/17
<input type="checkbox"/>	ELBA-2017-0393	audit bug fix update	N/A	3/2/17
<input type="checkbox"/>	ELBA-2017-0374	microcode_ctl bug fix update	N/A	3/2/17
<input type="checkbox"/>	ELSA-2017-0294	Important: kernel security update	N/A	2/22/17
<input type="checkbox"/>	ELSA-2017-0286	Moderate: openssl security update	N/A	2/20/17
<input type="checkbox"/>	ELSA-2017-0276	Moderate: bind security update	N/A	2/15/17

To list the available security updates and other errata for systems or system groups:

1. For systems:

- a. Go to **Systems** and click the system name.
- b. Select **Software** and then select the **Errata** tab.

Alternatively, click **Critical** or **Non-Critical** in the System Status pane to display the Relevant Errata page with security advisory or non-critical errata selected for display.

For system groups:

- a. Go to **System Groups** and click the system group name.
- b. On the **Details** page, click **work with group**.
Oracle Linux Manager loads the group into the System Set Manager.
- c. In the System Set Manager, select the **Errata** tab.

2. On the Relevant Errata List page, select **All**, **Non-Critical**, **Bug Fix Advisory**, **Product Enhancement Advisory**, or **Security Advisory** from the pull-down list and click **Show**.

- You can filter the list on the Synopsis value or sort the list by clicking **Advisory**, **Synopsis**, **Status**, **Affected** (system groups only), or **Updated**.

- To see more details about an erratum listed under **Advisory**, select its name.
The CVEs section lists the CVEs that are fixed by an erratum. Click on a CVE name for more details.
 - To display the packages that are affected by an erratum, select the **Packages** tab.
 - To display the systems to which you can apply the erratum, select the **Affected Systems** tab.
3. Apply errata to systems or system groups:
- a. Select the check boxes for the errata that you want to apply, or click **Select All** to select all of the listed errata.
 - b. Click **Apply Errata**.
 - c. On the Relevant Errata Confirm page, change the schedule if required, and click **Confirm**.
The page updates to include a link to the scheduled action.
If you have not edited the schedule and you have enabled the OSA daemon on the client, the OSA daemon usually installs the errata packages immediately. Otherwise, `rhnsd` applies the errata when it next runs on the client.
 - d. Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the status and details of the errata update on the client.

Using the `system_applyerrata` Command

To list the security, bug fix, and product-enhancement advisory errata that are available for a client system, use the `system_listerrata` command:

To find out more details about an erratum, use the `errata_details erratum-name` command.

The following example lists the details of ELSA-2017-1095.

```
spacecmd {SSM:0}> errata_details ELSA-2017-1095

Name: ELSA-2017-1095
Product: Oracle Linux 7
Type: Security Advisory
Issue Date: 4/19/17

Topic
-----

Description
-----
[32:9.9.4-38.3] - Fix CVE-2017-3136 (ISC change 4575) - Fix
CVE-2017-3137 (ISC change 4578)

CVEs
----
CVE-2017-3136
CVE-2017-3137

Solution
-----
This update is available via the Unbreakable Linux Network (ULN) and
```

the Oracle Yum Server Server. Details on how to use ULN or `http://yum.oracle.com` to apply this update are available at `https://linux.oracle.com/applying_updates.html`.

References

`https://linux.oracle.com/errata/ELSA-2017-1095.html`

Affected Channels

`ol7-x86_64-u3-patch`

Affected Systems

3

Affected Packages

`bind-9.9.4-38.el7_3.3:32.x86_64`
`bind-chroot-9.9.4-38.el7_3.3:32.x86_64`
`bind-libs-9.9.4-38.el7_3.3:32.i686`
`bind-libs-9.9.4-38.el7_3.3:32.x86_64`
`bind-libs-lite-9.9.4-38.el7_3.3:32.i686`
`bind-libs-lite-9.9.4-38.el7_3.3:32.x86_64`
`bind-license-9.9.4-38.el7_3.3:32.noarch`
`bind-pkcs11-9.9.4-38.el7_3.3:32.x86_64`
`bind-pkcs11-libs-9.9.4-38.el7_3.3:32.i686`
`bind-pkcs11-libs-9.9.4-38.el7_3.3:32.x86_64`
`bind-pkcs11-utils-9.9.4-38.el7_3.3:32.x86_64`
`bind-utils-9.9.4-38.el7_3.3:32.x86_64`

To find the errata that fix a CVE, use the `errata_findbycve` command:

```
spacecmd {SSM:0}> errata_findbycve CVE-2017-3136
```

To list the systems to which you could apply an erratum, use the `errata_listaffectedsystems` command:

```
spacecmd {SSM:0}> errata_listaffectedsystems ELSA-2017-1095
```

To apply an erratum to a system, use the following syntax: `system_applyerrata sys` command. Confirmation of command actions is required.

```
system_applyerrata system erratum-name
```

The following example applies the command to ELSA-2017-1095 for the server `svr1.mydom.com`:

```
spacecmd {SSM:0}> system_applyerrata svr1.mydom.com ELSA-2017-1095
```

```
Start Time [now]:
```

```
Errata Systems
```

```
-----  
ELSA-2017-1095 1
```

```
Start Time: 20170421T10:01:40
```

```
Apply these errata [y/N]: y
```

```
INFO: Scheduled 1 system(s) for ELSA-2017-1095
```

You can apply errata to multiple systems by specifying the following arguments in place of a system name:

- `channel:channel_name` : Matches systems that are subscribed to the specified software channel.
- `group:group_name` : Specifies the systems in the named system group.
- `search:criterion:value` : Matches systems that match a search criterion. See [Searching for Systems by Using the system_search Command](#).
- `ssm`: Specifies the systems that are currently in the system set, for example:

```
spacecmd {SSM:0}> ssm_add svr2.mydom.com svr3.mydom.com
spacecmd {SSM:2}> system_applyerrata ssm ELSA-2017-1095
```

Managing Client Packages for Systems

Managing packages for client systems involves listing, removing, installing, and upgrading packages and all other related administrative tasks.

When you upgrade packages on channels that have modules and streams defined, the software enforces a filtering mechanism. The filtering action ensures that only those packages that are relevant to the `modules:streams` combination are installed on the client. Oracle Linux Manager correctly installs the packages for individual systems, groups of systems, as well as system sets.



Note:

The upgrade functionality applies only to packages and errata that are provided by Oracle Linux. The functionality does not work with errata from third party vendors or with those that were created by the customer.

Using the Oracle Linux Manager Web Interface

Figure 9-3 Packages Page

The screenshot shows the Oracle Linux Manager web interface. The top navigation bar includes tabs for Overview, Systems, Errata, Channels, Audit, Configuration, Schedule, Users, Admin, and Help. A status bar indicates '0 systems selected' and has 'Manage' and 'Clear' buttons. The main content area is for system 'svr3.mydom.com', with options to 'Delete System' or 'Add to SSM'. Below this, there are tabs for Details, Software (selected), Configuration, Provisioning, Groups, Virtualization, Audit, and Events. Under the Software tab, there are sub-tabs for Errata, Packages (selected), Software Channels, and Software Crashes. A row of actions is visible: List / Remove, Upgrade, Install, Verify, Profiles, and Extra Packages. The 'Packages' section contains a list of actions: List / Remove Installed Packages, Verify Files and Packages, Upgrade Packages, Install New Packages, and Compare Package Profiles / Manage Package Profiles. An 'Update Package List' button is located at the bottom right of the packages section.

To manage packages for a system:

1. Go to **Systems** and click the system name.

2. Select **Software**.
3. On the Packages page, select the tab or link for the package operation that you want to perform:

List/Remove

- a. On the Removable Packages page, select the packages that you want to remove and click **Remove Packages**.
- b. On the Confirm Package Removal page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually removes the packages immediately.

Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the details and status of the package removals.

Upgrade

- a. On the Upgradable Packages page, select the packages that you want to upgrade and click **Upgrade Packages**.
- b. On the Confirm Package Upgrade page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually upgrades the packages immediately.

Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the details and status of the package upgrades.

Install

- a. On the Installable Packages page, select the packages that you want to install and click **Install Selected Packages**.

To see more information about a package, click its name.

The Details page for the package lists any errata that include the package. To find out more information about an erratum, click its name.

The Details page for the erratum lists the CVEs that the erratum fixes. To find out more information about a CVE, click its name.

- b. On the Confirm Package Install page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually installs the packages immediately.

Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the details and status of the package installations.

Verify

- a. On the Verifiable Packages page, select the packages that you want to verify and click **Verify Selected Packages**.
- b. On the Confirm Package Verification page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually verifies the packages immediately.

Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the details and status of the package verifications.

Profiles

On the Profiles page, you can do the following:

- Create a package profile from the set of packages that are currently installed on the system:
 - a. Click **Create System Profile**.
 - b. On the Create Stored Profile page, enter a name and description for the profile and then click **Create Profile**.
- Compare the packages installed on this system with a stored package profile for this system or for another system:
 - In the Compare to Stored Profile section, select the profile name from the pull-down list and click **Compare**.
- Compare the packages installed on this system with those installed on another system:
 - In the Compare to System section, select the system name from the pull-down list and click **Compare**.

Extra Packages

The Extra Packages page displays packages that are installed on a system, but which are not present in any of the subscribed channels.

Note:

If you registered an existing system, such as the Oracle Linux Manager server itself, as a client, some of the installed packages might not be present in any subscribed channel.

If Oracle Linux Manager server is a client of itself, Oracle recommends that you synchronize Oracle Linux Manager server repository and associate it with the server so that the server receives Oracle Linux Manager server software updates.

If one or more packages should not have been installed on a system:

- a. Select the packages that you want to remove and click **Remove Packages**.

- b. On the Confirm Package Removal page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually removes the packages immediately.

Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the details and status of the package removals.

Using Commands in a spacecmd Session

Inside a `spacecmd` session, you can use various commands to manage client packages.

To create a package profile from the set of packages that are currently installed on a system, use the `system_createpackageprofile` command, for example:

```
spacecmd {SSM:0}> system_createpackageprofile svr1.mydom.com -n svr1-profile1 -d "svr1
profile 1"
```

To compare the packages installed on this system with a stored package profile for this system or for another system, use the `system_comparepackageprofile` command as follows:

```
spacecmd {SSM:0}> spacecmd {SSM:0}> system_comparepackageprofile svr2.mydom.com svr1-
profile1
```

Use the same command to compare the packages installed on this system with those installed on another system, for example:

```
spacecmd {SSM:0}> system_comparepackages svr1.mydom.com svr2.mydom.com
```

To display the details of an installable package, use the `package_details package-name` command, for example:

```
spacecmd {SSM:0}> package_details zsh
```

To install a package on a system, use the `system_installpackage system-name package-name` command. A command confirmation is required, for example:

```
spacecmd {SSM:0}> system_installpackage svr1.mydom.com zsh
```

```
svr1.mydom.com:
** Generating package cache **
zsh-4.3.10-9.el6.x86_64
```

```
Install these packages [y/N]: y
INFO: Scheduled 1 system(s)
```

Use `schedule_list` to list pending actions the system needs to perform, including package installation. Each schedule has a corresponding event ID number. To obtain details about a specific scheduled event, use the `schedule_details event-ID`

To list the package upgrades that are available for a system, use the `system_listupgrades` command:

```
spacecmd {SSM:0}> system_listupgrades svr1.mydom.com
```

To upgrade the packages on a system, use the `system_upgradepackage system-name [package-name]` command. For names of packages, you can use a wildcard (*). Note that a confirmation of the command action is required.

```
spacecmd {SSM:0}> system_upgradepackage svr1.mydom.com *
```

To remove a package from a system, use the `system_removepackage` command. Prompts for confirmation are displayed during the process.

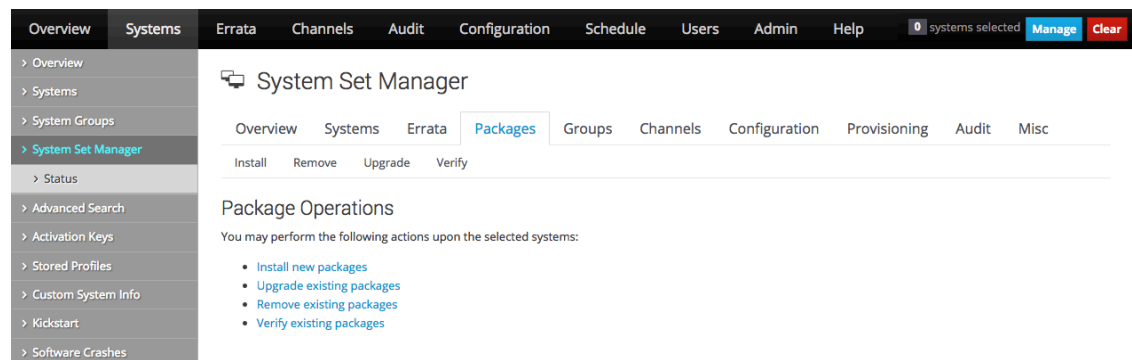
```
spacecmd {SSM:0}> system_removepackage svr1.mydom.com busybox*
```

Managing Packages for System Groups

This section describes how to manage packages for multiple systems that belong to a group.

Using the Oracle Linux Manager Web Interface

Figure 9-4 Package Operations Page



To manage packages for system groups, follow these steps:

1. Go to **System Groups** and click the system group name.
2. On the **Details** page, click **work with group**.

Oracle Linux Manager loads the group into the System Set Manager.

3. In the System Set Manager, select the **Packages** tab.
4. On the Package Operations page, select the tab or link for the package operation that you want to perform:

Install

- a. On the Select Channel page, select the channel that contains the packages that you want to install on the systems in the system group.
- b. On the Select Packages to Install page, select the packages that you want to install and click **Install Selected Packages**.

To see more information about a package, click its name.

The Details page for the package lists any errata that include the package. To find out more information about an erratum, click its name.

The Details page for the erratum lists the CVEs that the erratum fixes. To find out more information about a CVE, click its name.

- c. On the Confirm Package Install page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually installs the packages immediately.

The Tasks Log page in the System Set Manager shows the status of the package installations.

Remove

- a. On the Package Removal page, select the packages that you want to remove and click **Remove Selected Packages**.
- b. On the Confirm Package Removal page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually removes the packages immediately.

The Tasks Log page in the System Set Manager shows the status of the package removals.

Upgrade

- a. On the Select Packages to Upgrade page, select the packages that you want to upgrade and click **Upgrade Selected Packages**.
- b. On the Confirm Package Upgrade page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually upgrades the packages immediately.

The Tasks Log page in the System Set Manager shows the status of the package upgrades.

Verify

- a. On the Verifiable Packages page, select the packages that you want to verify and click **Verify Selected Packages**.
- b. On the Confirm Package Verification page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually verifies the packages immediately.

The Tasks Log page in the System Set Manager shows the status of the package verifications.

Using Commands in a spacecmd Session

Compare the packages that are installed on the systems in a system group with a stored package profile by using the `system_comparepackageprofile` command, as shown in the following example:

```
spacecmd {SSM:0}> system_comparepackageprofile group:group1 svr1-profile1
```

To install a package on the systems in a system group, use the following syntax:

```
spacecmd {SSM:0}> system_installpackage system-name
```

The command requires confirmation of the action to be performed, as shown in bold in the following example:

```
spacecmd {SSM:0}> system_installpackage svr3.mydom.com
```

```
svr3.mydom.com:
zsh-4.3.10-9.el6.x86_64

#####
```

```
svr4.mydom.com:
zsh-4.3.10-9.el6.x86_64
```

```
Install these packages [y/N]: y
INFO: Scheduled 2 system(s)
```

To list the package upgrades that are available for the systems in a system group, use the `system_listupgrades group:group-name` command:

To upgrade the packages on the systems in a system group, use the `system_upgradepackage group:group-name` command. A command action confirmation is required.

```
spacecmd {SSM:0}> system_upgradepackage group:group1 *
```

```
svr3.mydom.com:
bash-4.1.2-29.el6.0.1.x86_64
wget-1.12-5.el6_6.1.x86_64

#####
```

```
svr4.mydom.com:
wget-1.12-5.el6_6.1.x86_64
```

```
Install these packages [y/N]: y
INFO: Scheduled 2 system(s)
```

To remove a package from the systems in a system group, use the `system_removepackage` command. Ensure that you confirm the action when prompted.

```
spacecmd {SSM:0}> system_removepackage svr1.mydom.com busybox*
```

10

Controlling and Configuring Client Systems

 **WARNING:**

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

This chapter describes how you can configure Oracle Linux Manager to run command scripts on remote client systems. You can set up configuration channels, subscribe client systems to these channels, and customize client systems by using the channels to deploy configuration files.

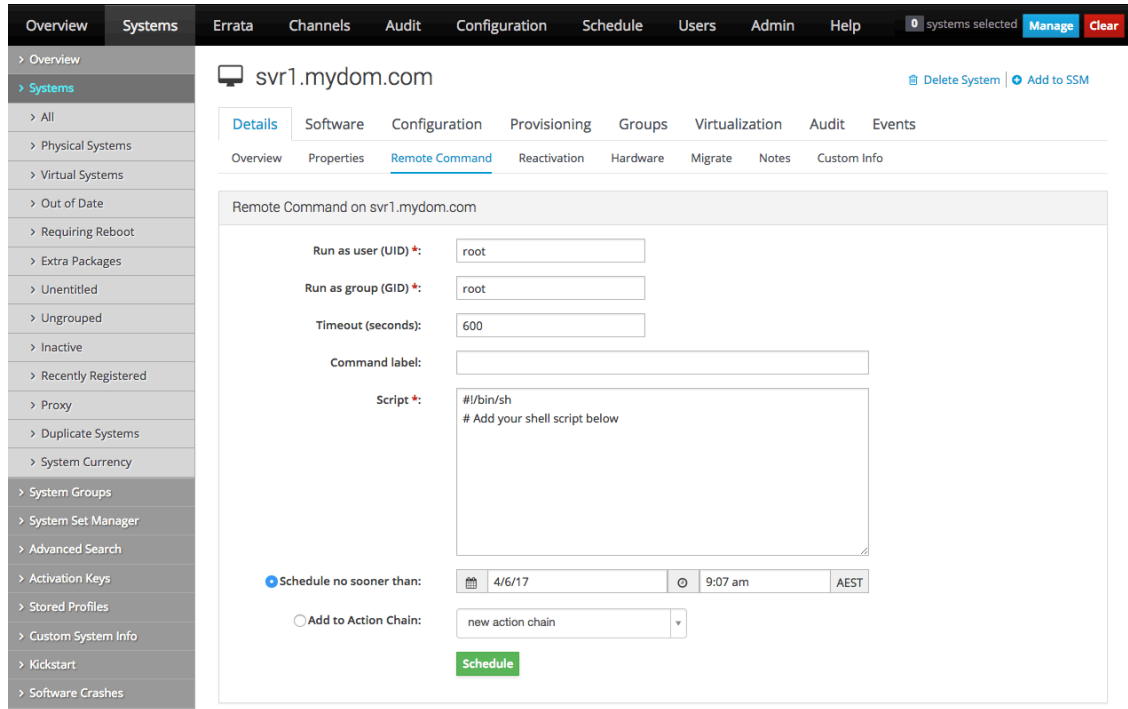
To run commands and perform tasks described in this chapter, the Oracle Linux Manager server must be permitted to run remote commands on client systems. See [Configuring Client Systems for Remote Management](#).

Running Command Scripts on Remote Clients

Use either the web interface or the `spacecmd` command to enable you to run commands and scripts remotely on client systems.

Using the Oracle Linux Manager Web Interface

Figure 10-1 Remote Command Page



Log in to Oracle Linux Manager server at `https://olmsvr-FQDN` and do the following:

1. Go to **Systems** and select the client system from the list.
2. Select **Details** and then select the **Remote Command** tab.
3. If required, change the user and group ID of the user that should run the command, the command timeout, and a command label of up to 10 characters.
4. In the **Script** text box, enter the command script that you want to run. The following example runs the `who` command:

```
#!/bin/sh
who
```

5. If required, change the schedule for the command.
6. Click **Schedule** to commit the command script to run according to the schedule that you specify.

If you have not edited the schedule and you have enabled the OSA daemon on the client, the OSA daemon usually runs the command immediately.

7. Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the details of the script and any output if it has already run on the client.

Using the `system_runscript` Command

To run a command on client systems, use the command syntax as follows. Ensure that you confirm the command as prompted.

```
system_runscript group:group-name -s start-time -t timeout -f path-to-script
```

The `-s` option specifies the date and time, `YYYYMMDD[hhmm]` format, when an event should start. If you do not specify a start time, Oracle Linux Manager assumes `0000` (midnight).

The `-t` option defines a timeout for a client to confirm that it has run a command. After this time has elapsed, Oracle Linux Manager assumes that the command has failed.

The following example illustrates use of the command and the output that is displayed:

```
spacecmd {SSM:0}> system_runscript group:group3 -s 20150617T0130 -t 60 -f /root/myscript

User:      root
Group:     root
Timeout:   60 seconds
Start Time: 20150617T01:30:00

Script Contents
-----
#!/bin/sh
yum update

Systems
-----
svr2.mydom.com
svr3.mydom.com

Is this ok [y/N]: y
INFO: Action ID: 343
INFO: Scheduled: 2 system(s)
```

Working With Scheduled Events

To display a list of completed, failed, and pending events, use this command:

```
schedule_list
```

For example, running the command might display the following information:

```
spacecmd {SSM:0}> schedule_list

ID      Date                C   F   P   Action
--      ----                -   -   -   -
...
343     20150617T01:30:00   0   0   2   Run an arbitrary script
...
```

The `C`, `F`, and `P` columns show the number of systems on which the event has completed, failed, or is pending. To display only completed, failed, or pending events, use the `schedule_listcompleted`, `schedule_listfailed`, or `schedule_listpending` commands.

The following commands apply actions to schedule related events:

- Display a pending event.

```
spacecmd {SSM:0}> schedule_details ID
```
- Cancel a pending event.

```
spacecmd {SSM:0}> schedule_cancel ID
```
- Reschedule a failed event.

```
spacecmd {SSM:0}> schedule_reschedule ID
```

Working With Configuration Channels

Configuration channels contain files for configuring client systems, for example, configuration information for services, applications, or users.

Creating Customer Information Keys

Custom information keys enable you to extract configuration information from clients. For example, you could define a custom information key named `uptrack-uname` to store the value of the Ksplice effective kernel version.

**Note:**

For information about activation keys, see [Working With Activation Keys in Oracle Linux Manager](#).

Using the Oracle Linux Manager Web Interface

To define a custom information key and assign it to a system, follow these steps:

1. Go to **Systems** and select **Custom System Info**.
2. Click **+ Create New Key**.
3. On the Create Custom Info Key page, enter a key label (for example, `asset_tag`) and description, and then click **Create Key**.
4. Go to **Systems** and click the name of the system for which you want to assign a value to the key.
5. Select the **Custom Info** tab.
6. On the Custom System Information page, click **+ Create New Value**.
7. On the Edit Custom Info Key page, select the key to which you want to assign a value. The page updates to display information about the key and a **Value** text box.
8. Enter the key value in the **Value** text box and click **Update Key**.

The Custom System Information page displays the key-value pairs that are associated with a system. You can modify a value by selecting the associated **Edit this value** link.

**Note:**

You can also define custom information keys for a system by using the **Custom Info** tab of a system's Kickstart profile.

You can define as many key-value pairs for a system as you require.

Using the `custominfo_createkey` Command

To create a custom information key, use the `custominfo_createkey` command as follows:

```
spacecmd {SSM:0}> custominfo_createkey admin_user "Email of admin contact"
```

To list the available custom information keys, use the `custominfo_listkeys` command:

```
spacecmd {SSM:0}> custominfo_listkeys
```

To assign a custom information key to a system or system group, use the `system_addcustomvalue`:

```
spacecmd {SSM:0}> system_addcustomvalue asset_tag "fc01568a" svr3.mydom.com
spacecmd {SSM:0}> system_addcustomvalue admin_user "an.admin@mydom.com" group:group3
```

To list the custom information keys for a system or system group, use the `system_listcustomvalues` command:

```
spacecmd {SSM:0}> system_listcustomvalues system-name
spacecmd {SSM:0}> system_listcustomvalues group:group
```

Applying Custom Information Keys

To apply a custom information key, follow these steps:

1. Install the `rhn-custom-info` package on each client with which you want to use the key.
2. Use the `rhn-custom-info` command to make the value of the key available in Oracle Linux Manager:

```
sudo rhn-custom-info uptrack-uname `uptrack-uname -r`
```

This command makes the value returned by `uptrack-uname -r` available as the value of the `uptrack-uname` key. You can then use the macro `rhn.system.custom_info(uptrack-uname)` to extract the value of `uptrack-name` within a configuration file.

Setting Up Configuration Channels

Use either the web interface or the `spacecmd` command to work with configuration channels.

Using the Oracle Linux Manager Web Interface

Figure 10-2 New Config Channel Page

To create a configuration channel:

1. Go to **Configuration** and select **Configuration Channels**
2. On the Centrally Managed Configuration Channels page, click **+ Create New Config channel**.
3. Enter a name, label, and description for the channel. The label should be a short representation of the target operating system, architecture (if appropriate), and the purpose of the channel, for example, `ol6_generic_configuration`.
4. Click **Create Config Channel**.
5. To add files to the configuration channel, on the New Channel page, select the **Add Files** tab.
 - To create a text file, directory, or symbolic link:
 - a. Select the **Create File** tab.
 - b. On the Create New Configuration File page, create a text file, directory, or symbolic link.

For example, to set up a message-of-the-day file that contains configuration information about the client, you might enter the following details:

File Type

Select **Text file**.

Filename/Path

Enter `/etc/motd`.

Ownership

Enter `root` for both the user name and group. (These are the default entries.)

File Permissions Mode

Enter `644`. (This is the default mode.)

File Contents

Select the file type as **Shell** from the drop-down list, and enter the file contents in the text field.

For example, the following file uses macros that Oracle Linux Manager replaces with the appropriate values for the system on which the file is deployed:

```

System Information
=====
Client system: {|rhn.system.hostname|}
Oracle Linux
  Manager SID: {|rhn.system.sid|}
Asset tag:     {|rhn.system.custom_info(asset_tag) = 'Asset tag missing'|}
Profile:      {|rhn.system.profile_name|}
Description:  {|rhn.system.description|}
IP address:   {|rhn.system.ip_address(eth0)|}
MAC address:  {|rhn.system.net_interface.hardware_address(eth0)|}

```

The custom macro `rhn.system.custom_info` substitutes the value of the custom system information key named `asset_tag` as assigned on the Custom Info tab for the system. Otherwise, it inserts the value `Asset tag missing`.

See [Using the Oracle Linux Manager Web Interface](#).

See [Configuration File Macros](#).

- c. After entering the details of the file, click **Create Configuration File**.
- To import files:
 - a. Select the **Import Files** tab.
On the Import Configuration File(s) from Another Channel page, you can import configuration files from other configuration channels.
 - b. Select the check boxes of the configuration files that you want to import.
 - c. Click **Import Configuration File(s)**.
- To upload files:
 - a. Select the **Upload File** tab.
 - b. On the Upload New Configuration File page, click **Browse...** and select the path of the file to upload.
 - c. Select the file type: **Text file** or **Binary file**.
 - d. Enter other details for the file, such as ownership and permissions, as required.
 - e. Click **Upload Configuration File**.

Using the `configchannel_create` Command

To create a configuration channel, use the `configchannel_create` command as shown in the following example. You would need to provide specific information, which is shown in bold.

```

spacecmd {SSM:0}> configchannel_create

Name: Oracle Linux 6 Server Configuration
Label: ol6-server-config
Description: Generic configuration channel for Oracle Linux 6 servers

```

To add a configuration file to a channel, use the `configchannel_addfile config0file` command. The command requires user input during the process, as shown in bold in the following example:

```
spacecmd {SSM:0}> configchannel_addfile ol6-server-config

Path: /etc/motd
Symlink [y/N]: N
Directory [y/N]: N
Owner [root]: [Enter]
Group [root]: [Enter]
Mode [0644]: [Enter]
SELinux Context [None]: [Enter]
Revision [next]: [Enter]
Read an existing file [y/N]: y
File: /var/config_file_templates/ol6-server/etc/motd
Path:          /etc/motd
Directory:     False
Owner:         root
Group:         root
Mode:          0644
SELinux Context:

Contents
-----
System Information
=====
Client system: {|rhn.system.hostname|}
Oracle Linux
  Manager SID: {|rhn.system.sid|}
Asset tag:     {|rhn.system.custom_info(asset_tag) = 'Asset tag missing'|}
Profile:       {|rhn.system.profile_name|}
Description:   {|rhn.system.description|}
IP address:    {|rhn.system.ip_address(eth0)|}
MAC address:   {|rhn.system.net_interface.hardware_address(eth0)|}

Is this ok [y/N]: y
```

The custom macro `rhn.system.custom_info` substitutes the value of the custom system information key named `asset_tag` as assigned on the Custom Info tab for the system. Otherwise, it inserts the value `Asset tag missing`.

See [Using the custominfo_createkey Command](#).

See [Configuration File Macros](#).

To display the details of a configuration channel, use the `configchannel_details` command:

```
spacecmd {SSM:0}> configchannel_details ol6-server-config
```

Subscribing Client Systems to Configuration Channels

Use either the web interface or the `spacecmd` command to subscribe client systems to configuration channels.

Using the Oracle Linux Manager Web Interface

To subscribe a client system to a configuration channel:

1. Go to **Systems** and click the system name.

2. Select the **Configuration** tab, then the **Manage Configuration Channels** tab, and finally the **Subscribe to Channels** tab.
3. For Step 1: Select Channels for Subscription page, select the check boxes for the channels to which you want to subscribe the system and click **Continue**.
4. For Step 2: Rank Channels for Subscription page, you can optionally change the order of the configuration channels according to priority. Higher-rank entries override lower-rank entries if several entries can modify the same files or directories.
5. To save your changes, click **Update Channel Rankings**.

See [Using the Oracle Linux Manager Web Interface](#).

Using the `system_addconfigchannels` Command

In a `spacecmd` session, list the available configuration channels as follows:

```
spacecmd {SSM:0}> configchannel_list
```

To subscribe a system or system group to a configuration channel, use the `system_addconfigchannels` command syntax:

```
spacecmd {SSM:0}> system_addconfigchannels svr3.mydom.com ol6-nfs-server-config -t
spacecmd {SSM:0}> system_addconfigchannels group:group3 ol6-server-config -b
```

To list the configuration channels to which a system or system group is subscribed, use the `system_listconfigchannels` command:

```
spacecmd {SSM:0}> system_listconfigchannels group:group3
```

To edit the configuration channels for a system, you can use the `system_setconfigchannelorder` command. The command lists the available channels for the system, and then enables you to select a channel to which you can specify a new rank, as shown in bold in the following example:

```
spacecmd {SSM:0}> system_setconfigchannelorder svr1.mydom.com
```

```
Current Selections
-----
1. ol6-server-config

a[dd], r[emove], c[lear], d[one]: a

Available Configuration Channels
-----
ol6-dns-server-config
ol6-http-server-config
ol6-nfs-server-config
ol6-server-config

Channel: ol6-dns-server-config
New Rank: 1

Current Selections
-----
1. ol6-dns-server-config
2. ol6-server-config

a[dd], r[emove], c[lear], d[one]: d
```

To unsubscribe a system from a configuration channel, use the `system_removeconfigchannels channel-name` command.

Deploying Configuration Files on Client Systems

Use either the web interface or the `spacecmd` command to deploy configuration files.

Using the Oracle Linux Manager Web Interface

Note:

You must have previously subscribed the system to the appropriate configuration channel for the configuration files that you want to deploy. See [Using the Oracle Linux Manager Web Interface](#).

The first deployment to a client system can fail if the `/var/log/rhncfg-actions` log file does not exist on the client. If the deployment fails, the log file is created automatically, and you can reschedule the deployment event. Alternatively, run the following command on the client system before deploying any files to it:

```
sudo touch /var/log/rhncfg-actions
```

Oracle recommends using configuration files to deploy SSL and GPG certificates to Oracle Linux Manager clients that were not previously provisioned by Oracle Linux Manager server.

Figure 10-3 Deploy Files Page

The screenshot shows the Oracle Linux Manager web interface. The top navigation bar includes tabs for Overview, Systems, Errata, Channels, Audit, Configuration, Schedule, Users, Admin, and Help. The 'Systems' tab is active, and the system 'svr1.mydom.com' is selected. The 'Configuration' tab is active, and the 'Deploy Files' sub-tab is selected. The page displays a table of deployable files with the following content:

Filename	Deployable Revision	Provider
<input checked="" type="checkbox"/> /etc/mod	Revision 1	Oracle Linux 7 Server Configuration

Below the table, there is a 'Select All' button and a 'Deploy Files' button. The page also shows a filter by filename input field and a '1 - 1 of 1 (1 selected)' indicator.

To deploy a configuration file to a client system:

1. Go to **Systems** and click the system name.
2. Select the **Configuration** tab and then the **Deploy Files** tab.

3. On the **Deploy Files** page, select the check boxes for the files that you want to deploy, and click **Deploy Files**.
4. On the **Confirm Deploy Files** page, change the schedule if required, and click **Schedule Deploy**.
5. Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the details of the deployment event.

Using the `system_deployconfigfiles` Command

Note:

You must have previously subscribed the system to the appropriate configuration channel for the configuration files that you want to deploy. See [Using the `system_addconfigchannels` Command](#).

The first deployment to a client system can fail if the `/var/log/rhncfg-actions` log file does not exist on the client. If the deployment fails, the log file is created automatically, and you can reschedule the deployment event. Alternatively, run the following command on the client system before deploying any files to it:

```
sudo touch /var/log/rhncfg-actions
```

In a `spacecmd` session, to deploy all configuration files to a system or system group, use the `system_deployconfigfiles` command: Ensure that you confirm the command, as shown in bold in the following example:

```
spacecmd {SSM:0}> system_deployconfigfiles group:group3

Systems
-----
svr3.mydom.com
svr4.mydom.com

Deploy ALL configuration files to these systems [y/N]: y
INFO: Scheduled deployment for 2 system(s)
```

To display the details of a pending deployment event, use the `schedule_details` command. See [Working With Scheduled Events](#).

Using OpenSCAP to Audit of Client Systems

 **WARNING:**

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

This chapter describes how to audit systems for security compliance within an Oracle Linux Manager setup by using the SCAP Security Guide or any OpenSCAP compliant SCDDF or OVAL files. Oracle provides OVAL files at <https://linux.oracle.com/security>.

For more information about OpenSCAP and its implementation on Oracle Linux, refer to the security guides corresponding to the Oracle Linux release on your systems at [Oracle® Linux Documentation](#)

See also the `oscap(8)` manual page for more information about the `oscap` command that is typically used with Oracle Linux Manager for system audits.

Requirements for Using OpenSCAP in Oracle Linux Manager

To run audit operations using OpenSCAP in Oracle Linux Manager, complete the following requirements:

- Enable Oracle Linux Manager server to run remote commands on client systems. See [Configuring Client Systems for Remote Management](#).
- Install the `spacewalk-oscaps` package on all client systems.
- Download the appropriate OVAL files against which the systems will be audited. You can obtain these files from <https://linux.oracle.com/security/>.

To be able to run scan operations on a client, the client must have at least the following packages installed:

- `openscap-utils`
- `openscap-scanner`
- `spacewalk-oscaps`

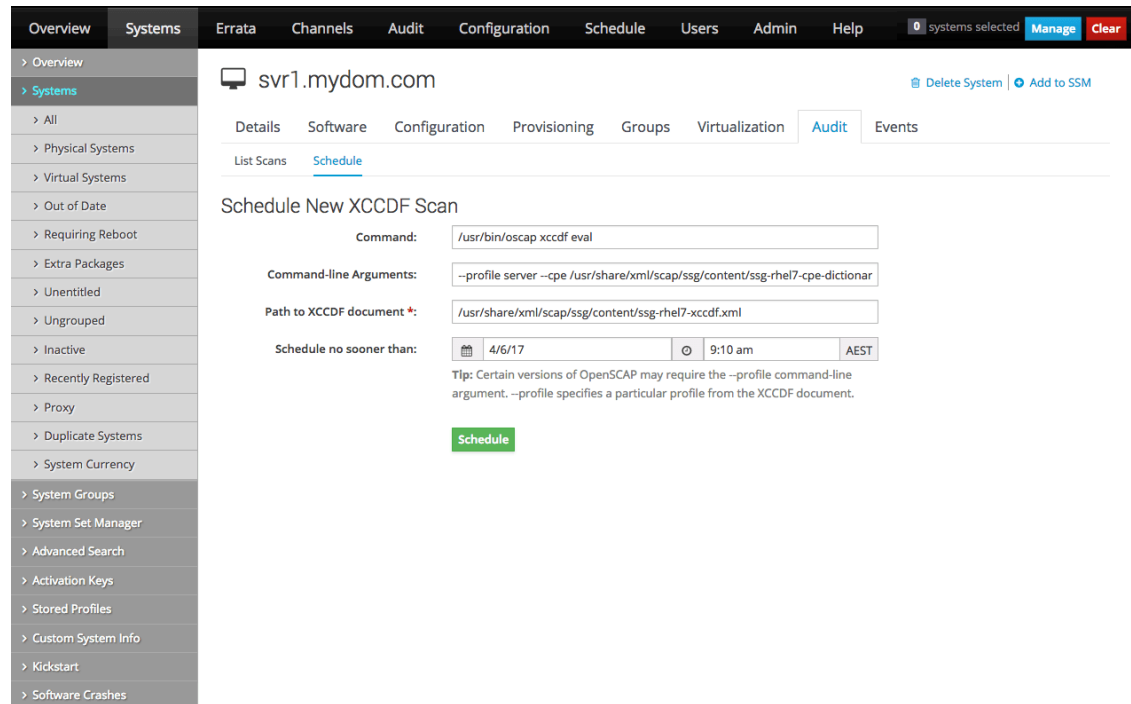
Optionally, you can also install `scap-security-guide` that provides all the content in `/usr/share/xml/scap/ssg/content/`.

Performing Audit Scans

This section describes how to schedule system scans on the web interface. You can schedule scans and audits for different systems or system groups.

Using the Oracle Linux Manager Web Interface

Figure 11-1 Schedule New XCCDF Scan Page



1. On the browser, log in to Oracle Linux Manager server (`https://server-fqdn`)
2. Go to **Systems**.
3. Select the target for scanning depending on the target.
 - To scan a system:
 - a. Click the system name.
 - b. Select the **Audit** tab.
 - c. Select the **Schedule** tab.
 - To scan a system group:
 - a. Select **System Groups**.
 - b. Click the system group name.
 - c. On the Details page, click **Work With Group**.
Oracle Linux Manager loads the group into the System Set Manager.
 - d. Select the **Audit** tab.
4. On the Schedule New XCCDF Scan page, enter the scan settings in the following fields:
 - **Command-line arguments**, for example, `--profile server`
 - **Path to XCCDF document**, for example, `/usr/share/xml/scap/ssg/content/ssg-ol7-xccdf.xml` or `com.oracle.elsa-2018.xml`.

 **Note:**

If necessary, download the XCCDF file from <https://linux.network.com/security>.

5. Change the schedule if required.
6. Click **Schedule**.

When the scan is complete, a summary of the results of the scan are displayed under the **List Scans** tab. Oracle recommends that you schedule regular scans to check for security regressions.

Using the `scap_scheduledxccdfscan` Command

The `spacecmd` command only supports XCCDF scans. For OVAL scans, use Oracle Linux Manager's remote command execution facility to run `oscap oval eval` on Oracle Linux Manager clients.

The following are examples of various `spacecmd` commands for auditing client systems.

- Scheduling an XCCDF scan:

```
spacecmd {SSM:0}> scap_scheduledxccdfscan '/usr/share/xml/scap/ssg/content/ssg-ol7-xccdf.xml' 'profile server' svr1.mydom.com
```

- Listing scheduled auditing scans:

```
spacecmd {SSM:0}> schedule_list
```

See [Working With Scheduled Events](#).

- Listing summary results of completed scans:

```
spacecmd {SSM:0}> scap_listxccdfscans svr1.mydom.com
```

- Listing details and results of an XCCDF scan:

```
spacecmd {SSM:0}> scap_getxccdfscandetails scan_ID  
spacecmd {SSM:0}> scap_getxccdfscanrulereults scan_ID
```

Configuring Ksplice Offline Clients

WARNING:

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

On average, the Linux kernel receives security updates and bug fixes about once per month. Traditionally, applying such updates would require you to obtain and install the updated kernel RPMs, to schedule downtime, and to reboot the server into the new kernel with the critical updates. As system setups become more complex with many interdependencies, and access to services and applications must remain as undisrupted as possible, scheduling such reboots becomes more difficult and costly.

Oracle Ksplice provides a way for you to keep your systems secure and highly available by enabling you to update them with the latest kernel, Xen hypervisor updates, and key user-space security and bug fix updates. Oracle Ksplice updates the running operating system and Xen hypervisor without requiring a reboot. Your systems remain up to date with OS vulnerability patches and downtime is minimized. A Ksplice update takes effect immediately when it is applied. The update is not the same as an on-disk change that only takes effect after a subsequent reboot.

Oracle creates each Ksplice update from a kernel update that originates either from Oracle or from the Linux kernel community.

The *Ksplice Offline client* removes the requirement for a server on your intranet to have a direct connection to the Oracle Uptrack server. All available Ksplice updates for each supported kernel version are bundled into an RPM that is specific to that version, and this package is updated every time that a new Ksplice patch becomes available for the kernel.

Note:

Ksplice Offline client is freely available for Oracle Linux customers that subscribe to Oracle Linux Premier Support. If you are an Oracle Linux Basic, Basic Limited, or Network Support subscriber, contact your sales representatives to discuss a potential upgrade of your subscription to a Premier Support plan.

You can configure an Oracle Linux Manager server as a mirror of the Ksplice for Oracle Linux channels on ULN. The Oracle Linux Manager server does not require access to the Oracle Uptrack server. Instead, you schedule Oracle Linux Manager to download the latest Ksplice update packages to a software channel. For older Ksplice updates, an archive channel is available. The `_archive` suffix is usually added to the channel for which it hosts archive packages. See [About Software Channel Configuration](#)

After installing Ksplice Offline client on your Oracle Linux Manager client systems, they can install the Ksplice update packages from the Oracle Linux Manager server. The clients also do not require access the Oracle Uptrack server.

 **Note:**

You cannot use the web interface or the Ksplice Uptrack API to monitor systems that are running Ksplice Offline client, as these systems are not registered with <https://status-ksplice.oracle.com>.

For more information about Ksplice see [Oracle Linux: Ksplice User's Guide](#).

Supported Kernels

You can use Ksplice Uptrack to keep Oracle Linux kernels up to date with the latest important security and bug fix patches. To confirm whether a particular kernel is supported, install the Uptrack client on a system that is running that kernel.

For up-to-date information about supported kernels, see [Oracle Linux: Ksplice User's Guide](#).

For additional questions, send an email to ksplice-support_ww@oracle.com.

Configuring an Oracle Linux Manager Server to Act as a Ksplice Mirror

 **Note:**

The following information applies to the configuration of an Oracle Linux Manager server *only*.

To configure an Oracle Linux Manager server to act as a Ksplice mirror, you configure repositories and associated software channels for the Oracle Linux releases and architectures of the clients on which you want to run Ksplice Offline client. Each Ksplice channel should be a child of the appropriate base software channel. See [Configuring Oracle Linux Manager Repositories](#) and [Configuring Software Channels](#).

 **Note:**

For a list of channels that are available for Ksplice on Oracle Linux, see Available Ksplice Channels in [Oracle Linux: Ksplice User's Guide](#).

For example, you would specify the URL of the Ksplice for Oracle Linux 7 (x86_64) channel on ULN as follows:

```
uln:///ol7_x86_64_ksplice
```

Provisioning Client Systems as Ksplice Offline Clients

To provision a client system as a Ksplice Offline client, configure its kickstart profile as follows:

- Under **Kickstart Details**, select the **Operating System** tab, ensure that the check box for the Ksplice child software channel is checked, and click **Update Kickstart**.
- Under **Software**, include `uptrack-offline` in the list of packages to install.
- Under **Scripts**, create a post-installation, `nochroot` shell script that installs the Ksplice update packages.

```
sudo yum install -y uptrack-updates-`uname -r`
```

Install new Ksplice updates as they become available. You can schedule Oracle Linux Manager to update the client system or you can set up an `anacron` script on the client itself. For example, you could use the following script with an Oracle Linux 6 or Oracle Linux 7 client:

```
#!/bin/sh
yum install -y uptrack-updates-`uname -r`
```

The script must be executable and also must be owned by `root`. If you place the script in `/etc/cron.daily` on the client, it runs once every day.

Installing and Configuring Existing Client Systems as Ksplice Offline Clients

After you have set up Oracle Linux Manager to act as a Ksplice mirror, you can configure your other systems to receive `yum` and Ksplice updates.

To configure a system as a Ksplice Offline client, follow these steps:

1. Subscribe the client system to the Ksplice software channel that corresponds to the Oracle Linux release and architecture.
2. Install the offline version of the enhanced Ksplice client package:

```
sudo yum install -y ksplice-offline
```

3. Insert a configuration directive into `/etc/uptrack/uptrack.conf` to provide the enhanced client with the label of the local user-space channel in your local Yum repository configuration. You do not need to do this if you did not use the `local_` prefix for the channel label and this label matches the label used on ULN exactly. If you used the `local_` prefix or labeled this channel differently, add the following lines and replace `local_ol6_x86_64_ksplice_userspace` with whatever you used to label the Ksplice user-space channel:

```
[User]
yum_userspace_ksplice_repo_name = local_ol6_x86_64_ksplice_userspace
```

4. To install offline update packages, you must install the relevant packages for your system. For example, you might install the following packages:

```
sudo yum install ksplice-updates-glibc ksplice-updates-openssl
```

When these packages have been installed, the offline version of the enhanced Ksplice client behaves exactly the same as the online version.

5. Update the system to install the Ksplice-aware versions of the user-space libraries:

```
sudo yum update -y
```

To install only the libraries and not update any other packages, limit the update to the `ol_arch_userspace_ksplice` channel as appropriate, for example:

```
sudo yum --disablerepo=* --enablerepo=ol7_x86_64_userspace_ksplice update
```

Alternatively, use the following command:

```
sudo yum update -y *glibc *openssl*
```

You may also use this client to perform kernel updates, in the same way that you are able to use the standard uptrack client:

```
sudo yum install -y uptrack-updates-`uname -r`
```

6. To enable the automatic installation of updates, ensure that `/etc/uptrack/uptrack.conf` has the following entry for `autoinstall`:

```
autoinstall = yes
```

7. Reboot the system so that the system uses the new libraries.

On Oracle Linux 6:

```
sudo reboot
```

On Oracle Linux 7 or Oracle Linux 8:

```
sudo systemctl reboot
```

Implementing Ksplice Synchronization Filters on Oracle Linux Manager Server

The Ksplice Offline repositories are massive in size and synchronizing Oracle Linux Manager server with the entire repositories would be inefficient. To reduce the overall space consumed by Ksplice Offline packages, Oracle strongly recommends using repository filters to limit downloaded packages to only those that apply to your client systems. You can apply a filter either on the Repository configuration in the Manage Repositories page in the web UI or by using the `spacewalk-repo-sync -i` command syntax.

The following command line example shows how to limit package synchronization only to Ksplice patches that are applicable to Unbreakable Enterprise Kernel Release 6:

```
sudo spacewalk-repo-sync -c channel_label -n -i ksplice-*,python-ksplice-*,uptrack-offline-*,uptrack-updates-version*
```

In the example, the `ksplice-*`, `python-ksplice-*`, and `uptrack-offline-*` packages must be permanently included in any Ksplice synchronization because these packages contain Ksplice tools. Then for the `uptrack-updates-version*` packages, you specify the versions that are required by the client systems, such as `uptrack-updates-5.4.17*`. Being very specific in the RPM versions you add to the filter results in faster operations as well as larger conserved server disk space.

An optimum configuration for obtaining high performance when synchronizing would be to have all Oracle Linux Manager servers to run an identical base kernel version at each upgrade point on the disks.

13

Kickstart Options

WARNING:

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

Using the Oracle Linux Manager web interface, you can specify the following options for a kickstart profile:

auth

(Mandatory) Specifies whether the shadow password file is used and the password algorithm. The default setting is `--enablesshadow --passalgo=sha256`, which enables the shadow password file and specifies SHA256 as the password algorithm. If you change the password algorithm, the password hash specified for `rootpwd` must have been generated by using the same algorithm or you will not be able to log in to the installed system.

See the `authconfig(8)` manual page for a list of the options that you can specify.

autopart

Specifies whether the installation should perform automatic partitioning. If you specify this option, you should also specify `clearpart` and `zerombr`. Use `ignoredisk` to specify the disks that the Installer should or should not use.

autostep

```
autostep [ --autoscreenshot ]
```

Specifies that the Installer should step through every screen.

bootloader

```
bootloader --location={mbr|none|partition}
           [ --append="boot-loader-kernel-parameters" ]
```

(Mandatory) Specifies whether the boot loader is installed in the MBR or in a disk partition. The default setting is `--location mbr`.

cdrom

Specifies that the installation is from the first CD-ROM drive on the system.

clearpart

```
clearpart [ --all [ --initlabel ] |
           --linux |
           --list=part1,... |
           --none ]
```

```
[ --drives=drive1,... ]
[ --initlabel ]
```

Specifies whether to clear any existing partitions. For example, `--drives=sda --all --initlabel` would clear all partitions on the disk device `sda` and reinitialize the disk label.

Caution:

The default setting of `--all` clears all partitions on all attached disks.

cmdline

Specifies that the installation should be performed in non-interactive, command-line mode.

device

```
device {eth|scsi}module_name --opts="module options"
```

Specifies the module name and options for a system device.

deviceprobe

Specifies how to probe for devices.

driverdisk

```
driverdisk partition [ --type=fstype ]
driverdisk --source=ftp://image_path
driverdisk --source=http://server/image_path
driverdisk --source=nfs:server:image_path
```

Specifies a driver disk.

firewall

```
firewall { --disabled | --enabled } [ --ftp ] [ --http ]
[ --port=inbound_port1:{tcp|udp},... ]
[ --smtp ] [ --ssh ]
[ --trust=network_interface ]...
```

Specifies the configuration of the system firewall. The default value is `--disabled`.

firstboot

```
firstboot { --enable | --disable | --reconfig }
```

Specifies how the setup agent starts when the system is first booted. If enabled, the `initial-setup` package must be installed.

graphical

Perform a graphical installation. It is not usual to select this option for non-interactive kickstart installations.

halt

Specifies that the Installer should halt the system after installation is complete and wait for a key to be pressed on the console before rebooting.

harddrive

```
harddrive [ --biospart=BIOS_partition | --partition=partition ]
          [ --dir=install_directory ]
```

Specifies an installation directory on a local hard drive.

ignoredisk

```
ignoredisk { --drives=[disk1,...] | --only-use=[disk1,...]
```

Specifies disks that the Installer should or should not use during installation.

install

Specifies that the Installer should perform a new installation. This option is specified by default.

interactive

Specifies that the installation should be interactive.

iscsi

```
iscsi --ipaddr=target_addr --target=target_IQN
      [ --iface=network_interface ] [ --port=target_port ]
      [ --user=target_username --password=target_password ]
      [ --reverse-username=initiator_username
        --reverse-password=initiator_password ]
```

Specifies iSCSI storage to be used during installation.

iscsiname

Specifies the iSCSI initiator name for the system.

key

Specifies an installation key for package selection and system identification.

keyboard

(Mandatory) Specifies the keyboard layout. The default setting is `us`.

lang

(Mandatory) Specifies the language to be used for installation and the default locale on the installed system. The default setting is `en_US`.

logging

```
logging [ --host=remote_host ]
        [ --level={critical|debug|error|info|warning} ]
        [ --port=remote_port ]
```

Configures installation error logging.

monitor

```
monitor [ --hsync=Hfreq ] [ --vsync=Vfreq ]
```

Specifies the monitor's horizontal and vertical synchronization frequency settings.

mouse

Deprecated. Do not use.

multipath

```
multipath --name=pathname --device=device --rule=rule
```

Specifies a multipath device.

network

```
network --bootproto=dhcp [ --device=interface ]
    [ --onboot={no|yes} ]

network --bootproto=static [ --device=network_interface ]
    [ --onboot={no|yes} ]
    [ --noipv4 | --ip=IP_addr --netmask=netmask ]
    [ --noipv6 | --ipv6={auto|dhcp|IPv6_addr/prefix} ]
    [ --gateway=gateway_addr ]
    [ --nameserver=namesvr_addr ]
```

Specifies the configuration of the network interfaces. The default setting is `--bootproto dhcp`, which configures the network interface used for installation to use DHCP to obtain its network settings.

nfs

```
nfs --server=NFSserver --dir=install_directory
    [ --opts=mount_options ]
```

Specifies an NFS server and directory path to use for installation.

poweroff

Specifies that the Installer should power down the system after installation is complete.

reboot

Specifies that the Installer should reboot the system after installation is complete. This option is specified by default. For unattended installations, the `poweroff` option might be preferable.

rootpw

```
rootpw { --iscrypted | --plaintext } password
```

(Mandatory) Specifies the `root` password as a hash value or in plain text. This option is specified by default. If you specify a plain text password, select the **Encrypt** check box.

selinux

```
selinux { --disabled | --enforcing | --permissive }
```

Specifies the SELinux mode as disabled, enforcing, or permissive. The default setting is `--permissive`.

services

```
services [ --disabled=service1,... ]
    [ --enabled=serviceA,... ]
```

Specifies which services to disable or enable at the default run level.

shutdown

Specifies that the Installer should shut down the system after installation is complete but not power it down.

skipx

Do not install X on the system. This option is specified by default.

text

Perform a text-only installation. This option is specified by default as kickstart installations are usually non-interactive.

timezone

```
timezone [ --utc ] timezone
```

(Mandatory) Specifies the time zone and whether the hardware clock uses UTC (`--utc`). The default setting is `America/New_York`.

upgrade

Specifies that the Installer should perform an upgrade installation.

url

```
url --url={file_path |
           ftp://username:password@server/path |
           http://server/path}
```

Specifies the URL of the kickstart file. By default, the kickstart URL is specified as a file path and Oracle Linux Manager writes the correct, full URL to the kickstart file, depending on whether the system being installed connects directly to an Oracle Linux Manager server or via a Oracle Linux Manager proxy. If you enter a full URL instead of a file path, Oracle Linux Manager does not modify the URL.

user

```
user --name=username [ --groups=group1,... ]
    [ --homedir=directory ]
    [ --password=password ] [ --iscrypted ]
    [ --shell=shell_path ] [ --uid=UID ]
```

Specifies a user to be created on the system.

vnc

```
vnc [ --host=hostname ] [ --port=port ]
    [ --password=password ]
```

Specifies parameters for running a VNC server on the system being installed.

xconfig

```
xconfig [ --defaultdesktop={GNOME|KDE} ]
        [ --depth={8|16|24|32} ]
        [ --resolution=XxY ]
        [ --startxonboot ]
```

Specifies X Window System parameters.

zerombr

Specifies whether to clear the existing disk partitions. This option is specified by default.

zfc

```
zfc [ --devnum=num ] [ --fcplun=lun ]  
    [ --scsiid=id ] [ --scsilun=lun ]  
    [ --wwpn=name ]
```

Specifies zFCP parameters for Fibre Channel-attached SCSI devices.

14

Sample Minimum Package Lists

 **WARNING:**

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

The following packages provide a suitable minimum (*Just Enough OS*) installation to support all Oracle Linux Manager management, monitoring configuration, and auditing functions. You can use the `yum` command to install any other packages that are required to configure an Oracle Linux server.

```
@ Base
osad
rhncfg
rhncfg-actions
rhncfg-client
rhncfg-management
```

15

Configuration File Macros

WARNING:

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

You can use the following standard macros with configuration files:

`rhn.system.custom_info(key_name)`

Value of the key `key_name` associated with the system.

`rhn.system.description`

System description.

`rhn.system.hostname`

System host name.

`rhn.system.ip_address`

Default system IP address.

`rhn.system.net_interface.broadcast(ethN)`

Broadcast address associated with `ethN` .

`rhn.system.net_interface.driver_module(ethN)`

Network interface driver module associated with `ethN` .

`rhn.system.net_interface.hardware_address(ethN)`

MAC address associated with `ethN` .

`rhn.system.net_interface.ip_address(ethN)`

IP address associated with `ethN` .

`rhn.system.net_interface.netmask(ethN)`

Network mask associated with `ethN` .

`rhn.system.profile_name`

Kickstart profile associated with a system.

`rhn.system.sid`

Oracle Linux Manager system ID.

Oracle Linux Manager XML/RPC API

⚠ WARNING:

The software described in this documentation is supported for a limited period under Extended Support. Oracle Linux 7 is now in Extended Support. See [Oracle Linux Extended Support](#) and [Oracle Open Source Support Policies](#) for more information.

Consider using OS Management Hub to manage operating system infrastructure. See [OS Management Hub](#) for more information.

If you are an advanced user, you can use Oracle Linux Manager XML/RPC API to create web interfaces and scripts to perform or automate tasks. More information about the API is available at https://olmsvr_FQDN/rpc/api on an Oracle Linux Manager server.

For example, the following `get-channel-summaries` Python script uses the API to obtain a list of channels, the numbers of packages in each channel, and the number of systems that are subscribed to each channel:

```
#!/usr/bin/python
#
# get-channel-summaries [--server URL <url>] [--username <user>] [--password <passwd>]

import getopt, struct, sys, xmlrpclib
from array import *

# Insert default values for Oracle Linux Manager server API URL,
# Oracle Linux Manager admin user name, and Oracle Linux Manager admin password
url = "https://olmsvr.mydom.com/rpc/api"
username = "swadmin"
password = "swadmin"

usage1 = "Usage: get-channel-summaries [--serverUrl <url>] \\n"
usage2 = "          [--username <user>] [--password <passwd>]"

try:
    opts,args = getopt.getopt(sys.argv[1:], "s:u:p:",
["serverUrl=", "username=", "password="])
except getopt.GetoptError as err:
    print(usage1+usage2)
    sys.exit(1)

for o,a in opts:
    if o in ("-s", "--serverUrl"):
        url = a
    elif o in ("-u", "--username"):
        username = a
    elif o in ("-p", "--password"):
        password = a
    else:
        assert False, "Unknown option"
```

```

# Connect to Oracle Linux Manager
client = xmlrpclib.Server(url,verbose=0)
session = client.auth.login(username,password)

# Get channel list
channels = client.channel.listAllChannels(session)

# Build channel arrays indexed by channel ID
channel_label = {}
channel_packages = {}
channel_systems = {}

for channel in channels:
    channel_label[channel['id']] = channel['label']
    channel_packages[channel['id']] = channel['packages']
    channel_systems[channel['id']] = channel['systems']

# Print output header
fmt1 = '{0:<40s}{1:<10s}{2:<10s}'
print fmt1.format('Channel label','Packages','Systems')
print fmt1.format('-----','-----','-----')

# Print channel label, package count, and system count -- sorted by label
fmt2 = '{0:<40s}{1:<10d}{2:<10d}'
for key,value in sorted(channel_label.iteritems(),key=lambda(k,v): (v,k)):
    id = int(key)
    print fmt2.format(value,channel_packages[id],channel_systems[id])

# Disconnect from Oracle Linux Manager
client.auth.logout(session)

```

The following is sample output from running this script:

Channel label	Packages	Systems
-----	-----	-----
base-channel	0	0
epel6-channel-label	68	0
epel6-channell1-label	3	0
epel6-channel2-label	1	0
ol5-i386-channel-label	44	0
ol5-i386-test-label	34	0
ol5-x64-channel-label	44	0
ol6-i386-channel-label	45	0
ol6-x64-channel-label	45	1
ol7-channel-label	45	1
oraclelinux6-x86_64	0	0
oraclelinux6-x86_64-addons	0	0
oraclelinux6-x86_64-spacewalk22-client	0	0
oraclelinux6-x86_64-spacewalk22-server	0	0
oraclelinux6-x86_64-spacewalk24-client	0	0
oraclelinux6-x86_64-spacewalk24-server	0	0
oraclelinux6-x86_64-spacewalk26-client	0	0
oraclelinux6-x86_64-spacewalk26-server	0	0
oraclelinux6-x86_64-spacewalk27-server	0	0
uln-channel-label	173	0

The following example shows the `get-reposync-list` script, which displays the schedules for synchronizing repositories.

```

#!/usr/bin/python
#
# get-reposync-list [--serverUrl <url>] [--username <user>] [--password <passwd>]

```

```

import getopt, struct, sys, xmlrpclib
from array import *

# Insert default values for Oracle Linux Manager server API URL,
# Oracle Linux Manager admin user name, and Oracle Linux Manager admin password
url = "https://olmsvr.mydom.com/rpc/api"
username = "swadmin"
password = "swadmin"

usage1 = "Usage: get-reposync-list [--serverUrl <url>] \\n\\n"
usage2 = "          [--username <user>] [--password <passwd>]"

try:
    opts,args = getopt.getopt(sys.argv[1:], "s:u:p:",
["serverUrl=", "username=", "password="])
except getopt.GetoptError as err:
    print(usage1+usage2)
    sys.exit(1)

for o,a in opts:
    if o in ("-s", "--serverUrl"):
        url = a
    elif o in ("-u", "--username"):
        username = a
    elif o in ("-p", "--password"):
        password = a
    else:
        assert False, "Unknown option"

# Connect to Oracle Linux Manager
client = xmlrpclib.Server(url, verbose=0)
session = client.auth.login(username,password)

# Get channel list
channels = client.channel.listAllChannels(session)

# Build channel name array indexed by channel ID
channel_label = {}
channel_schedule = {}

for channel in channels:
    id = int(channel['id'])
    channel_label[id] = channel['label']
    channel_schedule[id] = ''

# Get repository synchronization list
schedules = client.taskomatic.org.listActiveSchedulesByBunch(session, 'repo-sync-bunch')

# Construct schedule array indexed by channel ID
for schedule in schedules:
    channel_schedule[int(schedule['data_map']['channel_id'])] = schedule['cron_expr']

# Print output header
fmt = '{0:<40s}{1:<40s}'
print fmt.format('Channel label', 'Schedule')
print fmt.format('-----', '-----')

# Print channel labels and repository synchronization schedule (if defined)
for key,value in sorted(channel_label.iteritems(), key=lambda(k,v):(v,k)):
    id = int(key)
    sched = channel_schedule[id]

```

```
if (len(sched) > 0):
    print fmt.format(value,sched)
else:
    print fmt.format(value,"Sync not scheduled")

# Disconnect from Oracle Linux Manager
client.auth.logout(session)
```

The following example shows what the output of this command might look like:

Channel label	Schedule
-----	-----
oraclelinux6-x86_64	0 30 0 ? * *
oraclelinux6-x86_64-addons	0 30 2 ? * *
oraclelinux6-x86_64-mysql	0 30 4 ? * *
oraclelinux6-x86_64-playground	0 30 3 ? * *
oraclelinux6-x86_64-spacewalk24-client	0 0 5 ? * *
oraclelinux6-x86_64-spacewalk24-server	0 30 5 ? * *
oraclelinux6-x86_64-spacewalk26-client	0 0 2 ? * *
oraclelinux6-x86_64-spacewalk26-server	0 0 3 ? * *
oraclelinux6-x86_64-spacewalk27-server	0 0 4 ? * *
oraclelinux6-x86_64-uek	0 0 5 ? * *
oraclelinux6-x86_64-uek-r3	0 30 1 ? * *
...	