



Red Hat Update Infrastructure 2.1

Administration Guide

Configuration of Red Hat Update Infrastructure

Red Hat Update Infrastructure 2.1 Administration Guide

Configuration of Red Hat Update Infrastructure

Red Hat Update Infrastructure Documentation Team

Legal Notice

Copyright © 2014 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

The Red Hat Update Infrastructure Administration Guide provides requirements and instructions for the configuration of Red Hat Update Infrastructure for cloud providers.

Table of Contents

CHAPTER 1. USING THE RED HAT UPDATE INFRASTRUCTURE MANAGER	4
1.1. RED HAT UPDATE INFRASTRUCTURE MANAGER FIRST LAUNCH	4
1.2. USING THE RED HAT UPDATE INFRASTRUCTURE MANAGER	6
1.3. USING THE RED HAT UPDATE INFRASTRUCTURE MANAGER FROM THE SHELL PROMPT	9
CHAPTER 2. USING REPOSITORIES	13
2.1. USING CUSTOM REPOSITORIES	22
CHAPTER 3. USING CONTENT DELIVERY SERVER INSTANCES	27
CHAPTER 4. USING CLIENT ENTITLEMENTS	38
CHAPTER 5. MANAGING ENTITLEMENT CERTIFICATES	43
CHAPTER 6. USING IDENTITY CERTIFICATES	47
6.1. UPDATING THE ENTITLEMENT-SIGNING CA CERTIFICATE	48
CHAPTER 7. USING QPID SSL CERTIFICATES	49
CHAPTER 8. IMPLEMENTING A CERTIFICATION REVOCATION LIST	51
CHAPTER 9. STATUS AND SCHEDULING	54
CHAPTER 10. CONFIGURING USERS	61
CHAPTER 11. MONITORING	62
CHAPTER 12. UPDATING RED HAT UPDATE INFRASTRUCTURE	64
12.1. ADDITIONAL INSTRUCTIONS FOR UPDATING RED HAT UPDATE INFRASTRUCTURE 2.1.1 AND LOWER	66
CHAPTER 13. DELETING ORPHANED PACKAGES	69
CHAPTER 14. BACKUP AND RESTORE FOR RED HAT UPDATE INFRASTRUCTURE	72
14.1. BACKUP AND RESTORE FOR THE RED HAT UPDATE APPLIANCE	72
14.2. BACKUP AND RESTORE FOR A CONTENT DELIVERY SERVER	74
APPENDIX A. RED HAT UPDATE INFRASTRUCTURE 2.1 API REFERENCE	77
A.1. REPOSITORIES	77
A.2. LIST AVAILABLE REPOSITORIES	79
A.3. SYNCHRONIZATION	81
A.4. ADD FILTERS	81
A.5. ADD GROUP	82
A.6. CLONE	82
A.7. PUBLISH	83
A.8. REPOSITORY DISTRIBUTION	83
A.9. ADD NOTE	83
A.10. DELETE NOTE	84
A.11. UPDATE NOTE	84
A.12. SYNCHRONIZATION SCHEDULES	85
A.13. DELETE SCHEDULE	85
A.14. GET SCHEDULE	85
A.15. CREATE OR REPLACE SCHEDULE	86
A.16. GET DISTRIBUTIONS	86
A.17. LIST ALL	87
A.18. CONTENT DELIVERY SERVER (CDS)	88

A.19. FILTERS	95
A.20. TASKS	97
APPENDIX B. REVISION HISTORY	101
INDEX	104

CHAPTER 1. USING THE RED HAT UPDATE INFRASTRUCTURE MANAGER

Red Hat Update Infrastructure Manager is used to interact with Red Hat Update Infrastructure. It allows you to manage CDS instances, repositories, certificates, and users.

1.1. RED HAT UPDATE INFRASTRUCTURE MANAGER FIRST LAUNCH

To start Red Hat Update Infrastructure Manager run `rhui-manager` command from the command prompt. This operation must be performed as the root user in order to obtain the necessary permissions:

```
# rhui-manager
```

Starting the Red Hat Update Infrastructure Manager for the First Time

When Red Hat Update Infrastructure Manager is run for the first time, it will prompt for some extra information before it will start.

Entitlement-Signing CA Certificate and its Private Key

User will be prompted for the entitlement-signing CA certificate and its private key.

Password for CA Certificate (Optional)

Red Hat Update Infrastructure Manager will generate an identity certificate using the entitlement signing CA certificate. If the CA private key requires a password, then you will need to enter that information.

Default Name and Password

Red Hat Update Infrastructure Manager will ask you to log in using the default name *admin* and password *admin*:

```
--> rhui-manager
Previous authentication credentials could not be found. Logging into the
RHUI.
```

```
If this is the first time using the RHUI, it is recommended to change
the user's password in the User Management section of RHUI Tools.
```

```
RHUI Username: admin
RHUI Password: admin
```

The Home screen will be displayed after the password has been entered.

This login is used for the initial connection to the Red Hat Update Appliance. After the first login, a certificate is stored at `~/ .rhui` that identifies the user with the Red Hat Update Appliance for all future calls and prevents you from having to log in each time. Red Hat Update Infrastructure Manager will store authentication credentials for one week. Once authentication details have expired, all communications will fail until they have been refreshed. If you are using Red Hat Update Infrastructure when the credentials expire, you will be prompted to renew them the next time they are required for an operation. Otherwise, you will be asked to log in again next time you start Red Hat Update Infrastructure Manager.



IMPORTANT

Ensure you change the default password for the Red Hat Update Appliance before you begin using the system. For details on how to do this, see [Chapter 10, *Configuring Users*](#).

New Content Certificate

You will require a Red Hat content certificate and private key in order to access repositories with Red Hat Update Infrastructure Manager. Each product requires its own content certificate, which is provided by Red Hat. Once you have run Red Hat Update Infrastructure Manager for the first time, upload a new content certificate for each product you require by following the directions in [Procedure 5.3, “Upload Content Certificate”](#).

Content certificates are installed to `/etc/pki/rhui`. This means that you will require write permission for `/etc/pki/rhui` to be able to install a new content certificate.

Example 1.1. Running Red Hat Update Infrastructure Manager for the First Time

This example shows the information that Red Hat Update Infrastructure Manager requests when being run for the first time. This information must be provided before Red Hat Update Infrastructure Manager can operate.

```
# rhui-manager
An entitlement signing CA certificate is required to use RHUI Tools
but was not found.

Full path to the new signing CA certificate:
/root/rhui_certs/ca.crt

Full path to the new signing CA certificate private key:
/root/rhui_certs/ca.key

A RHUI identity certificate is required to use RHUI Tools but was not
found.
A new identity certificate will be generated now using the CA
certificate
found at /etc/pki/rhui/entitlement-ca.crt.

.....+++
.....+++
Previous authentication credentials could not be found. Logging into
the RHUI.

If this is the first time using the RHUI, it is recommended to change
the user's password in the User Management section of RHUI Tools.

RHUI Username: admin
RHUI Password:

-----
-----

                                -= Red Hat Update Infrastructure Management

Tool =-
```

```
-- Home --

r manage repositories
c manage content delivery servers (CDS)
s synchronization status and scheduling
e create entitlement certificates and client configuration RPMs
n manage Red Hat entitlement certificates
u manage users
i manage identity certificate

logout
  removes stored authentication credentials and exits
<
  move to the previous screen
^, home
  move to the home screen
/, clear
  clears the screen
?, help
  display help
q, quit, exit
  exit

          Connected: rhua.example.com
-----
-----
rhui (home) =>
```

[Report a bug](#)

1.2. USING THE RED HAT UPDATE INFRASTRUCTURE MANAGER

Red Hat Update Infrastructure Manager operates using an interactive shell. This section describes how to interact with Red Hat Update Infrastructure Manager, including common concepts and commands.

Procedure 1.1. Starting Red Hat Update Infrastructure Manager

1. Start Red Hat Update Infrastructure Manager by running the `rhui-manager` command from the command prompt:

```
# rhui-manager
```

2. If Red Hat Update Infrastructure Manager has not already stored a password for you, it will ask you to login. Once you have successfully authenticated, Red Hat Update Infrastructure Manager will display a prompt, ready to accept commands:

```
rhui (home) =>
```

3. When a menu is displayed, Red Hat Update Infrastructure Manager will display the name of the server it is connected to at the bottom, immediately above the prompt. At the prompt, the text in the parentheses () indicates which screen you are currently viewing. In this example, Red

Hat Update Infrastructure Manager is connected to `rhua.example.com`, and the current screen is the Home screen:

```
Connected: rhua.example.com
-----
rhui (home) =>
```

Procedure 1.2. Logging Out of the Red Hat Update Infrastructure Manager

1. The `logout` command is accessible at any time. Logging out will remove any stored authentication credentials that were saved when first logging in to Red Hat Update Infrastructure Manager. Next time Red Hat Update Infrastructure Manager is run, the user will be prompted to log in again.
2. To log out of Red Hat Update Infrastructure Manager, type `logout` at the prompt. You will be returned to the command line:

```
rhui (home) => logout
[user@example]$
```

Procedure 1.3. Exiting the Red Hat Update Infrastructure Manager

1. To exit Red Hat Update Infrastructure Manager, type `q`, `quit`, or `exit` at the prompt. You will be returned to the command line:

```
rhui (home) => quit
[user@example]$
```

2. Red Hat Update Infrastructure Manager can also be exited by pressing `Ctrl+C` from within the interface.



NOTE

`Quit` and `exit` perform the same operation. Credentials for a session are still valid after executing any of the two commands.

Procedure 1.4. Navigating around Red Hat Update Infrastructure Manager

There are several commands that can be used at any time to navigate around Red Hat Update Infrastructure Manager:

1. `<` to move to the previously viewed screen
2. `home` or `^` to return to the home screen
3. `clear` or `/` to clear the screen
4. `help` or `?` to display the help screen

The help screen is divided into screen and shell commands. Screen commands are used at the Red Hat Update Infrastructure Manager prompt, and shell commands are used at the shell prompt. For more information about using Red Hat Update Infrastructure Manager from the

shell prompt, see [Section 1.3, “Using the Red Hat Update Infrastructure Manager from the Shell Prompt”](#)

Red Hat Update Infrastructure Manager Screens

Red Hat Update Infrastructure Manager uses *screens* to provide access to functions. Each screen has its own menu of related commands

Home

The Home screen is the first screen displayed after logging in. It provides access to the other screens.

Repository Management

The Repository Management screen is used to list, add, and delete repositories. You can also create and upload packages to custom repositories.

CDS Management

The CDS Management screen is used to list, add, and delete content delivery server (CDS) instances.

From this screen, you can also access the CDS Repository Management screen, which allows you to view and change information for individual CDS instances. While in this screen, the prompt will indicate the hostname of the CDS being edited:

```
rhui (cds1.example.com) =>
```

Synchronization Status

The Synchronization Status and Scheduling screen is used to display and change the synchronization schedules for CDS instances and repositories, and view past synchronizations. You can also trigger individual synchronizations.

Client Entitlement Management

The Client Entitlement Management screen is used to create new entitlement certificates and client configuration RPMs.

Entitlements Manager

The Entitlements Manager screen is used to list entitled products in the current content certificate and upload new certificates.

User Manager

The User screen is used to change the user password.

Example 1.2. The Home Screen

This example shows the Home screen on a typical Red Hat Update Infrastructure Manager setup. Return to this screen at any time by typing `home` or `^` at the prompt:

```
-----
-----
      -= Red Hat Update Infrastructure Management Tool =-
```

```

-= Home =-

r  manage repositories
c  manage content delivery servers (CDS)
s  synchronization status and scheduling
e  create entitlement certificates and client configuration RPMs
n  manage Red Hat entitlement certificates
u  manage users
i  manage identity certificate

<  move to the previous screen
^, home
   move to the home screen
/, clear
   clears the screen
?, help
   display help
q, quit, exit
   exit

                                                    Connected:
rhua.example.com
-----
-----
rhui (home) =>

```

[Report a bug](#)

1.3. USING THE RED HAT UPDATE INFRASTRUCTURE MANAGER FROM THE SHELL PROMPT

Red Hat Update Infrastructure Manager is designed to be run using the interactive shell, however some functions can also be run from a standard shell prompt.

View all options and commands by using the `--help` command:

```

# rhui-manager --help
Usage: rhui-manager [options] [command]

OPTIONS
  -h/--help  show this help message and exit
  --debug    enables debug logging
  --config   absolute path to the configuration file; defaults to
/etc/rhui/rhui.conf
  --server   location of the RHUA server (overrides the config file)
  --username if specified, previously saved authentication credentials
are ignored and this username is used to login
  --password used in conjunction with --username

COMMANDS
  cert      : Red Hat content certificate management
  packages  : package manipulation on repositories

```

```
repo      : repository listing and manipulation
status    : RHUI status and health information
```

**NOTE**

Commands in this list that begin with a `--` can be used both in the interactive shell and at the shell prompt. All other commands can only be used at the shell prompt.

Flags**--debug**

Enables more verbose logging. The log file can be found at `~/ .rhui/rhui.log`

--config /path/examplefilename.conf

This command instructs Red Hat Update Infrastructure Manager (both the interactive shell and the shell prompt) to use the specified configuration file, instead of the default file located at `/etc/rhui/rhui-tools.conf`

--server rhua.example.com

This command instructs Red Hat Update Infrastructure Manager (both the interactive shell and the shell prompt) to connect to the specified server instead of the server specified in the configuration file. Authentication credentials are stored on a per server basis, so this flag can be used to allow a single Red Hat Update Infrastructure Manager installation to manage multiple Red Hat Update Infrastructure installations.

--username username --password password

This command instructs Red Hat Update Infrastructure Manager to use the specified credentials to log into the Red Hat Update Appliance. When using these commands, the authentication certificate will *not* be stored. This allows scripts to execute shell prompt commands on the Red Hat Update Infrastructure without needing to manage authentication certificates as well.

Content Certificate Management

The `cert` command is used to manipulate Red Hat content certificates.

rhui-manager cert info

Display information about the current content certificate.

rhui-manager cert upload

Use this command to upload a new content certificate. Specify the full path to the new certificate with the `--cert` flag, and the full path to the new key with the `--key` flag. Red Hat Update Infrastructure Manager and all repositories will be updated automatically when using this command.

```
# rhui-manager cert upload --cert /home/example/content-cert.crt --key
/home/example/content-key.pem
```

Repository Management

The `repo` command is used to manipulate repositories on the Red Hat Update Infrastructure.

`rhui-manager repo list`

Displays a list of repositories currently in the Red Hat Update Infrastructure:

```
# rhui-manager repo list
ID                               :: Repository Name

Red Hat Repositories
-----
rhui-srpms-1.2-5Server-x86_64    :: Red Hat Update Infrastructure 1.2
(SRPMS)
(5Server-x86_64)
rhui-srpms-1.2-5Server-i386     :: Red Hat Update Infrastructure 1.2
(SRPMS)
(5Server-i386)
rhui-1.2-5Server-x86_64        :: Red Hat Update Infrastructure 1.2
(RPMs)
(5Server-x86_64)
rhui-1.2-5Server-i386         :: Red Hat Update Infrastructure 1.2
(RPMs)
(5Server-i386)

Custom Repositories
-----
repo_1                          :: Custom Repository 1
```

`rhui-manager repo info`

Displays information about a specified repository. Specify the name of the repository with the `--repo_id` flag. The repository ID is shown in the output of the `rhui-manager repo list` command.

```
# rhui-manager repo info --repo_id rhui-1.2-5Server-x86_64
Name:                Red Hat Update Infrastructure 1.2 (RPMs) (5Server-
x86_64)
Type:                Red Hat
Relative Path:
content/dist/rhel/rhui/server/5Server/x86_64/rhui/1.2/os
Package Count:      13
Last Sync:          04-16-2011 21:58
Next Sync:          04-16-2011 23:44
```

Package Management

The `packages` command is used to manipulate packages on repositories.

`rhui-manager packages list`

Displays a list of all packages in a repository. Specify the name of the repository with the `--repo_id` flag. The repository ID is shown in the output of the `rhui-manager repo list` command:

```
# rhui-manager packages list --repo_id rhui-1.2-5Server-x86_64
```

```
PyYAML-3.08-4.el5.x86_64.rpm
grinder-0.0.57-1.el5.noarch.rpm
httpd-2.2.3-43.el5_5.3.x86_64.rpm
libyaml-0.1.2-3.el5.x86_64.rpm
m2crypto-0.16-6.1.el5_5.1.x86_64.rpm
mod_python-3.3.1-12.el5.x86_64.rpm
python-hashlib-20081119-5.el5.x86_64.rpm
python-pycurl-7.15.5.1-4.el5.x86_64.rpm
rh-cds-0.27-1.el5_5.noarch.rpm
rh-rhua-0.91-1.el5_5.noarch.rpm
rh-rhua-0.95-1.el5_5.noarch.rpm
rh-rhui-tools-0.76-1.el5_5.noarch.rpm
rpm-build-4.4.2.3-20.el5_5.1.x86_64.rpm
```

rhui-manager packages upload

Use this command to upload a new package to a custom repository. Specify the repository to upload the packages to with the `--repo_id` flag, and the full path to an `.rpm` file or directory containing `.rpm` files with the `--packages` flag. The repository ID is shown in the output of the `rhui-manager repo list` command.

```
# rhui-manager packages upload --repo_id rhui-1.2-5Server-x86_64 --
packages /home/example/filename.rpm
```

Status

The `status` command is used to check the status of the Red Hat Update Infrastructure.

rhui-manager status

Displays a short summary of whether or not CDS instances are running.

```
# rhui-manager status
US East Region CDS ..... [ UP ]
```

Use the `--code` flag to display a code that reflect the state of the Red Hat Update Infrastructure. An output of `0` means the Red Hat Update Infrastructure is running successfully. For any other outputs and more information about status codes, see [Procedure 11.1, “Monitoring the state of Red Hat Update Infrastructure”](#)

```
# rhui-manager status --code
0
```

[Report a bug](#)

CHAPTER 2. USING REPOSITORIES

The Repository Management screen is used to list, add, and delete repositories. You can also create and upload packages to custom repositories.

You will require a Red Hat content certificate and private key in order to access repositories with Red Hat Update Infrastructure Manager. Each product requires its own content certificate, which is provided by Red Hat. Once you have run Red Hat Update Infrastructure Manager for the first time, upload a new content certificate for each product you require by following the directions in [Procedure 5.3, “Upload Content Certificate”](#).

If you attempt to work with repositories without having a valid content certificate, you will see the following error message:

```
No Red Hat content certificates have been loaded. Upload
a content certificate to provide entitled products for import.
```

To access the Repository Management screen, go to the Home screen and type `r` at the prompt.

```
rhui (home) => r

-= Repository Management =-

l  list repositories currently managed by the RHUI
i  display detailed information on a repository
a  add a new Red Hat content repository
c  create a new custom repository
d  delete a repository from the RHUI
u  upload content to a custom repository
p  list packages in a repository

                               Connected: rhua.example.com
-----
rhui (repo) =>
```

Procedure 2.1. List Repositories

- From the Repository Management screen, type `l` at the prompt to list the repositories currently being managed by the Red Hat Update Infrastructure.

```
-----
-----
rhui (repo) => l

Custom Repositories
  Custom Repository 1

Red Hat Repositories
  Red Hat Update Infrastructure 1.2 (RPMs) (5Server-i386)
  Red Hat Update Infrastructure 1.2 (RPMs) (5Server-x86_64)
  Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-i386)
  Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-x86_64)
```

```
-----
-----
```

Procedure 2.2. Display Repository Information

1. From the Repository Management screen, type `i` at the prompt to display information about a particular repository.

```
-----
-----
rhui (repo) => i
```

2. A list of all repositories currently being managed by the Red Hat Update Infrastructure will be displayed. Select which repositories to view by typing the number of the repository at the prompt.

```
Select one or more repositories:
```

```
Custom Repositories
- 1 : Custom Repository 1

Red Hat Repositories
- 2 : Red Hat Update Infrastructure 1.2 (RPMs) (5Server-i386)
- 3 : Red Hat Update Infrastructure 1.2 (RPMs) (5Server-x86_64)
- 4 : Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-i386)
- 5 : Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-
x86_64)

Enter value (1-5) to toggle selection, 'c' to confirm selections, or
'?'
for more commands: 1
```

Typing the number of a repository will place a checkmark next to the name of that repository.

```
Select one or more repositories:
```

```
Custom Repositories
x 1 : Custom Repository 1

Red Hat Repositories
- 2 : Red Hat Update Infrastructure 1.2 (RPMs) (5Server-i386)
- 3 : Red Hat Update Infrastructure 1.2 (RPMs) (5Server-x86_64)
- 4 : Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-i386)
- 5 : Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-
x86_64)

Enter value (1-5) to toggle selection, 'c' to confirm selections, or
'?'
for more commands: 2
```

```
Select one or more repositories:
```

```
Custom Repositories
```

```

x 1 : Custom Repository 1

Red Hat Repositories
x 2 : Red Hat Update Infrastructure 1.2 (RPMs) (5Server-i386)
- 3 : Red Hat Update Infrastructure 1.2 (RPMs) (5Server-x86_64)
- 4 : Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-i386)
- 5 : Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-
x86_64)

Enter value (1-5) to toggle selection, 'c' to confirm selections, or
'?'
for more commands:

```

Continue until all repositories you wish to view have been checked, and then type **c** at the prompt to display the details.

```

Enter value (1-5) to toggle selection, 'c' to confirm selections, or
'?'
for more commands: c

Name:                Custom Repository 1
Type:                Custom
Relative Path:       custom/i386
Entitlement Path:    /custom/$basearch
Package Count:      0

Name:                Red Hat Update Infrastructure 1.2 (RPMs)
(5Server-i386)
Type:                Red Hat
Relative Path:       content/dist/rhel/rhui/server/5Server/i386/rhui/1.2/os
Package Count:      13
Last Sync:           05-11-2011 16:01
Next Sync:           05-13-2011 16:01

-----
-----
rhui (repo) =>

```

Procedure 2.3. Add a Red Hat Repository

1. Before adding a new Red Hat repository, you will need to have the specific repositories loaded for your entitled products. For example, if your content certificate grants access to the product **Red Hat Enterprise Linux Server (RPMs)**, you will need to have both 32-bit and 64-bit repositories loaded. This occurs automatically when Red Hat Update Infrastructure Manager contacts RHN.
2. From the Repository Management screen, type **a** at the prompt to add a new Red Hat repository.

```
rhui (repo) => a
```

3. There are three ways to add new Red Hat repositories:

- *All in Certificate*: Adds all available repositories for all entitled products in the certificate. You will be required to confirm the list of repositories before they are added.
- *By Product*: Adds all available repositories for chosen products. As each repository is added, it will be displayed.
- *By Repository*: This option shows a list of all available repositories for all entitled products. Select each individual repository to be added.

```
rhui (repo) => a
```

```
Loading latest entitled products from Red Hat...
... listings loaded
Determining undeployed products...
... product list calculated
```

```
Import Repositories:
 1 - All in Certificate
 2 - By Product
 3 - By Repository
Enter value (1-3) or 'b' to abort:
```

- **All in Certificate method**

From the Add a Red Hat Repository screen, select the **All in Certificate** method by typing **1** at the prompt.

```
Enter value (1-3) or 'b' to abort: 1
```

```
All repositories for all entitled products will be imported in
this process. This may result in multiple repositories being
imported,
such as if a product is supported on multiple architectures.
```

```
The following products will be deployed:
```

```
Red Hat Enterprise Linux Server 6 Releases (RPMs)
  Red Hat Enterprise Linux Server 6 Releases (RPMs) (6Server-
x86_64)
  Red Hat Enterprise Linux Server 6 Releases (RPMs) (6Server-
i386)
  Red Hat Enterprise Linux Server 6 Releases (SRPMS)
  Red Hat Enterprise Linux Server 6 Releases (SRPMS) (6Server-
x86_64)
  Red Hat Enterprise Linux Server 6 Releases (SRPMS) (6Server-
i386)
  Red Hat Enterprise Linux Server 6 Updates (RPMs)
  Red Hat Enterprise Linux Server 6 Updates (RPMs) (6Server-
x86_64)
  Red Hat Enterprise Linux Server 6 Updates (RPMs) (6Server-
i386)
  Red Hat Enterprise Linux Server 6 Updates (SRPMS)
  Red Hat Enterprise Linux Server 6 Updates (SRPMS) (6Server-
i386)
  Red Hat Enterprise Linux Server 6 Updates (SRPMS) (6Server-
x86_64)
Proceed? (y/n)
```

Confirm the list of repositories by typing **y** at the prompt.



NOTE

Content will not be downloaded to the newly imported repositories immediately. It will download when the next synchronization occurs.

o By Product

From the Add a Red Hat Repository screen, select the **By Product** method by typing **2** at the prompt.

```
Import Repositories:
 1 - All in Certificate
 2 - By Product
 3 - By Repository
Enter value from (1-3) or 'b' to abort: 2
```

A list of products with repositories available to be added will be displayed. Select the products to be deployed to the Red Hat Update Infrastructure by typing the number of each product at the prompt. Typing the number of a product will place a checkmark next to the name of that product. Continue until all products you want to deploy have been checked, and then type **c** at the prompt to confirm.

```
All repositories for selected products will be imported in this process.
```

```
Select the products to be deployed to the RHUI (only undeployed products are displayed):
```

```
- 1 : Red Hat Enterprise Linux Server (RPMs)
- 2 : Red Hat Enterprise Linux Server (SRPMS)
- 3 : Red Hat Enterprise Linux Server (STS)
- 4 : Red Hat Enterprise Linux Server 6 Optional Releases (RPMs)
- 5 : Red Hat Enterprise Linux Server 6 Optional Releases (SRPMS)
- 6 : Red Hat Enterprise Linux Server 6 Optional Updates (RPMs)
- 7 : Red Hat Enterprise Linux Server 6 Optional Updates (SRPMS)
- 8 : Red Hat Enterprise Linux Server 6 Releases (RPMs)
- 9 : Red Hat Enterprise Linux Server 6 Releases (SRPMS)
- 10: Red Hat Enterprise Linux Server 6 Updates (RPMs)
- 11: Red Hat Enterprise Linux Server 6 Updates (SRPMS)
- 12: Red Hat Update Infrastructure 1.2 (SRPMS)
```

```
Enter value from (1-12) to toggle selection, 'c' to confirm selections, or '?' for more commands:
```

**NOTE**

Content will not be downloaded to the newly imported repositories immediately. It will download when the next synchronization occurs.

- o **By Repository**

From the Add a Red Hat Repository screen, select the **By Repository** method by typing **3** at the prompt.

```
Enter value (1-3) or 'b' to abort: 3
```

```
Select the product repositories to be deployed to the RHUI (only
undeclared
products are displayed):
```

```
Red Hat Enterprise Linux Server 6 Optional Updates (SRPMS)
- 1 : Red Hat Enterprise Linux Server 6 Optional Updates
(SRPMS)
(6Server-i386)
- 2 : Red Hat Enterprise Linux Server 6 Optional Updates
(SRPMS)
(6Server-x86_64)
```

```
Red Hat Enterprise Linux Server 6 Updates (RPMs)
- 3 : Red Hat Enterprise Linux Server 6 Updates (RPMs)
(6Server-x86_64)
- 4 : Red Hat Enterprise Linux Server 6 Updates (RPMs)
(6Server-i386)
```

```
Enter value (1-4) to toggle selection, 'c' to confirm selections,
or '?' for
more commands: 1
```

```
... [output truncated] ...
```

```
Enter value (1-4) to toggle selection, 'c' to confirm selections,
or '?' for
more commands: c
```

```
The following product repositories will be deployed:
```

```
Red Hat Enterprise Linux Server 6 Optional Updates (SRPMS)
Red Hat Enterprise Linux Server 6 Optional Updates (SRPMS)
(6Server-i386)
Proceed? (y/n)
```

A list of the repositories to be deployed will be displayed. Select the required repositories, then type **c** to confirm the selection. Type **y** at the prompt to confirm the selection and deploy the products.

**NOTE**

Content will not be downloaded to the newly imported repositories immediately. It will download when the next synchronization occurs.

Procedure 2.4. Delete a Red Hat Repository

1. When a Red Hat repository is deleted using Red Hat Update Infrastructure Manager, it gets removed from both Red Hat Update Appliance and all CDS instances it is deployed on.

From the Repository Management screen, type `d` at the prompt to delete a Red Hat repository.

```
rhui (repo) => d
```

2. A list of all repositories currently being managed by the Red Hat Update Infrastructure will be displayed. Select which repositories to delete by typing the number of the repository at the prompt. Typing the number of a repository will place a checkmark next to the name of that repository. Continue until all repositories you wish to delete have been checked, and then type `c` at the prompt to confirm.

```
Select one or more repositories:
```

```
Custom Repositories
```

```
- 1 : Custom Repository 1
```

```
Red Hat Repositories
```

```
- 2 : Red Hat Enterprise Linux Server 6 Updates (RPMs)
(6Server-i386)
- 3 : Red Hat Enterprise Linux Server 6 Updates (RPMs)
(6Server-x86_64)
- 4 : Red Hat Enterprise Linux Server 6 Updates (SRPMS)
(6Server-i386)
- 5 : Red Hat Enterprise Linux Server 6 Updates (SRPMS)
(6Server-x86_64)
- 6 : Red Hat Update Infrastructure 1.2 (RPMs) (5Server-i386)
- 7 : Red Hat Update Infrastructure 1.2 (RPMs) (5Server-x86_64)
- 8 : Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-i386)
- 9 : Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-
x86_64)
```

```
Enter value (1-9) to toggle selection, 'c' to confirm selections, or
'?'
```

```
for more commands: 2-4
```

```
Select one or more repositories:
```

```
Custom Repositories
```

```
- 1 : Custom Repository 1
```

```
Red Hat Repositories
```

```
x 2 : Red Hat Enterprise Linux Server 6 Updates (RPMs)
(6Server-i386)
x 3 : Red Hat Enterprise Linux Server 6 Updates (RPMs)
(6Server-x86_64)
x 4 : Red Hat Enterprise Linux Server 6 Updates (SRPMS)
(6Server-i386)
- 5 : Red Hat Enterprise Linux Server 6 Updates (SRPMS)
(6Server-x86_64)
- 6 : Red Hat Update Infrastructure 1.2 (RPMs) (5Server-i386)
- 7 : Red Hat Update Infrastructure 1.2 (RPMs) (5Server-x86_64)
- 8 : Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-i386)
- 9 : Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-
```

```
x86_64)
```

```
Enter value (1-9) to toggle selection, 'c' to confirm selections, or
'?'
```

```
for more commands: c
```

3. A list of the repositories to be deleted will be displayed. Type **y** at the prompt to confirm the selection and delete the repositories.

```
The following repositories will be deleted:
```

```
Red Hat Repositories
```

```
Red Hat Enterprise Linux Server 6 Updates (RPMs) (6Server-i386)
```

```
Red Hat Enterprise Linux Server 6 Updates (RPMs) (6Server-
x86_64)
```

```
Red Hat Enterprise Linux Server 6 Updates (SRPMS) (6Server-i386)
```

```
Proceed? (y/n) y
```



NOTE

Repositories will be deleted only after the next CDS synchronization occurs (For synchronization, see [Section A.18.8, “Trigger a CDS Synchronization”](#)). Once the repositories have been deleted, the client configuration rpms that refer to the deleted repositories won't be able to be executed by yum.

Procedure 2.5. List Packages in a Repository



NOTE

When listing repositories within Red Hat Update Infrastructure Manager, only repositories that contain less than 100 packages will have their contents displayed. Results with more than 100 packages will display a package count only. To see a complete list, regardless of how many packages are contained within a repository, use the `rhui-manager packages list` command at the shell prompt instead. See [Section 1.3, “Using the Red Hat Update Infrastructure Manager from the Shell Prompt”](#) for more information about managing Red Hat Update Infrastructure from the shell prompt.

1. From the Repository Management screen, type **p** at the prompt to list the packages in a particular repository.

```
rhui (repo) => p
```

```
Choose a repository:
```

```
1 - Custom Repository 1
```

```
2 - Red Hat Enterprise Linux Server 6 Updates (RPMs) (6Server-
i386)
```

```
3 - Red Hat Enterprise Linux Server 6 Updates (RPMs) (6Server-
x86_64)
```

```
4 - Red Hat Enterprise Linux Server 6 Updates (SRPMS) (6Server-
i386)
```

```
5 - Red Hat Enterprise Linux Server 6 Updates (SRPMS) (6Server-
x86_64)
```



```

6 - Red Hat Update Infrastructure 1.2 (RPMs) (5Server-i386)
7 - Red Hat Update Infrastructure 1.2 (RPMs) (5Server-x86_64)
8 - Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-i386)
9 - Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-x86_64)

```

Enter value (1-9) or 'b' to abort:

2. Select the number of the repository you wish to view. Red Hat Update Infrastructure Manager will ask if you want to filter the results. Leave the line blank to see the results without a filter.

Enter value (1-9) or 'b' to abort: 8

Enter the first few characters (case insensitive) of an RPM to filter the results (blank line for no filter):

Only filtered results that contain less than 100 packages will have their contents displayed. Results with more than 100 packages will display a package count only.

Packages:

```

PyYAML-3.08-4.el5.src.rpm
grinder-0.0.57-1.el5.src.rpm
httpd-2.2.3-43.el5_5.3.src.rpm
libyaml-0.1.2-3.el5.src.rpm
m2crypto-0.16-6.1.el5_5.1.src.rpm
mod_python-3.3.1-12.el5.src.rpm
python-hashlib-20081119-5.el5.src.rpm
python-pycurl-7.15.5.1-4.el5.src.rpm
rh-cds-0.27-1.el5_5.src.rpm
rh-rhua-0.91-1.el5_5.src.rpm
rh-rhua-0.95-1.el5_5.src.rpm
rh-rhui-tools-0.76-1.el5_5.src.rpm
rpm-4.4.2.3-20.el5_5.1.src.rpm

```

```

-----
-----
rhui (repo) =>

```

3. Alternatively, type the first few letters of the RPM name you are looking for to filter the results.

Enter the first few characters (case insensitive) of an RPM to filter the results (blank line for no filter):
py

Only filtered results that contain less than 100 packages will have their contents displayed. Results with more than 100 packages will display a package count only.

```

Packages:
  PyYAML-3.08-4.el5.src.rpm
  python-hashlib-20081119-5.el5.src.rpm
  python-pycurl-7.15.5.1-4.el5.src.rpm

```

```

-----
-----
rhui (repo) =>

```

2.1. USING CUSTOM REPOSITORIES

Procedure 2.6. Create a Custom Repository

1. From the Repository Management screen, type **c** at the prompt to create a custom repository.

```
rhui (repo) => c
```

2. Enter a unique ID for the repository. Only alphanumeric characters, **_** (underscore), and **-** (hyphen) are permitted. You cannot use spaces in the unique ID. For example: **repo1**, **repo_1**, and **repo-1** are all valid entries.

```
Unique ID for the custom repository (alphanumerics, _, and - only):
repo_1
```

3. Enter a display name for the repository. This name is used to identify the repository within Red Hat Update Infrastructure Manager.

```
Display name for the custom repository [repo_1]:
Custom Repository 1
```

4. Specify the path that will host the repository. The path must be unique across all repos hosted by the Red Hat Update Infrastructure. For example: if you specify the path at this step as **foo/bar/baz** then the repository will be located at **https://<server>/pulp/repos/foo/bar/baz**.

```
Path at which the repository will be served [repo_1]:
/custom/i386
```

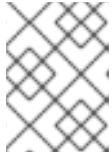
5. Select the checksum type to be used for the repository metadata.

```

Algorithm to use when calculating the checksum values for repository
metadata:
* Select "sha256" for RHEL6:
* Select "sha1" for either RHEL5 or RHEL6:

  1 - sha256
  2 - sha1
Enter value (1-2) or 'b' to abort: 1

```

**NOTE**

The current Red Hat Update Appliance Manager menu presents sha256 as default, and not sha1.

6. Choose whether or not to protect the new repository. If you answer **no** to this question, any client will be able to access the repository. If you answer **yes** then only clients with an appropriate entitlement certificate will be able to access the repository.

```
Should the repository require an entitlement certificate to access?
(y/n)
y
```

7. If you choose to protect the new repository, Red Hat Update Infrastructure Manager will ask for the entitlement path. It will also make a suggestion for the entitlement path, based on the repository's relative path.

Client entitlement certificates contain the download URLs that they are allowed to access. Red Hat Update Infrastructure analyzes the contents of the certificate to determine if the repository requested matches any of the permitted URLs, which determines whether or not it will allow the client to authenticate. For example: if an entitlement certificate grants access to `/foo/bar/baz`, and the request is made to a repository located at `https://server/pulp/repos/foo/bar/baz/os/reodata` Red Hat Update Infrastructure will approve the request and grant the authentication, because the path begins with one of the entitled download URLs. The URL only needs to begin with the correct information, it does not need to match exactly.

Entitlements can also contain variables, as long as `yum` knows the value for the variable. The two most common variables to use are `$basearch` and `$releasever`, which are populated with details of the client making the request. For example: if an entitlement certificate grants access to `/foo/$basearch/bar`, and the request is made to a repository located at `https://server/pulp/repos/foo/i386/bar` Red Hat Update Infrastructure will approve the request and grant the authentication, because the path matches when the variable is populated.

Red Hat Update Infrastructure Manager will suggest a path to use, based on the variables you used when you gave it a path for the repository. Leave the field blank to accept the suggested path. In this example, the path was given as `/custom/i386`, so Red Hat Update Infrastructure Manager has suggested that the `/i386` parameter be replaced with `$basearch` instead.

```
Based on the repository's relative path, the suggested entitlement
path is:
/custom/$basearch
```

```
The following repositories have the same entitlement path as
suggested:
Custom Repository 2
```

```
Path that should be used when granting an entitlement for this
repository. This
may use yum variable substitutions (e.g. $basearch) to group this
together with
other repositories that share the entitlement [/custom/$basearch].
```

- Red Hat Update Infrastructure Manager will ask if you want gpg signature turned on for content in that repository. If you input 'y', you will be asked if the content will be signed by Red Hat. Answering yes will include Red Hat's gpg key in the repo config. You are then asked if the content will be signed by a custom gpg key. Answering yes will prompt for a path to a public gpg key to include in the repo config. You can continue entering multiple paths to public gpg keys.

```

Should the repository require clients to perform a GPG check and
verify packages are signed by a GPG key? (y/n)
y

Will the repository be used to host any Red Hat GPG signed content?
(y/n)
y

Will the repository be used to host any custom GPG signed content?
(y/n)
y

Enter the absolute path to the public key of the GPG keypair:
/tmp/rhuitest1.gpg

Would you like to enter another public key? (y/n)
y

Enter the absolute path to the public key of the GPG keypair:
/tmp/rhuitest2.gpg

Would you like to enter another public key? (y/n)
n

```

- The details of the new repository will be displayed. Type y at the prompt to confirm the information and create the repository.

```

ID:                custom_repo_1
Name:              Custom Repository 1
Path:              /custom/i386
Entitlement:       /custom/$basearch
GPG Check         Yes
Custom GPG Keys:  '/tmp/rhuitest1.gpg', '/tmp/rhuitest2.gpg'
Red Hat GPG Key:  Yes
Proceed? (y/n) y

Successfully created repository Custom Repository 1

-----
-----
rhui (repo) =>

```

Procedure 2.7. Upload Packages to a Custom Repository

- It is possible to upload multiple packages at a time, and to upload to more than one repository at a time. Packages are uploaded to the Red Hat Update Appliance immediately, but will not be available on the CDS instance until the next time the CDS is synchronized.

From the Repository Management screen, type **u** at the prompt to upload new packages to a particular repository.

```
rhui (repo) => u
```

2. A list of all available custom repositories will be displayed. Packages cannot be uploaded to Red Hat repositories. Select which repositories to add the packages to by typing the number of the repository at the prompt. Typing the number of a repository will place a checkmark next to the name of that repository. Continue until all repositories you wish to add to have been checked, and then type **c** at the prompt to confirm.

```
Select the repositories to upload the package into:
- 1 : Custom Repository 1

Enter value (1-1) to toggle selection, 'c' to confirm selections, or
'?'
for more commands: 1

Select the repositories to upload the package into:
x 1 : Custom Repository 1

Enter value (1-1) to toggle selection, 'c' to confirm selections, or
'?'
for more commands: c
```

3. Specify the location of the RPMs to be uploaded. This can be a single `.rpm` file, or it can be a directory containing several `.rpm` files. If you specify a directory, all `.rpm` files in that directory will be uploaded.

```
Enter the location of the packages to upload. If the location is an
RPM,
the file will be uploaded. If the location is a directory, all RPMs
in that
directory will be uploaded.
/home/example/data/rpms
```

4. The details of the new packages to be uploaded will be displayed. Type **y** at the prompt to confirm the information and upload the packages.

```
The following RPMs will be uploaded:
emoticons-0.1-2.x86_64.rpm
feedless-1.0-1.noarch.rpm
origin-1.0-1.noarch.rpm
parent-1.0-1.noarch.rpm
patb-0.1-2.x86_64.rpm
Proceed? (y/n) y
```



NOTE

The packages will be uploaded to the Red Hat Update Appliance immediately, but will not be available on the CDS until the CDS has performed a synchronization.

Procedure 2.8. Delete Packages from a Custom Repository

1. Run the following command to save the CA certificate used for authentication. The certificate will be valid for 7 days.

```
# pulp-admin auth login -u USERNAME -p PASSWORD
```

2. Run the following command to remove a package from a custom repository:

```
# pulp-admin repo remove_package --id repo id -p package file name
```

3. You will be prompted to remove any packages that may depend on the package you are removing. Answer **yes** or **no**.

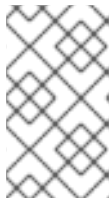
4. Regenerate the yum repository metadata for the custom repository:

```
# pulp-admin repo generate_metadata --id repo id
```

5. Use the following command to check on the status of the regenerate metadata task:

```
# pulp-admin repo generate_metadata --id repo id --status
```

6. Synchronize the CDS to remove packages on the CDS. See [Procedure 9.5, “Trigger a CDS Synchronization”](#)



NOTE

The steps above do not remove packages from `/var/lib/pulp/packages` on Red Hat Update Appliance and `/var/lib/pulp-cds/packages` on CDS. To remove packages from the two locations, use the `rm` command:

```
# cd /var/lib/pulp/packages  
# rm package file name
```

```
# cd /var/lib/pulp-cds/packages  
# rm package file name
```

[Report a bug](#)

CHAPTER 3. USING CONTENT DELIVERY SERVER INSTANCES

The CDS Management screen is used to list, add, and delete Content Delivery Server (CDS) instances. From this screen, you can also access the CDS Repository Management screen, which allows you to view and change information for individual CDS instances.

To access the CDS Management screen, go to the Home screen and type `c` at the prompt.

```
rhui (home) => c
-----
-----
                -= Red Hat Update Infrastructure Management Tool -=

-= Content Delivery Server (CDS) Management =-

l  list all CDS clusters and instances managed by the RHUI
i  display detailed information on a CDS cluster
a  register (add) a new CDS instance
m  move CDS(s) to a different cluster
d  unregister (delete) a CDS instance from the RHUI
s  associate a repository with a CDS cluster
u  unassociate a repository from a CDS cluster

logout
    removes stored authentication credentials and exits

<  move to the previous screen
^   home, move to the home screen
/   clear, clears the screen
?   help, display help
q   quit, exit

                                                    Connected:
rhua.example.com
-----
-----
rhui (cds) =>
```

Procedure 3.1. List CDS instances

- From the CDS Management screen, type `l` at the prompt to list the CDS instances currently being managed by the Red Hat Update Infrastructure.

```
rhui (cds) => l

-= RHUI Content Delivery Servers =-

new-york
-----
    Content Deliver Servers
    -----
    CDS01 New York
```

```

CDS02 New York

san-francisco
-----
Content Deliver Servers
-----
CDS01 San Francisco
CDS02 San Francisco

-----
-----
rhui (cds) =>

```

Procedure 3.2. View CDS Logs

- To view the Pulp specific logs see the log file located at `/var/log/pulp-cds/pulp.log`

To view the Grinder specific logs see the log file located at `/var/log/pulp-cds/grinder.log`

To view the Gofer specific logs see the log file located at `/var/log/gofer/agent.log`

Procedure 3.3. Display a CDS Cluster

1. From the CDS Management screen, type `i` at the prompt to display detailed information on a CDS cluster.

```

rhui (cds) => i

Select one or more CDS clusters:

- 1 : new-york
- 2 : san-francisco
Enter value (1-2) to toggle selection, 'c' to confirm selections, or
'?' for more commands: 1

Select one or more CDS clusters:

x 1 : new-york
- 2 : san-francisco
Enter value (1-2) to toggle selection, 'c' to confirm selections, or
'?' for more commands: c

```

2. Typing the number of a CDS cluster will place a checkmark next to your selection. Continue until all CDS clusters you wish to display to have been checked, then type `c` at the prompt to confirm.
3. The detailed information of the selected CDS cluster will be displayed.

```

-= RHUI CDS Clusters =-

new-york
-----

CDS Instances

```



```

-----
CDS01 New York
  hostname:          cds01-ny.domain.com
  client hostname:  cds01-ny.domain.com
CDS02 New York
  hostname:          cds02-ny.domain.com
  client hostname:  cds02-ny.domain.com

Repositories
-----
Red Hat Update Infrastructure 2 (RPMs) (6Server-i386)
Red Hat Update Infrastructure 2 (RPMs) (6Server-x86_64)

-----
-----
rhui (cds) =>

```

Procedure 3.4. Add a CDS Instance

1. From the CDS Management screen, type **a** at the prompt to add a new CDS instance.

```
rhui (cds) => a
```



IMPORTANT

Ensure that the CDS you are adding is configured and running. If the CDS is not running before you try to register it, the registration will fail.

2. Enter the hostname of the CDS to be added. This must match the hostname configured on the CDS itself. You cannot use the IP address. If the hostname does not match exactly, the operation will fail.

```

Hostname of the CDS to register:
cds01-london.domain.com

```

```

Client hostname (hostname clients will use to connect to the CDS)
[cds01-london.domain.com]:
cds01-london.client.domain.com

```



NOTE

If you are not sure what the hostname is, run the `hostname` command on the CDS instance to check.

3. Enter a display name for the CDS. This name is used to identify the CDS within Red Hat Update Infrastructure Manager. Alternatively, leave the field blank to use the hostname as the display name.

```

Display name for the CDS [cds01-london.domain.com]:
CDS01 London

```

4. Select an existing CDS cluster or Create a new CDS cluster.

```
Select a CDS cluster or enter a new one:
 1 - *** Create a new cluster ***
 2 - new-york
 3 - san-francisco
Enter value (1-3) or 'b' to abort: 1

Enter a CDS cluster name:
london
```

5. The details of the new CDS will be displayed. Type y at the prompt to confirm the information and create the CDS:

```
The following CDS instance will be registered:
  Hostname: cds01-london.domain.com
  Client Hostname: cds01-london.client.domain.com
  Name:      CDS01 London
  Cluster:   london
Proceed? (y/n) y

Successfully registered CDS01 London
-----
-----
rhui (cds) => 1
```

**NOTE**

If adding the CDS fails, check that the CDS daemon `pulp-cds` is running and that the firewall rules permit access between the Red Hat Update Appliance and the CDS.

Procedure 3.5. Move a CDS Instance

1. From the CDS Management screen, type `m` at the prompt to move a CDS instance to a different CDS instance cluster.

```
rhui (cds) => m
```

2. A list of all CDS instances currently being managed by the Red Hat Update Infrastructure will be displayed. Select the CDS instances that you wish to move by typing the number of the CDS at the prompt. Typing the number of a CDS will place a checkmark next to the name of that CDS. Continue until all CDS instances you wish to delete have been checked, and then type `c` at the prompt to confirm.

```
rhui (cds) => m
Select one or more Content Delivery Servers to move to a different
cluster:
- 1 : CDS01 London
- 2 : CDS01 New York
- 3 : CDS01 San Francisco
- 4 : CDS02 New York
```

```

- 5 : CDS02 San Francisco
Enter value (1-5) to toggle selection, 'c' to confirm selections, or
'?' for more commands: 2

Select one or more Content Delivery Servers to move to a different
cluster:
- 1 : CDS01 London
x 2 : CDS01 New York
- 3 : CDS01 San Francisco
- 4 : CDS02 New York
- 5 : CDS02 San Francisco
Enter value (1-5) to toggle selection, 'c' to confirm selections, or
'?' for more commands: c

Select a cluster in which to move the CDS's:
1 - *** Create a new cluster ***
2 - london
3 - new-york
4 - san-francisco
Enter value (1-4) or 'b' to abort: 4

```

3. A list of the CDS instances to be moved will be displayed. Type **y** at the prompt to confirm the movement of the CDS instances.

```

The following Content Delivery Servers will be moved to the san-
francisco cluster:

Content Delivery Servers
-----
CDS01 New York

Proceed? (y/n) y

Moving CDS01 New York to the san-francisco cluster...
... successfully moved CDS

-----
-----
rhui (cds) => 1

```

Procedure 3.6. Delete a CDS Instance



WARNING

When a CDS is deleted, all of the repositories and data will be lost. It is important that Red Hat Update Infrastructure Manager can access the CDS to ensure that content is no longer being hosted from the CDS instance before it is deleted.

1. From the CDS Management screen, type **d** at the prompt to delete a CDS instance.

```
rhui (cds) => d
```

- A list of all CDS instances currently being managed by the Red Hat Update Infrastructure will be displayed. Select which CDS instances to delete by typing the number of the CDS at the prompt. Typing the number of a CDS will place a checkmark next to the name of that CDS. Continue until all CDS instances you wish to delete have been checked, and then type `c` at the prompt to confirm.

```
Select the CDS Cluster that contains the instance(s) to be removed
from the RHUI:
```

```
 1 - london
 2 - new-york
 3 - san-francisco
```

```
Enter value (1-3) or 'b' to abort: 1
```

```
Select the CDS instances from the london cluster to be removed from
the RHUI:
```

```
- 1 : CDS01 London
- 2 : CDS02 London
```

```
Enter value (1-2) to toggle selection, 'c' to confirm selections, or
'?' for more commands: 1
```

```
Select the CDS instances from the london cluster to be removed from
the RHUI:
```

```
x 1 : CDS01 London
- 2 : CDS02 London
```

```
Enter value (1-2) to toggle selection, 'c' to confirm selections, or
'?' for more commands: c
```

- A list of the CDS instances to be deleted will be displayed. Type `y` at the prompt to confirm the selection and delete the CDS instances.

```
The following CDS instances from the london cluster will be
unregistered:
```

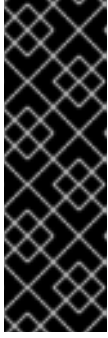
```
  CDS01 London
Proceed? (y/n) y
```

```
Unregistering CDS01 London...
```

```
-----
-----
```

```
rhui (cds) => >
```

- Red Hat Update Infrastructure Manager contacts the CDS to ensure it is properly cleaned up when it is deleted. Red Hat Update Infrastructure Manager is unable to delete a CDS instance that it cannot communicate with. However, there are some cases where you might need to remove an unresponsive CDS instance.



IMPORTANT

CDS instances exchange a secret with the Red Hat Update Appliance so that only the Red Hat Update Appliance is able to communicate with it. This secret is removed as part of the cleanup operation when a CDS is deleted normally. When a CDS instance is forcibly removed without being properly cleaned up, this secret is not deleted. You will not be able to register the CDS again until that secret has been removed. You can do this manually by deleting the `/var/lib/pulp-cds/.gofer/secret` file.

When you attempt to perform a delete operation on a CDS instance that is not running, Red Hat Update Infrastructure Manager will notify you that it is unable to delete the CDS, and ask if you want to forcibly remove the CDS. Type `y` to forcibly remove the unresponsive CDS.



WARNING

Be very careful with this command. If you delete a CDS instance that is not running, Red Hat Update Infrastructure Manager cannot confirm that the CDS instance is not hosting repositories that have previously been deployed.

```

-----
-----
rhui (cds) => d

Select the CDS instances to be removed from the RHUI:
  - 1 : CDS 1
Enter value (1-1) to toggle selection, 'c' to confirm selections, or
'?' for
more commands: 1

Select the CDS instances to be removed from the RHUI:
  x 1 : CDS 1
Enter value (1-1) to toggle selection, 'c' to confirm selections, or
'?' for
more commands: c

All previously deployed repositories will be deleted when a CDS is
unregistered.

The following CDS instances will be unregistered:
  CDS 1
Proceed? (y/n) y

Unregistering CDS 1...

The following CDS instances could not be contacted to be
unregistered.
```

```

They may still be removed from the RHUI database, however the RHUI
is unable
confirm that they have stopped hosting repositories.
CDS 1

Forcibly remove these CDS instances? (y/n)y

Forcibly unregistering CDS 1...

```

Procedure 3.7. Associate a Repository with a CDS Cluster

1. From the CDS Management screen, type `s` at the prompt to associate a repository with a CDS cluster.

```

rhui (cds) => s

-= RHUI Content Delivery Server Clusters =-

Select the CDS Cluster for repo association:
 1 - london
 2 - new-york
 3 - san-francisco
Enter value (1-3) or 'b' to abort: 1

```

2. A list of repositories currently being managed by the Red Hat Update Infrastructure will be displayed. Select which repositories to associate with the CDS cluster by typing the number of the repositories at the prompt. Typing the number of the repository will place a checkmark next to the name of the repository. Continue until all repositories have been checked, and then type `c` to confirm.

```

Select one or more repositories to associate with the london
cluster:

Custom Repositories

Red Hat Repositories
- 1 : Red Hat Storage 2 VSA Beta (RPMs) from RHUI (x86_64)
- 2 : Red Hat Storage 2 VSA Beta (Source RPMs) from RHUI
(x86_64)
- 3 : Red Hat Update Infrastructure 2 (RPMs) (6Server-i386)
- 4 : Red Hat Update Infrastructure 2 (RPMs) (6Server-x86_64)
- 5 : Red Hat Update Infrastructure 2 (SRPMS) (6Server-i386)
- 6 : Red Hat Update Infrastructure 2 (SRPMS) (6Server-x86_64)

Enter value (1-6) to toggle selection, 'c' to confirm selections, or
'?' for more commands: 1-2

Select one or more repositories to associate with the london
cluster:

Custom Repositories

Red Hat Repositories
x 1 : Red Hat Storage 2 VSA Beta (RPMs) from RHUI (x86_64)
x 2 : Red Hat Storage 2 VSA Beta (Source RPMs) from RHUI

```

```
(x86_64)
- 3 : Red Hat Update Infrastructure 2 (RPMs) (6Server-i386)
- 4 : Red Hat Update Infrastructure 2 (RPMs) (6Server-x86_64)
- 5 : Red Hat Update Infrastructure 2 (SRPMS) (6Server-i386)
- 6 : Red Hat Update Infrastructure 2 (SRPMS) (6Server-x86_64)

Enter value (1-6) to toggle selection, 'c' to confirm selections, or
'?' for more commands: c
```

3. The repositories for association will be displayed. Type **y** at the prompt to confirm the information and associate the selected repository with the selected the CDS cluster.

```
The following repositories will be associated with the london
cluster:

Red Hat Repositories
  Red Hat Storage 2 VSA Beta (RPMs) from RHUI (x86_64)
  Red Hat Storage 2 VSA Beta (Source RPMs) from RHUI (x86_64)

Proceed? (y/n) y

Associating repository Red Hat Storage 2 VSA Beta (RPMs) from RHUI
(x86_64)...
... successfully associated repository
Associating repository Red Hat Storage 2 VSA Beta (Source RPMs) from
RHUI (x86_64)...
... successfully associated repository

Content will not be downloaded to the cluster until the next cluster
sync is run.

-----
-----
rhui (cds) =>
```

Procedure 3.8. Unassociate a Repository from a CDS Cluster

1. From the CDS Management screen, type **u** at the prompt to unassociate a repository from a CDS cluster.

```
rhui (cds) => u

-= RHUI Content Delivery Server Clusters =-

Select the CDS Cluster for repo unassociation:
  1 - london
  2 - new-york
  3 - san-francisco
Enter value (1-3) or 'b' to abort: 1
```

2. A list of repositories currently being managed by the Red Hat Update Infrastructure will be displayed. Select which repositories to unassociate from the CDS cluster by typing the number of the repositories at the prompt. Typing the number of the repository will place a checkmark

next to the name of the repository. Continue until all repositories have been checked, and then type **c** to confirm.

```
Select one or more repositories to unassociate from the london
cluster:
```

```
Custom Repositories
```

```
Red Hat Repositories
```

```
- 1 : Red Hat Storage 2 VSA Beta (RPMs) from RHUI (x86_64)
- 2 : Red Hat Storage 2 VSA Beta (Source RPMs) from RHUI
(x86_64)
```

```
Enter value (1-2) to toggle selection, 'c' to confirm selections, or
'?' for more commands: a
```

```
Select one or more repositories to unassociate from the london
cluster:
```

```
Custom Repositories
```

```
Red Hat Repositories
```

```
x 1 : Red Hat Storage 2 VSA Beta (RPMs) from RHUI (x86_64)
x 2 : Red Hat Storage 2 VSA Beta (Source RPMs) from RHUI
(x86_64)
```

```
Enter value (1-2) to toggle selection, 'c' to confirm selections, or
'?' for more commands: c
```

- The repositories for unassociation will be displayed. Type **y** at the prompt to confirm the information and unassociate the selected repository from the selected the CDS cluster.

```
The following repositories will be unassociated from the london
cluster:
```

```
Red Hat Repositories
```

```
Red Hat Storage 2 VSA Beta (RPMs) from RHUI (x86_64)
Red Hat Storage 2 VSA Beta (Source RPMs) from RHUI (x86_64)
```

```
Proceed? (y/n) y
```

```
Unassociating repository Red Hat Storage 2 VSA Beta (RPMs) from RHUI
(x86_64)...
```

```
... successfully unassociated repository
```

```
Unassociating repository Red Hat Storage 2 VSA Beta (Source RPMs)
from RHUI (x86_64)...
```

```
... successfully unassociated repository
```

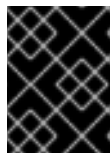
```
-----
-----
rhui (cds) =>
```


**NOTE**

Content will only be deleted from the cluster when the next cluster sync is run.

Procedure 3.9. Add a New CDS Log

1. Open `etc/pulp/logging/cds.cfg` in a text editor.

**IMPORTANT**

Do not leave spaces between commas as this will cause problems with Python 2.4.

2. Add a new entry to keys under `[loggers]` for the new log.

```
[loggers]
keys: root,pulp,gofer,grinder,qpid,newlog
```

3. Add a new section under the other loggers that starts with `loggers_` followed by the name of the new log.

```
[logger_newlog]
```

4. Set whether the log messages should only be handled by this log or passed to higher up loggers. Set `propagate` to 0 to prevent the messages from being passed higher up or to 1 to allow the messages to be passed higher up.

```
propagate: 0
```

5. Set the level. The level can be set to **DEBUG**, **INFO**, **WARNING**, **ERROR**, **CRITICAL** or **NOTSET**. **NOTSET** will log all messages.

```
level: NOTSET
```

6. Set the `qualname`. This is the module name in the Python file.

```
qualname: pulp
```

7. Select the handlers to be used.

```
handlers: pulp_file,console
```

8. Save and exit the file.

[Report a bug](#)

CHAPTER 4. USING CLIENT ENTITLEMENTS

The Create Certificates and Configuration RPMs screen is used to create new entitlement certificates and configuration RPMs.

When Red Hat issues the original entitlement certificate, it will grant access to the repositories you requested. When you create client entitlement certificates, you will need to decide how to sub-divide your clients, and create a separate certificate for each one. Each certificate can then be used to create individual RPMs for installation on the appropriate guest images. For example, you might create separate certificates for clients that require access to Red Hat Enterprise Linux 5 and those that require access to Red Hat Enterprise Linux 5 and JBoss channels.

To access the Create Certificates and Configuration RPMs screen, go to the Home screen and type `e` at the prompt:

```

-----
-----
                -= Red Hat Update Infrastructure Management Tool -=

-= Client Entitlement Management -=

e   generate an entitlement certificate
c   create a client configuration RPM from an entitlement certificate

logout
    removes stored authentication credentials and exits
<  move to the previous screen
^   move to the home screen
?   display help
q   quit, exit

                                                    Connected:
rhua.example.com
-----
-----
rhui (client) =>

```

Procedure 4.1. Generating an Entitlement Certificate

1. From the Create Certificates and Configuration RPMs screen, type `e` at the prompt to generate a new Red Hat entitlement certificate:

```
rhui (client) => e
```

2. A list of all available repositories will be displayed. This includes all custom repositories, and all products that have been granted entitlements in the content certificate that Red Hat granted. Select which repositories to include in the entitlement certificate by typing the number of the repository at the prompt. Typing the number of a repository will place a checkmark next to the name of that repository. Continue until all repositories you wish to add have been checked, and then type `c` at the prompt to confirm.

Repositories that are shown with an asterisk (*) indicates that they are deployed in the Red Hat Update Infrastructure.

Select the CDS cluster that clients using this entitlement certificate will use to download content:

- 1 - london
- 2 - new-york
- 3 - san-francisco

Enter value (1-3) or 'b' to abort: 1

Select one or more repositories to include in the entitlement certificate:

(an * next to a Red Hat repository indicates it is deployed in the RHUI)

Custom Repositories

Red Hat Repositories

- 1 : Red Hat Storage 2 VSA Beta (RPMs) from RHUI
- 2 : Red Hat Storage 2 VSA Beta (Source RPMs) from RHUI

Enter value (1-2) to toggle selection, 'c' to confirm selections, or '?' for more commands: a

Select one or more repositories to include in the entitlement certificate:

(an * next to a Red Hat repository indicates it is deployed in the RHUI)

Custom Repositories

Red Hat Repositories

- x 1 : Red Hat Storage 2 VSA Beta (RPMs) from RHUI
- x 2 : Red Hat Storage 2 VSA Beta (Source RPMs) from RHUI

Enter value (1-2) to toggle selection, 'c' to confirm selections, or '?' for more commands: c

3. Enter a name for the certificate. This name is used to identify the certificate within Red Hat Update Infrastructure Manager, and is also used to generate the name of the certificate and key files.

Name of the certificate. This will be used as the name of the certificate file (name.crt) and its associated private key (name.key). Choose something that will help identify the products contained with it:
london

4. Enter a path to save the certificate to. Leave the field blank to save to the current working directory:

Local directory in which to save the generated certificate [current directory]:
/tmp/

5. Enter the number of days the certificate should be valid for. Leave the field blank for 365 days:

```
Number of days the certificate should be valid [365]:
```

6. The details of the repositories to be included in the certificate will be displayed. Type `y` at the prompt to confirm the information and create the entitlement certificate.

If your CA private key requires a pass phrase to sign certificates, enter it at the prompt.

```
Repositories to be included in the entitlement certificate:
```

```
Red Hat Repositories
  Red Hat Enterprise Linux 5 Update Infrastructure MRG 2 (Debug)
  Red Hat Enterprise Linux 5 Update Infrastructure MRG 2 (RPMs)
```

```
Proceed? (y/n) y
```

```
.....+++
```

```
.....+++
```

```
Entitlement certificate created at /tmp/london.crt
```

```
-----
```

```
-----
```

```
rhui (client) =>
```

Procedure 4.2. Create a Client Configuration RPM

1. From the Create Certificates and Configuration RPMs screen, type `c` at the prompt to create a client configuration RPM:

```
rhui (client) => c
```

2. Enter a path to a local directory to save the configuration files to. Leave the field blank to save to the current working directory:

```
Full path to local directory in which the client configuration files
generated by this tool
should be stored (if this directory does not exist, it will be
created):
/tmp/
```

3. Enter a name for the RPM. Do not specify the `.rpm` extension:

```
Name of the RPM:
london-cc
```

4. Enter a version number for the configuration RPM. Leave the field blank to use version 2:

```
Version of the configuration RPM [2]:
```

5. Enter the full path to the entitlement certificate to be used (the entitlement certificate is generated in [Procedure 4.1, “Generating an Entitlement Certificate”](#)):

```
Full path to the entitlement certificate authorizing the client to
access
specific channels:
/tmp/london.crt
```

6. Enter the full path to the private key to be used with the entitlement certificate (the private key is generated in [Procedure 4.1, “Generating an Entitlement Certificate”](#)):

```
Full path to the private key for the above entitlement certificate:
/tmp/london.key
```

7. All CDS instances are able to function as load balancers. You will be required to nominate one CDS as a primary load balancer, however if that CDS becomes unavailable, or is unable to function as a load balancer, load balancing tasks will fall to the other available CDS instances. Select a CDS instance to be the primary load balancer for the client:

```
Select the CDS cluster that clients will use to download content:
```

- ```
1 - london
2 - new-york
3 - san-francisco
```

```
Enter value (1-3) or 'b' to abort: 1
```

```
Select the CDS instance that should be the primary load balancer for
the
client. All other CDS instances will be listed as back up load
balancers
in the client's mirror list:
```

- ```
1 - cds02-london.client.domain.com
2 - cds01-london.client.domain.com
```

```
Enter value (1-2) or 'b' to abort: 1
```

A list of the CDS instances to be used for load balancing will be displayed, in priority order:

```
Load Balancer Order:
cds02-london.client.domain.com
cds01-london.client.domain.com
```

8. A list of all unprotected repositories will be displayed. Select which repositories (if any) to include in the RPM by typing the number of the repository at the prompt. Typing the number of a repository will place a checkmark next to the name of that repository. Continue until all repositories you wish to add have been checked, and then type `c` at the prompt to confirm. If you don't want to add any repositories to the RPM, leave all repositories unchecked and type `c` at the prompt to confirm.

Unprotected repositories that are added to the RPM will be included in the generated `.repo` file, along with the repository definitions for all entitlements included in the certificate.

```
Select any unprotected repositories to be included in the client
configuration:
```

- ```
- 1 : Unprotected Repo 1
```

```
Enter value from (1-1) to toggle selection, 'c' to confirm
```

```
selections, or '?'
for more commands: 1
```

```
Select any unprotected repositories to be included in the client
configuration:
```

```
 x 1 : Unprotected Repo 1
Enter value from (1-1) to toggle selection, 'c' to confirm
selections, or '?'
for more commands: c
```

**9. The client configuration RPM will be created, and the location of the file displayed:**

```
Successfully created client configuration RPM.
RPMs can be found at /tmp/
```

```


rhui (client) =>
```

[Report a bug](#)

## CHAPTER 5. MANAGING ENTITLEMENT CERTIFICATES

The Entitlements Manager screen is used to list entitled products in the current Red Hat content certificates and to upload new certificates.

To access the Entitlements Manager screen, go to the Home screen and type `n` at the prompt:

```
rhui (entitlements) => n

 -= Red Hat Update Infrastructure Management Tool -=

-= Entitlements Manager -=

l list Red Hat content certificate entitlements
c list custom repository entitlements
u upload a new or updated Red Hat content certificate

 Connected:
rhua.example.com

rhui (entitlements) =>
```

### Procedure 5.1. List Certificate Information

- From the Entitlements Manager screen, type `l` at the prompt to list data about the current content certificate.

Red Hat Update Infrastructure Manager will give the following information about the certificate:

```
rhui (entitlements) => l

Red Hat Entitlements

Valid
 Red Hat Enterprise Linux Server (RPMs)
 Expiration: 01-31-2012 Certificate: big-content-cert.pem

 Red Hat Enterprise Linux Server (SRPMS)
 Expiration: 01-31-2012 Certificate: big-content-cert.pem

 Red Hat Enterprise Linux Server 6 Releases (RPMs)
 Expiration: 01-31-2012 Certificate: big-content-cert.pem

 Red Hat Enterprise Linux Server 6 Releases (SRPMS)
 Expiration: 01-31-2012 Certificate: big-content-cert.pem

 Red Hat Enterprise Linux Server 6 Updates (RPMs)
 Expiration: 01-31-2012 Certificate: big-content-cert.pem

 Red Hat Enterprise Linux Server 6 Updates (SRPMS)
```

```

Expiration: 01-31-2012 Certificate: big-content-cert.pem

Red Hat Update Infrastructure 1.2 (RPMs)
Expiration: 01-31-2012 Certificate: big-content-cert.pem

Red Hat Update Infrastructure 1.2 (SRPMS)
Expiration: 01-31-2012 Certificate: big-content-cert.pem

Expired
Red Hat Update Infrastructure 1.1 (RPMs)
Expiration: 06-25-2011 Certificate: content-cert.pem

rhui (entitlements) =>

```

### Procedure 5.2. List Custom Repository Entitlements

- From the Entitlements Manager screen, type `c` at the prompt to list data about the custom repository entitlements.

```

rhui (entitlements) => c

Custom Repository Entitlements

For each entitlement URL listed, the corresponding repositories
that are configured with that entitlement are listed.

/protected/$basearch/os
- Name: Repo 1
 URL: protected/i386/os
- Name: Repo 2
 URL: protected/x86_64/os

rhui (entitlements) =>

```

### Procedure 5.3. Upload Content Certificate

Red Hat might need to issue a new content certificate if your old content certificate is about to expire, or to change the entitlements of the certificate. If Red Hat issues a new content certificate, it will need to be uploaded to Red Hat Update Infrastructure.



#### NOTE

When you upload a new content certificate, it will also be updated in the Red Hat Update Appliance, and will be used for synchronizing Red Hat repositories. For this reason, do not upload a new content certificate before it becomes valid, as it will cause your synchronizations to fail until the valid date is reached.

If two or more content certificates provide the same entitlements, the certificate with an expiration date furthest in the future will be used.



Red Hat Update Infrastructure Manager expects that both the content certificate and its private key will be contained in the same file. If you have existing content certificates with separate keys, you can create the single file using the `cat` command at a shell prompt:

```
cat content.crt content.key > content-2.pem
```

1. From the Entitlements Manager screen, type `u` at the prompt to upload a new content certificate:

```
rhui (entitlements) => u
```

### IMPORTANT

Content certificates are stored on the same system Red Hat Update Infrastructure Manager is installed on, at `/etc/pki/rhui`. For security reasons, this directory requires root permissions. If you do not have the correct permissions, Red Hat Update Infrastructure Manager will not allow you to proceed:

```
rhui (entitlements) => u
```

```
The current user does not have write privileges to update
the
content cert at /etc/pki/rhui/content-cert.crt.
```

```


```

```
rhui (entitlements) =>
```

2. Enter the full path to the new content certificate:

```
Full path to the new content certificate:
/home/example/rhui-certs/cert-1.pem
```

3. The details of the new certificate to be uploaded will be displayed. Type `y` at the prompt to confirm the information and upload the packages. Red Hat Update Infrastructure Manager will then list the current certificates:

```
The RHUI will be updated with the following certificate:
```

```
Certificate:
/home/example/rhui-certs/cert-1.pem
```

```
Proceed? (y/n) y
```

```
Red Hat Entitlements
```

```
RHEL 5
Expiration: 06-06-2012 Certificate: cert-2.pem
```

```
RHUI 1.1
Expiration: 06-07-2012 Certificate: cert-1.pem
```

```
RHUI 1.2
Expiration: 06-07-2012 Certificate: cert-1.pem
```

```


rhui (entitlements) =>
```

[Report a bug](#)

## CHAPTER 6. USING IDENTITY CERTIFICATES

The Identity Certificate Management screen is used to create new entitlement certificates and configuration RPMs.

An identity certificate is used by Red Hat Update Infrastructure to authenticate the CDS to the Red Hat Update Appliance and secure the communication between them. Entitlement certificates grant a client access to a specific set of entitled repositories, but the Red Hat Update Appliance monitoring functionality and the load balancer require access to all repositories in the system. The identity certificate can be used to identify each component as part of Red Hat Update Infrastructure, so that the entitlement-checking procedure will not deny it access.

An identity certificate is generated for you the first time you use Red Hat Update Infrastructure Manager. When the identity certificate generated by that process expires, you will need to regenerate the certificate in order to continue using Red Hat Update Infrastructure.

Only one identity certificate is needed for the entire infrastructure.

To access the Identity Certificate Management screen, go to the Home screen and type `i` at the prompt:

```


 -= Red Hat Update Infrastructure Management Tool -=

-= Identity Certificate Management =-

 g generate a new identity certificate

 Connected:
rhua.example.com

rhui (client) =>
```

### Procedure 6.1. Generate a New Identity Certificate

1. From the Identity Certificate Management screen, type `g` at the prompt to generate a new identity certificate:

```
rhui (client) => g
```

2. Confirm that the new identity certificate will over-write the existing certificate by typing `y` at the prompt:

```
Generating a new RHUI identity certificate will replace
the one currently stored at /etc/pki/rhui/identity.crt. Proceed?
[y/n]: y
```

3. Enter the number of days that the identity certificate should be valid for. If left blank, this field will default to 3650 (ten years):

```
Enter the number of days the RHUI identity certificate will be
```

```
valid.
```

```
If the identity certificate ever expires, it will need to be
regenerated using rhui-manager [Default: 3650]:
```

4. The new identity certificate will be created. You should restart the service to pick up the changes:

```
.....+++
.....+++
Successfully regenerated RHUI Identity certificate
```

## 6.1. UPDATING THE ENTITLEMENT-SIGNING CA CERTIFICATE

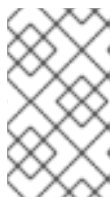
This section deals with the entitlement-signing CA certificate which is configured at the first launch of rhui-manager.

Before re-generating the entitlement-signing CA certificate, note that any client instances that have client configuration rpm's installed containing certificates signed by your existing entitlement-signing CA certificates will cease to work. These clients will need to be updated by installing new client configuration rpm's manually, or perhaps from an unprotected custom repository hosted in your Red Hat Update Infrastructure.

**Procedure 6.2. Follow these steps to update the entitlement-signing CA certificate and its private key**

1. Remove the following files from the `/etc/pki/rhui` directory. As a precaution, backup the files before deleting them.

```
entitlement-ca.crt
entitlement-ca-key.pem
entitlement-ca.srl
identity.crt
identity.key
```



### NOTE

The Identity certificate and its private key (`identity.crt` and `identity.key`) are removed because they are signed by the entitlement-signing CA certificate and thus must be regenerated.

2. The next time you start rhui-manager you will be prompted for the new path to the entitlement-signing CA certificate and key, and a new identity certificate and key will also be generated.

This is further detailed in [Procedure 12.1, “Updating Red Hat Update Infrastructure”](#).

[Report a bug](#)

## CHAPTER 7. USING QPID SSL CERTIFICATES

This chapter shows you how to update Qpid SSL certificates generated by the `nss-db-gen` script.

The following certificates generated by `nss-db-gen` have an expiration date:

- `/etc/pki/rhua/qpid-ca.crt`
- `/etc/pki/rhua/qpid-client.crt`

### Procedure 7.1. Checking Certificate Expiration Date

1. To determine the expiration date of any of the three certificates, run this command:

```
openssl x509 -text -in /etc/pki/rhua/$file | less
```

2. Go to the **Validity** section to see the expiration date.

```
Validity
 Not After : Sep 9 12:41:20 2013 GMT
```

### Procedure 7.2. Updating Qpid SSL Certificates

1. On the Red Hat Update Appliance server, edit the `nss-db-gen` script with your preferred text editor. Change the **VALID** section to 48:

```
vi /usr/bin/nss-db-gen
VALID="48"
```

2. Execute the `nss-db-gen` script.

```
nss-db-gen
```

3. Edit the latest answers file.

```
vi myanswersfile
```

Update the **version** parameter to a higher number. For example, if the **version** is 1.1, update it to 1.2:

```
[general]
version: 1.2
```

4. Execute `rhui-installer` using the updated answers file.

```
rhui-installer myanswersfile
```

5. Copy the updated configuration RPMs to the CDS servers.

```
scp /tmp/rhui/rh-cds1-config-1.2-2.el6.noarch.rpm root@host-
```

```
cds1.com:/root
scp /tmp/rhui/rh-cds2-config-1.2-2.el6.noarch.rpm root@host-
cds2.com:/root
```

6. On the Red Hat Update Appliance server, install the configuration RPM.

```
yum install /tmp/rhui/rh-rhua-config-1.2-2.el6.noarch.rpm
```

7. On the CDS servers, install the configuration RPM.

**CDS1:**

```
yum install rh-cds1-config-1.2-2.el6.noarch.rpm
```

**CDS2:**

```
yum install rh-cds2-config-1.2-2.el6.noarch.rpm
```

[Report a bug](#)

## CHAPTER 8. IMPLEMENTING A CERTIFICATION REVOCATION LIST

A Certificate Revocation List (CRL) is a list of certificates that have been revoked and should not be relied on. This chapter shows you how to implement a CRL in a Red Hat Update Infrastructure environment using the openssl x509 certificates.

### Procedure 8.1. Implementing a CRL list in a Red Hat Update Infrastructure environment

1. Make a directory for a CRL:

```
mkdir /etc/pki/pulp/content/crl
```

2. Create an index file with the following command:

```
touch /etc/pki/pulp/content/crl/index.txt
```

3. Create a file for the CRL number. This file should contain the text `00` only.

```
echo 00 > /etc/pki/pulp/content/crl/pulp_crl_number
```

4. In `/etc/pulp/`, create and write the following contents into a `crl_openssl.conf` file.

```
cat <<EOF > crl_openssl.conf
OpenSSL configuration for CRL generation
#
#####
[ca]
default_ca = CA_default # The default ca section

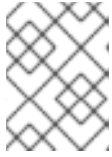
#####
[CA_default]
database = /etc/pki/pulp/content/crl/index.txt
crlnumber = /etc/pki/pulp/content/crl/pulp_crl_number

default_days = 365 # how long to certify for
default_crl_days= 30 # how long before next CRL
default_md = default # use public key default MD
preserve = no # keep passed DN ordering

#####
[crl_ext]
CRL extensions.
Only issuerAltName and authorityKeyIdentifier make any sense in a
CRL.
issuerAltName=issuer:copy
authorityKeyIdentifier=keyid:always,issuer:always
EOF
```

5. Create the CRL file with the following command:

```
openssl ca -gencrl -keyfile /home/example/ca.key -cert
/home/example/ca.crt -out /etc/pki/pulp/content/crl/pulp_crl.pem -
config /etc/pulp/crl_openssl.conf
```

**NOTE**

The CA cert and key are to be the same ones used to create the client entitlement RPMs.

**Procedure 8.2. Revoking a certificate**

1. Revoke a certificate with the following command:

```
openssl ca -revoke <Content certificate> -keyfile
/home/example/ca.key -cert /home/example/ca.crt -config
/etc/pulp/crl_openssl.conf
```

**Example 8.1. Example Revoking Certificate**

```
openssl ca -revoke /tmp/dir/rhui-client.crt -keyfile
/root/pem/ca.key -cert /root/pem/ca.crt -config
/etc/pulp/crl_openssl.conf
Using configuration from /etc/pulp/admin/crl_openssl.conf
Adding Entry with serial number 0102 to DB for /CN=Red Hat Update
Infrastructure
Revoking Certificate 0102.
Data Base Updated
```

2. Regenerate the CRL list with the following command:

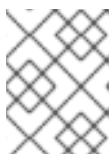
```
openssl ca -gencrl -keyfile /home/example/ca.key -cert
/home/example/ca.crt -out /etc/pki/pulp/content/crl/pulp_crl.pem -
config /etc/pulp/crl_openssl.conf
```

3. Test the CRL list with the following command:

```
cat /home/example/ca.crt /etc/pki/pulp/content/crl/pulp_crl.pem >
/tmp/test.pem
```

4. Verify the CRL list with the following command:

```
openssl verify -extended_crl -verbose -CAfile /tmp/test.pem -
crl_check <Content certificate>
```

**NOTE**

Code #23 indicates the certificate has been revoked. For more error codes, refer to [openssl.org](http://openssl.org).



5. In `/etc/pki/pulp/content`, create the CRL hash with the following command:

```
ln -s crl/pulp_crl.pem `openssl crl -hash -noout -in
crl/pulp_crl.pem`.r0
```

6. Copy the CRL information to each CDS instance.

```
tar -cf - crl *.r0 | ssh root@fqdn_cds 'cd /etc/pki/pulp/content &&
tar -xvf -'
```

[Report a bug](#)

## CHAPTER 9. STATUS AND SCHEDULING

Synchronization is used to keep data on CDS instances and repositories current. By default, a synchronization will occur every six hours, but this schedule can be adjusted. It is also possible to manually run a synchronization when required. Manual synchronizations are additional to the schedule, and do not change the scheduled synchronization times.

The Synchronization Status and Scheduling screen is used to display the synchronization schedules for CDS instances and repositories, and view past synchronizations. You can also trigger individual synchronizations from this screen.

Within the Synchronization Status and Scheduling screen, the following terms are used to describe the final status of the most recent synchronization:

**Table 9.1. Synchronization Status Descriptions**

| Status                    | Description                                                                                                                          |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <b>In Progress</b>        | A synchronization is in progress. You will not be able to trigger another synchronization until after the running one has completed. |
| <b>Awaiting Execution</b> | A synchronization is ready to run, but is waiting for other tasks to complete.                                                       |
| <b>Success</b>            | The last synchronization finished successfully.                                                                                      |
| <b>Error</b>              | The last synchronization failed.                                                                                                     |

To access the Synchronization Status and Scheduling screen, go to the Home screen and type `s` at the prompt:

```
rhui (home) => s

 -= Red Hat Update Infrastructure Management Tool -=

-= Synchronization Status =-

dr display repo sync summary
vr view the details of the last repository sync
sr sync an individual repository immediately
dc display CDS sync summary
vc view the details of the last CDS sync
sc sync an individual CDS immediately

 Connected:
rhua.example.com

rhui (sync) =>
```

**Procedure 9.1. List Repository Synchronization Status**

1. From the Synchronization Status and Scheduling screen, type `dr` at the prompt to see the synchronization status of all available repositories:

```

rhui (sync) => dr

 -= Red Hat Update Infrastructure Management Tool -=

-= Repository Synchronization Status =-

Last Refreshed: 13:53:52
(updated every 5 seconds, ctrl+c to exit)

Next Sync Last Sync Last
Result

Red Hat Enterprise Linux Server 6 Releases (RPMs) (6Server-i386)
In Progress 06-20-2011 14:33 Success

Red Hat Enterprise Linux Server 6 Releases (RPMs) (6Server-x86_64)
In Progress Never Error

Red Hat Update Infrastructure 1.2 (RPMs) (5Server-i386)
In Progress 06-20-2011 13:52 Success

Red Hat Update Infrastructure 1.2 (RPMs) (5Server-x86_64)
In Progress 06-20-2011 13:52 Success

Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-i386)
Awaiting Execution 06-20-2011 13:53 Success

Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-x86_64)
Awaiting Execution 06-20-2011 13:53 Success

 Connected:
rhua.example.com

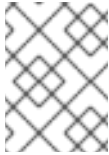

```

**Last Result**

Shows the result of the last synchronization. Possible values are **Never** if the repository has not been synchronized before, **Success** if the last synchronization completed successfully, or **Error** if the last synchronization did not complete successfully.

**Last Sync**

Shows the time the last synchronization occurred. Possible values are **Never** if the repository has not been synchronized before, or the date and time of the last synchronization.

**NOTE**

Regular scheduled synchronizations will occur even if a manual synchronization as been triggered.

**Next Sync**

Shows when the next synchronization will occur. If a synchronization is currently running, this column will display either **In Progress** if the synchronization is running, or **Awaiting Execution** if a synchronization is ready to run, but is waiting on other tasks to complete.

2. The display will continue to update every five seconds. Press **Ctrl+C** to return to the prompt.

**Procedure 9.2. View Last Synchronization Details**

1. From the Synchronization Status and Scheduling screen, type **vr** at the prompt to see details about the last repository synchronization:

```
rhui (sync) => vr
```

2. A list of all repositories currently being managed by the Red Hat Update Infrastructure will be displayed. Select which repository to view synchronization information for by typing the number of the repository at the prompt:

```
Select a repository:
1 - Red Hat Enterprise Linux Server 6 Releases (RPMs) (6Server-i386)
2 - Red Hat Enterprise Linux Server 6 Releases (RPMs) (6Server-x86_64)
3 - Red Hat Update Infrastructure 1.2 (RPMs) (5Server-i386)
4 - Red Hat Update Infrastructure 1.2 (RPMs) (5Server-x86_64)
5 - Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-i386)
6 - Red Hat Update Infrastructure 1.2 (SRPMS) (5Server-x86_64)
Enter value (1-6) or 'b' to abort: 1
```

3. The synchronization information for the selected repository will be displayed:

```
Start Time: 06-20-2011 13:52
Finish Time: 06-20-2011 14:33
Elapsed Time: 0:41:08
Result: Success
```

The **Elapsed Time** displays the amount of time the synchronization took to run, in the format **hours:minutes:seconds**. The **Result** field shows the result of the synchronization, and will display either **Success** if the last synchronization completed successfully, or **Error** if the last synchronization failed. If the last synchronization failed, Red Hat Update Infrastructure Manager will also display **Exception** a one line description of the error, and **Traceback**, a full traceback to be used when reporting the error.

**Procedure 9.3. List CDS Synchronization Status**

1. From the Synchronization Status and Scheduling screen, type `dc` at the prompt to see the synchronization status of all available repositories:

```

rhui (sync) => dc

 -= Red Hat Update Infrastructure Management Tool -=

-= CDS Synchronization Status -=

Last Refreshed: 16:08:54
(updated every 5 seconds, ctrl+c to exit)

cds00133 [UP
]
cds00175 [UP
]

Next Sync Last Sync Last
Result

cds00133
07-04-2011 16:23 07-04-2011 16:07 Success

cds00175
07-04-2011 16:24 07-04-2011 16:07 Success

 Connected:
rhua.example.com


```

2. The display will continue to update every five seconds. Press `Ctrl+C` to return to the prompt.

#### Procedure 9.4. Trigger a Repository Synchronization

1. From the Synchronization Status and Scheduling screen, type `sr` at the prompt to trigger a manual repository synchronization. This will trigger another synchronization in addition to the ordinary scheduled synchronizations:

```
rhui (sync) => sr
```

2. A list of available repositories will be displayed. Select which repositories to synchronize by typing the number of each repository at the prompt. Typing the number of a repository will place a checkmark next to the name of that repository. Continue until all repositories you want to synchronize have been checked, and then type `c` at the prompt to confirm:

```
Select one or more repositories to schedule to be synchronized
```

before its scheduled time.

The sync will happen as soon as possible depending on other tasks that may be executing in the RHUI. Only repositories that are not currently synchronizing are displayed.

```

 Last Result Next Sync Repository

- 1 : Success 07-04-2011 19:26 Red Hat Enterprise Linux
Server 6 Updates (RPMs) (6Server-x86_64)
- 2 : Success 07-04-2011 21:12 Red Hat Update Infrastructure
1.1 (RPMs) (5Server-i386)
- 3 : Success 07-04-2011 21:12 Red Hat Update Infrastructure
1.1 (RPMs) (5Server-x86_64)
x 4 : Success 07-04-2011 19:26 Red Hat Update Infrastructure
1.2 (RPMs) (5Server-i386)
x 5 : Success 07-04-2011 19:26 Red Hat Update Infrastructure
1.2 (RPMs) (5Server-x86_64)
Enter value (1-5) to toggle selection, 'c' to confirm selections, or
'?' for more commands: c

```

- A list of the repositories to be synchronized will be displayed. Type **y** at the prompt to confirm the selection and trigger the synchronizations:

```

The following repositories will be scheduled for synchronization:
 Red Hat Update Infrastructure 1.2 (RPMs) (5Server-i386)
 Red Hat Update Infrastructure 1.2 (RPMs) (5Server-x86_64)
Proceed? (y/n) y

```

```

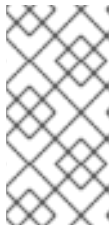
Scheduling sync for Red Hat Update Infrastructure 1.2 (RPMs)
(5Server-i386)...
... successfully scheduled for the next available timeslot.
Scheduling sync for Red Hat Update Infrastructure 1.2 (RPMs)
(5Server-x86_64)...
... successfully scheduled for the next available timeslot.

```

```


rhui (sync) =>

```



#### NOTE

Manual synchronizations will not always occur immediately, depending on other tasks the Red Hat Update Infrastructure is currently performing. The synchronization will occur as soon as the Red Hat Update Infrastructure has an available timeslot.

### Procedure 9.5. Trigger a CDS Synchronization

- From the Synchronization Status and Scheduling screen, type **sc** at the prompt to trigger a manual CDS synchronization. This will trigger another synchronization in addition to the ordinary scheduled synchronizations:

```
rhui (sync) => sc
```

Select one or more CDS instances to schedule to be synchronized before its scheduled time.

The sync will happen as soon as possible depending on other tasks that may be executing in the RHUI. Only CDS instances that are not currently synchronizing are displayed.

```

 Last Result Next Sync CDS

- 1 : Error 07-04-2011 16:23 cds00133
- 2 : Error 07-04-2011 16:24 cds00175
Enter value (1-2) to toggle selection, 'c' to confirm selections, or
'?' for more commands: 1-2

```

2. A list of available CDS instances will be displayed. Select which CDS instances to synchronize by typing the number of each CDS instance at the prompt. Typing the number of a CDS instance will place a checkmark next to the name of that CDS. Continue until all CDS instances you want to synchronize have been checked, and then type `c` at the prompt to confirm:

Select one or more CDS instances to schedule to be synchronized before its scheduled time.

The sync will happen as soon as possible depending on other tasks that may be executing in the RHUI. Only CDS instances that are not currently synchronizing are displayed.

```

 Last Result Next Sync CDS

x 1 : Error 07-04-2011 16:23 cds00133
x 2 : Error 07-04-2011 16:24 cds00175
Enter value (1-2) to toggle selection, 'c' to confirm selections, or
'?' for more commands: c

```

3. A list of the CDS instances to be synchronized will be displayed. Type `y` at the prompt to confirm the selection and trigger the synchronizations:

```

The following CDS instances will be scheduled for synchronization:
 cds00133
 cds00175
Proceed? (y/n) y

```

```

Scheduling sync for cds00133...
... successfully scheduled for the next available timeslot.
Scheduling sync for cds00175...
... successfully scheduled for the next available timeslot.

```

```

rhui (sync) =>

```

**NOTE**

Manual synchronizations will not always occur immediately, depending on other tasks the Red Hat Update Infrastructure is currently performing. The synchronization will occur as soon as the Red Hat Update Infrastructure has an available timeslot.

[Report a bug](#)



## CHAPTER 10. CONFIGURING USERS

The User screen is used to change the user password.



### IMPORTANT

Ensure you change the default password for the Red Hat Update Appliance before you begin using the system.

To access the User screen, go to the Home screen and type `u` at the prompt:

```
rhui (home) => u

 -= Red Hat Update Infrastructure Management Tool -=

-= User Manager =-

 p change a user's password

 Connected:
rhua.example.com

rhui (users) =>
```

### Procedure 10.1. Change User Password

1. From the User screen, type `p` at the prompt to change the user password:

```
rhui (users) => p
```

2. Enter the username of the user that requires the password change:

```
Username: admin
```

3. Enter the new password for the user, and re-enter it:

```
New Password:
Re-enter Password:
```

4. Red Hat Update Infrastructure Manager will update the password:

```
Password successfully updated.
```

```


rhui (users) =>
```

[Report a bug](#)

## CHAPTER 11. MONITORING

Red Hat Update Infrastructure Manager can display information about the state of the Red Hat Update Infrastructure installation in several different ways. The most human-readable information is available in the Synchronization Status and Scheduling screen, which is described in [Chapter 9, Status and Scheduling](#). Status and monitoring information can also be obtained in a machine-readable state, for use with automated monitoring solutions such as Nagios.

### Procedure 11.1. Monitoring the state of Red Hat Update Infrastructure

1. To retrieve a status code that provides a summary of the current state of the Red Hat Update Infrastructure, run the `rhui-manager` command from the shell prompt with the `status` command, and `echo` the result to standard output:

```
rhui-manager --username admin --password admin status
echo $?
0
```

This command should be automated to suit your individual requirements.



#### NOTE

It is recommended that this command always be used with the `--username` and `--password` switches specified, in order to bypass the authentication certificate and avoid the command failing due to expired credentials.

2. A status code of `0` indicates that the Red Hat Update Infrastructure is running correctly. Other values, such as listed in [Table 11.1, “Status Codes”](#), indicates an error exists. If multiple errors exists, the status codes are logically OR'd together.

**Table 11.1. Status Codes**

| Status Code | Description                                   |
|-------------|-----------------------------------------------|
| <b>0</b>    | Success                                       |
| <b>1</b>    | Repository synchronization error              |
| <b>2</b>    | CDS synchronization error                     |
| <b>4</b>    | CDS heartbeat error                           |
| <b>8</b>    | Identity certificate expiration warning       |
| <b>16</b>   | Identity certificate expiration error         |
| <b>32</b>   | Entitlement CA certificate expiration warning |
| <b>64</b>   | Entitlement CA certificate expiration error   |

## Log Files

- `~/ .rhui/rhui .log`

Red Hat Update Infrastructure Manager logs

- `/var/log/pulp/pulp .log`

Red Hat Update Appliance logs



### NOTE

The `pulp .log` file is automatically written to a new enumerated log file when the base file reaches 1MB in size. The most recent four versions are kept in addition to the currently written file.

- `/var/log/pulp/grinder .log`

Package synchronization output



### NOTE

The `grinder .log` file is automatically written to a new enumerated log file when the base file reaches 1MB in size. The most recent four versions are kept in addition to the currently written file.

- `/var/log/pulp-cds/gofer .log`

The `gofer .log` file is located on CDS instances, and logs information about errors related to the CDS function.

- `/var/log/gofer/agent .log`

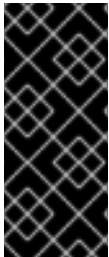
The `agent .log` file is located on CDS instances, and logs information about communication errors. This file contains information about whether the CDS instance was able to successfully connect to the Qpid broker running on the Red Hat Update Appliance.

[Report a bug](#)

## CHAPTER 12. UPDATING RED HAT UPDATE INFRASTRUCTURE

This chapter explains how to update your Red Hat Update Infrastructure installation.

Red Hat Update Infrastructure can be configured to provide and use a repository that will update the Red Hat Update Infrastructure installation itself. The repository can be created using Red Hat Update Infrastructure Manager, which can then generate an entitlement certificate and client configuration RPM. The RPM is then installed on the Red Hat Update Appliance and each CDS instance, and future updates can be downloaded and installed using the `yum` command.



### IMPORTANT

If updating from a version 2.1.1 and lower, follow the steps from 1 to 9 in this section and then see the additional update instructions in [Section 12.1, “Additional Instructions for Updating Red Hat Update Infrastructure 2.1.1 and Lower”](#).

Updates from version 2.1.2 and higher only require the steps in this section.

### Procedure 12.1. Updating Red Hat Update Infrastructure

1. Add the Red Hat Update Infrastructure repositories to the Red Hat Update Appliance. Use [Procedure 2.3, “Add a Red Hat Repository”](#) to install the following repositories:

```
Red Hat Update Infrastructure 2 (RPMs) (6Server-x86_64)
Red Hat Update Infrastructure 2 (RPMs) (6.0-i386)
Red Hat Update Infrastructure 2 (RPMs) (6.1-i386)
Red Hat Update Infrastructure 2 (RPMs) (6.1-x86_64)
Red Hat Update Infrastructure 2 (RPMs) (6Server-i386)
Red Hat Update Infrastructure 2 (RPMs) (6.0-x86_64)
Red Hat Update Infrastructure 2 (SRPMS) (6.1-i386)
Red Hat Update Infrastructure 2 (SRPMS) (6.0-i386)
Red Hat Update Infrastructure 2 (SRPMS) (6.0-x86_64)
Red Hat Update Infrastructure 2 (SRPMS) (6Server-x86_64)
Red Hat Update Infrastructure 2 (SRPMS) (6Server-i386)
Red Hat Update Infrastructure 2 (SRPMS) (6.1-x86_64)
```



### NOTE

Using the *All in Certificate* method to add the repositories will add all these repositories at the same time. If you prefer to use the *By Product* method, add the product called **Red Hat Update Infrastructure 2 (RPMs)** and, optionally, **Red Hat Update Infrastructure 2 (SRPMS)**.

2. Check that the correct repositories have been installed using Red Hat Update Infrastructure Manager:

```


rhui (repo) => 1

Red Hat Repositories
 Red Hat Update Infrastructure 2 (RPMs) (6Server-x86_64)
```

```

Red Hat Update Infrastructure 2 (RPMs) (6.0-i386)
Red Hat Update Infrastructure 2 (RPMs) (6.1-i386)
Red Hat Update Infrastructure 2 (RPMs) (6.1-x86_64)
Red Hat Update Infrastructure 2 (RPMs) (6Server-i386)
Red Hat Update Infrastructure 2 (RPMs) (6.0-x86_64)
Red Hat Update Infrastructure 2 (SRPMS) (6.1-i386)
Red Hat Update Infrastructure 2 (SRPMS) (6.0-i386)
Red Hat Update Infrastructure 2 (SRPMS) (6.0-x86_64)
Red Hat Update Infrastructure 2 (SRPMS) (6Server-x86_64)
Red Hat Update Infrastructure 2 (SRPMS) (6Server-i386)
Red Hat Update Infrastructure 2 (SRPMS) (6.1-x86_64)

```

```


```

3. Synchronize the repositories. Use [Procedure 9.4, “Trigger a Repository Synchronization”](#).
4. Wait for the repository synchronization to complete, and then synchronize the CDS instances. Use [Procedure 9.5, “Trigger a CDS Synchronization”](#).
5. Generate an entitlement certificate for the Red Hat Update Infrastructure repositories. Use [Procedure 4.1, “Generating an Entitlement Certificate”](#).
6. Generate a client configuration rpm. Use [Procedure 4.2, “Create a Client Configuration RPM”](#).



#### NOTE

When you name the client configuration RPM, choose a name that will help you to identify that the RPM is to be used for providing updates to Red Hat Update Infrastructure, such as `rhui-update`.

7. Install the client configuration RPM on the Red Hat Update Appliance and each CDS that requires updating:

```
yum install /tmp/rhui/rhui-update.el6.noarch.rpm
```

8. Perform the update using `yum`. The client configuration RPM will configure a yum repository called `rhui-rhui-2`. Use the `yum update` command to update each instance:

```
yum update
```



#### NOTE

Running `yum update` will pull updates from all enabled yum repositories. To pull updates from the `rhui-rhui-2` yum repository only, use the following command instead:

```
yum --disablerepo=* --enablerepo=rhui-rhui-2 update
```

9. Run the `pulp-migrate` command to apply database migrations if applicable:

```
pulp-migrate
```

- 
- 10. Restart the service to pick up any changes.

On the Red Hat Update Appliance, restart the **pulp-server** service:

```
service pulp-server restart
```

On each CDS instance, restart the **pulp-cds** service:

```
service pulp-cds restart
```



#### NOTE

Once the Red Hat Update Infrastructure yum repositories have been configured on the Red Hat Update Appliance and the CDS, only the final step is required to update those systems.

[Report a bug](#)

## 12.1. ADDITIONAL INSTRUCTIONS FOR UPDATING RED HAT UPDATE INFRASTRUCTURE 2.1.1 AND LOWER

This section contains additional instructions for updating from Red Hat Update Infrastructure 2.1.1 and lower.

On the Red Hat Update Appliance, follow the steps from 1 to 9 in [Procedure 12.1, “Updating Red Hat Update Infrastructure”](#) before following these additional steps.

### Procedure 12.2. Updating Red Hat Update Infrastructure from 2.1.1 and lower

1. Edit `/etc/pulp/pulp.conf` using your preferred text editor and set `verify_checksum` and `verify_size` in the `[yum]` section to `true`.

```
[yum]
...
verify_checksum: true
verify_size: true
```

2. Stop the `pulp-server` service.

```
service pulp-server stop
```

3. Run the migration script.

```
pulp-package-migrate --migrate -d /var/lib/pulp
```

**NOTE**

The script could take several hours to run and logs to `/tmp/pulp_package_migrate.log`. Backup the log file to a different directory such as `/root/` after the script completes so that it will be saved.

4. Start the pulp-server service.

```
service pulp-server start
```

5. Synchronize the repositories. Use [Procedure 9.4, “Trigger a Repository Synchronization”](#).
6. After the synchronization is completed, edit `/etc/pulp/pulp.conf` using your preferred text editor and set `verify_checksum` and `verify_size` in the `[yum]` section to `false`.

```
[yum]
...
verify_checksum: false
verify_size: false
```

7. Restart the pulp-server service.

```
service pulp-server restart
```

8. For each CDS, follow the steps from 1 to 9 in [Procedure 12.1, “Updating Red Hat Update Infrastructure”](#).

**IMPORTANT**

The migration script requires `httpd` to be stopped on the CDS, so clients will not be able to connect to the CDS to install or update packages. In deployments with more than 1 CDS server, it is recommended to only run the script on one CDS at a time so that clients will fail over and use one of the other CDS servers that is still up.

9. Edit `/etc/pulp/cds.conf` using your preferred text editor and set `verify_checksum` and `verify_size` in the `[cds]` section to `true`.

```
[cds]
...
verify_checksum = true
verify_size = true
```

10. Stop the pulp-cds service.

```
service pulp-cds stop
```

11. Run the migration script.

```
pulp-package-migrate --migrate -d /var/lib/pulp-cds
```

**NOTE**

The script could take several hours to run and logs to `/tmp/pulp_package_migrate.log`. Backup the log file to a different directory such as `/root/` after the script completes so that it will be saved.

12. Start the goferd service.

```
service goferd start
```

13. Return to the Red Hat Update Appliance server and start a CDS synchronization. Use [Procedure 9.5, “Trigger a CDS Synchronization”](#).

14. After the synchronization is completed, edit `/etc/pulp/cds.conf` using your preferred text editor and set `verify_checksum` and `verify_size` in the `[cds]` section to `false`.

```
[cds]
...
verify_checksum = false
verify_size = false
```

15. Restart the pulp-cds service.

```
service pulp-cds restart
```

[Report a bug](#)



## CHAPTER 13. DELETING ORPHANED PACKAGES

Red Hat Update Infrastructure does not delete orphaned packages. Orphaned packages are package files no longer referenced by any repositories but remain on the file system and consume disk space. Package files can become orphaned packages as a result of the configuration settings or repository deletion. If unsure about the deletion of these packages consider enlarging disk space instead of removing the packages.

This chapter explains how to delete orphaned packages on Red Hat Update Appliance and CDS servers to reclaim disk space. This can either be done by modifying the configuration settings or running the `pulp-purge-packages` script.



### IMPORTANT

These procedures delete packages from Red Hat Update Infrastructure. A complete back up should be performed before using the following procedures. See [Chapter 14, Backup and Restore for Red Hat Update Infrastructure](#) for more information.

#### Procedure 13.1. Deleting Orphaned Packages by modifying configuration settings

- On the Red Hat Update Appliance Server, open `/etc/pulp/pulp.conf` in your preferred text editor. Set `remove_old_versions` to `true` to have Red Hat Update Infrastructure delete references to old packages versions beyond the versions specified in `num_old_pkgs_keep`.

```
True/False to flag if we should remove older rpms
remove_old_versions: true
Integer to specify how many old packages to keep.
num_old_pkgs_keep: 2
```

#### Procedure 13.2. Deleting Orphaned Packages from a CDS

- Open `/etc/pulp/cds/cds.conf` in your preferred text editor.
- Set `cds_remove_old_versions` to `true`.



### NOTE

Setting `remove_old_versions` to `true` on the Red Hat Update Appliance server and `cds_remove_old_versions` to `false` on the CDS will result in errors as the CDS will attempt to synchronize packages that no longer exist on the Red Hat Update Appliance server.

To correct this set both to `true` to remove all old packages beyond the version specified by `num_old_pkgs_keep` or set `remove_old_packages` to `false` on the Red Hat Update Appliance server to retain all old packages and make the required changes for the CDS.

- Set `num_old_pkgs_keep` to the required number of packages to keep.

**NOTE**

Make sure that *num\_old\_pkgs\_keep* on the CDS is set to the same or lower than on the Red Hat Update Appliance server if *remove\_old\_versions* is *true* on the Red Hat Update Appliance server to ensure that the CDS is not trying to synchronize packages that do not exist on the Red Hat Update Appliance server.

4. Save and exit the file.

```
True/False to flag if we should remove older rpms
cds_remove_old_versions: true
Integer to specify how many old packages to keep.
num_old_pkgs_keep: 2
```

This will remove old package versions beyond the versions specified in *num\_old\_pkgs\_keep* on the CDS the next time it is synchronized.

**Procedure 13.3. Deleting Orphaned Packages using the `pulp-purge-packages` script**

Packages can become orphaned when the associated repositories are deleted. To delete orphaned packages:

1. Stop the `httpd` service.

```
service httpd stop
```

2. View packages available for deletion.

```
pulp-purge-packages --dry-run
```

3. Run the `pulp-purge-packages` script to delete orphaned packages and reclaim disk space.

**WARNING**

This script will delete all packages not associated with any repositories. Including packages that were uploaded with the `'pulp-admin content upload'` command.

```
pulp-purge-packages
```

**NOTE**

The `pulp-purge-packages` script logs to `/var/log/pulp-purge-packages`.

## 4. Start the httpd service.

```
service httpd start
```

**NOTE**

The **pulp-purge-packages** script automatically detects the packages and repositories for package deletion, so the script can be run interchangeably on both the Red Hat Update Appliance and CDS servers. View all options and commands by using the **--help** command:

```
Options:
 -h, --help show this help message and exit
 -p PACKAGES_DIR, --packages-dir=PACKAGES_DIR
 packages directory to purge of
 orphaned packages (will
 be detected if not specified)
 -r REPOS_DIR, --repos-dir=REPOS_DIR
 repository directory to search for
 referenced packages
 (will be detected if not specified)
 -d, --dry-run dry run only, do not actually purge any
 packages
```

**Example 13.1. Example Script Output**

```
[root@rhui2 ~]# pulp-purge-packages --dry-run
2013-03-22 15:28:20,405 - Starting run of ./pulp-purge-packages

WARNING: httpd must be stopped before this script can be run.

WARNING: This script will delete all packages not associated with any
repositories. Including packages that were uploaded with the 'pulp-
admin content upload' command.

Continue? Y/N: y

2013-03-22 15:28:21,550 - Detecting packages and repository directories.
2013-03-22 15:28:21,551 - Using /var/lib/pulp/repos as repository
directory.
2013-03-22 15:28:21,551 - Using /var/lib/pulp/packages as packages
directory to purge.
2013-03-22 15:28:21,551 - Dry run specificed, No actual deletes will
occur.
2013-03-22 15:28:21,552 - Getting all referenced packages, this may take
a few minutes...
<logging of found packages to delete>
```

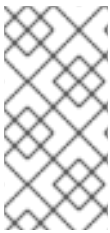
[Report a bug](#)

# CHAPTER 14. BACKUP AND RESTORE FOR RED HAT UPDATE INFRASTRUCTURE

This chapter explains the procedure of backing up and restoring your Red Hat Update Infrastructure.

## 14.1. BACKUP AND RESTORE FOR THE RED HAT UPDATE APPLIANCE

Follow these steps to perform a backup of the Red Hat Update Appliance server. Note that stopping services does not disable any client instances from updating or installing packages since clients are only connected to CDS, and not to the Red Hat Update Appliance server. However, if you have an automated monitoring solution in place as detailed in [Chapter 11, \*Monitoring\*](#), your monitoring may fail during the backup process.



### NOTE

The `/var/lib/pulp` directory is likely to be quite large depending on how many repositories have been deployed on the Red Hat Update Appliance. Refer to Section 2.1 of the Installation guide to estimate how large this directory may be, or use the "du" command from the command line to determine its size.

### 14.1.1. Backup Process

#### Procedure 14.1. Backup Red Hat Update Appliance

1. Stop the pulp-server service:

```
service pulp-server stop
```

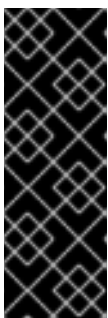
2. Verify each service has stopped:

```
service pulp-server status
mongod is stopped
```

```
service httpd status
httpd is stopped
```

```
service qpidd status
qpidd is stopped
```

3. Backup the following files:



### IMPORTANT

It is important that the files included in the backup retain their current attributes.

Use the following command to backup the files:

```
cp -a source_files_path destination_files_path
```

```

/etc/pulp/*
/etc/rhui/rhui-tools.conf
/etc/pki/pulp/*
/etc/pki/rhua/*
/etc/pki/rhui/*
/var/log/pulp/*
/var/log/httpd/*
/var/lib/mongodb/pulp_database*
/var/lib/pulp/*
/etc/httpd/conf.d/pulp.conf
/etc/httpd/conf.d/ssl.conf
/etc/qpidd.conf
/etc/rhui/myanswersfile

```



#### NOTE

In addition to the above files, you may wish to backup any generated client entitlement certificates and client configuration RPMs.

- Restart the pulp-server service:

```
service pulp-server start
```

[Report a bug](#)

### 14.1.2. Restore Process

Prepare a new Red Hat Update Appliance instance by following steps 1 to 5 from Procedure 2.2 in the *Red Hat Update Infrastructure Installation Guide* Once those steps are completed, proceed with the following restoration steps:

#### Procedure 14.2. Restore Red Hat Update Appliance

- Stop the pulp-server service:

```
service pulp-server stop
```

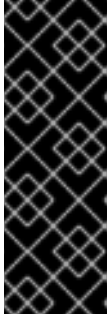
- Verify each service has stopped:

```
service pulp-server status
mongod is stopped
```

```
service httpd status
httpd is stopped
```

```
service qpidd status
qpidd is stopped
```

- Restore all the files that were backed up to their original locations.



### IMPORTANT

It is important that the files included in the restore retain their current attributes.

Use the following command to restore the files:

```
cp -a source_files_path destination_files_path
```

4. Restart the pulp-server service:

```
service pulp-server start
```

[Report a bug](#)

## 14.2. BACKUP AND RESTORE FOR A CONTENT DELIVERY SERVER

Follow these steps on each CDS to perform a backup. Since client instances connect directly to CDS instances to install and update packages, stopping services on a CDS can possibly cause an outage for clients.

To mitigate the outage, if you have more than one CDS installed in each CDS cluster, only backup a single CDS in each cluster at a time, and client instances will automatically fail over to other running CDS instances.



### NOTE

The `/var/lib/pulp-cds` directory will likely be quite large depending on how many repositories have been deployed on the CDS. Refer to Section 2.1 of the Installation guide to estimate how large this directory may be, or use the "du" command from the command line to determine its size.

### 14.2.1. Backup Process

#### Procedure 14.3. Backup Content Delivery Server

1. Stop the pulp-cds service:

```
service pulp-cds stop
```

2. Verify each service has stopped:

```
service httpd status
httpd is stopped
```

```
service goferd status
goferd is not running
```

3. On the CDS, the following files need to be backed up:

**IMPORTANT**

It is important that the files included in the backup retain their current attributes.

Use the following command to backup the files:

```
cp -a source_files_path destination_files_path
```

```
/etc/pulp/*
/etc/pki/cds/*
/var/log/pulp-cds/*
/var/log/gofer/agent.log
/var/log/httpd/*
/var/lib/pulp-cds/*
/etc/httpd/conf.d/ssl.conf
```

**NOTE**

In addition to the above files, you may wish to backup any generated client entitlement certificates and client configuration RPMs.

- Restart the pulp-server service:

```
service pulp-cds start
```

[Report a bug](#)

### 14.2.2. Restore Process

Prepare a new CDS instance by following all steps from Procedure 2.3 in the *Red Hat Update Infrastructure Installation Guide*. Once those steps are completed, proceed with the following restoration steps:

#### Procedure 14.4. Restore Content Delivery Server

- Stop the pulp-cds service:

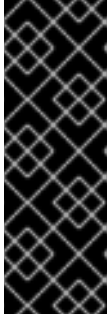
```
service pulp-cds stop
```

- Verify each service has stopped:

```
service httpd status
httpd is stopped
```

```
service goferd status
goferd is not running
```

- Restore all the files that were backed up to their original locations.



## IMPORTANT

It is important that the files included in the restore retain their current attributes.

Use the following command to restore the files:

```
cp -a source_files_path destination_files_path
```

4. Restart the pulp-server service:

```
service pulp-cds start
```

[Report a bug](#)



## APPENDIX A. RED HAT UPDATE INFRASTRUCTURE 2.1 API REFERENCE

Red Hat Update Infrastructure makes use of an open source project called [Pulp](#) for its web service components. A portion of the Pulp API's has been officially documented here as supported by Red Hat Update Infrastructure API's.

### A.1. REPOSITORIES

#### A.1.1. Repo Object Fields

- **id** <str> *repository identifier*
- **source** <RepoSource object> *upstream content source*
- **name** <str> *human-friendly name*
- **arch** <str> *hardware architecture that repository is for*
- **release** <str> *release number*
- **packages** <list of str> *list of package ids in the repository [deferred field]*
- **package\_count** <int> *number of packages in the repository*
- **packagegroups** <object> *map of package group names to list of package ids in the group [deferred field]*
- **packagegroupcategories** <object> *map of categories to lists of package group names [deferred field]*
- **repomd\_xml\_path** <str> *path to the repository's repomd xml file*
- **group\_xml\_path** <str> *path to the repository's group xml file*
- **group\_gz\_xml\_path** <str> *path to the repository's compressed group xml file*
- **sync\_schedule** <iso8601> *formatted recurring interval*
- **last\_sync** <str or nil> *date and time of last successful sync in iso8601 format, nil if has not been synced*
- **feed\_ca** <str> *full path on the Pulp server to the certificate authority used to verify SSL connections to the repo's feed*
- **feed\_cert** <str> *full path on the Pulp server to the certificate used to authenticate Pulp with the repo's feed server when synchronizing content*
- **feed\_key** <str> *full path on the Pulp server to the private key for the feed certificate*
- **consumer\_ca** <str> *full path on the Pulp server to the certificate authority used to verify consumer entitlement certificates*
- **consumer\_cert** <str> *full path on the Pulp server to the entitlement certificate that will be given to bound consumers to authenticate access to the repository*

- **consumer\_key** <str> *full path on the Pulp server to the private key for the consumer's entitlement certificate*
- **errata** <object> *map of errata names to lists of package ids in each errata [deferred field]*
- **groupid** <list of str> *list of repository group ids this repository belongs to*
- **relative\_path** <str> *repository's path relative to the configured root*
- **files** <list of str> *list of ids of the non-package files in the repository [deferred field]*
- **publish** <bool> *whether or not the repository is available*
- **clone\_ids** <list of str> *list of repository ids that are clones of this repository*
- **distributionid** <list of str> *list of distribution ids this repository belongs to [deferred fields]*
- **checksum\_type** <str> *name of the algorithm used for checksums of the repository's content for feedless repos; For feed repos, this gets overwritten by source checksum type from repomd.xml.*
- **filters** <list of str> *list of filter ids associated with the repository*
- **content\_types** <str> *content type allowed in this repository; default:yum; supported: [yum, file]*
- **notes** <dict> *custom key-value attributes for this repository*

### A.1.2. RepoSource Object Fields

- **supported\_types** <list of str> *list of supported types of repositories*
- **type** <str> *repository source type*
- **url** <str> *repository source url*

### A.1.3. Task Object Fields

- **id** <str> *unique id (usually a uuid) for the task*
- **method\_name** <str> *name of the pulp library method that was called*
- **state** <str> *one of several valid states of the tasks lifetime: waiting, running, finished, error, timed\_out, canceled, reset, suspended*
- **start\_time** <str or nil> *time the task started running in iso8601 format, nil if the task has not yet started*
- **finish\_time** <str or nil> *time the task finished running in iso8601 format, nil if the task has not yet finished*
- **result** <object or nil>> *the result of the pulp library method upon return, usually nil*
- **exception** <str or ni> *a string representation of an error in the pulp library call, if any*
- **traceback** <str or nil> *a string print out of the trace back for the exception, if any*
- **progress** <object or nil> *object representing the pulp library call's progress, nil if no information is available*

- **scheduled\_time** <str or nil> *time the task is scheduled to run in iso8601 format, applicable only for scheduled tasks*

#### A.1.4. Progress Object Fields

- **step** <str> *name of the step the pulp library call is on*
- **items\_total** <int> *the total number of items to be processed by the call*
- **items\_left** <int> *the remaining number of items to be processed by the call*
- **details** <object> *object providing further details on the progress*

#### A.1.5. Details Object Fields

- **num\_success** <int> *the number of items successfully processed*
- **total\_count** <int> *the number of items that were attempted to be processed*

#### A.1.6. TaskHistory Object Fields

- **id** <str> *uuid of task*
- **class\_name** <str> *name of the task was a instance method*
- **method\_name** <str> *name of the task callable*
- **args** <list> *name of the task was a instance method*
- **kwargs** <dict> *dictionary of arguments passed to the task callable*
- **state** <str> *final state of the task*
- **progress** <object> *Progress object for the progress at the end of the task*
- **result** <object> *result returned by the task, most likely null*
- **exception** <str> *error, if one occurred*
- **traceback** <str> *traceback if error occurred*
- **consecutive\_failures** <int> *number of failures since last success*
- **scheduled\_time** <str> *iso8601 combined date time when task was scheduler to run*
- **start\_time** <str> *iso8601 combined date time when task actually ran*
- **finished\_time** <str> *iso8601 combined date time when task completed*

## A.2. LIST AVAILABLE REPOSITORIES

To get a list of all repositories managed by Pulp:

- **method:** GET
- **path:** /repositories/

- **permission:** READ
- **success response:** 200 OK
- **failure response:** None
- **return:** list of Repo objects, possibly empty

**example:**

```
[
 {
 'arch': 'noarch',
 'checksum_type': 'sha256',
 'clone_ids': ['0ad-clone', '0ad-clone-again'],
 'comps': '/pulp/api/repositories/0ad/comps/',
 'consumer_ca': None,
 'consumer_cert': None,
 'content_types': 'yum',
 'distribution': '/pulp/api/repositories/0ad/distribution/',
 'distributionid': [],
 'errata': '/pulp/api/repositories/0ad/errata/',
 'feed_ca': None,
 'feed_cert': None,
 'files': '/pulp/api/repositories/0ad/files/',
 'files_count': 0,
 'filters': [],
 'groupid': [],
 'id': '0ad',
 'keys': '/pulp/api/repositories/0ad/keys/',
 'last_sync': '2012-01-04T13:55:11-07:00',
 'name': '0ad',
 'notes': {},
 'package_count': 2,
 'packagegroupcategories':
'/pulp/api/repositories/0ad/packagegroupcategories/',
 'packagegroups': '/pulp/api/repositories/0ad/packagegroups/',
 'packages': '/pulp/api/repositories/0ad/packages/',
 'preserve_metadata': False,
 'publish': True,
 'relative_path': 'repos/bioinformatics/0ad/fedora-16/x86_64',
 'source': {'type': 'remote',
'url':
'http://repos.fedorapeople.org/repos/bioinformatics/0ad/fedora-
16/x86_64/'},
 'sync_options': {'skip': {}},
 'sync_schedule': '2011-12-13T13:45:00-07:00/PT5M',
 'uri':
'https://localhost/pulp/repos/repos/bioinformatics/0ad/fedora-
16/x86_64/',
 'uri_ref': '/pulp/api/repositories/0ad/'},
 ...
]
```

- **filters:**
  - **id** <str> repository id

- **name** <str> *repository name*
- **arch** <str> *repository content architecture*
- **groupid** <str> *repository group id*
- **relative\_path** <str> *repository's on disk path*
- **note** <str> *repository note in the format key:value*

### A.3. SYNCHRONIZATION

To synchronize the repository's content from its source:

- **method:** POST
- **path:** /repositories/<id>/sync/
- **permission:** EXECUTE
- **success response:** 202 Accepted
- **failure response:**
  - 404 Not Found if the id does not match a repository
  - 406 Not Acceptable if the repository does not have a source
  - 409 Conflict if a sync is already in progress for the repository
- **return:** a Task object
- **parameters:**
  - **timeout** (*optional*) <str> *repository id*
  - **skip** (*optional*) <str> *repository name*
  - **limit** (*optional*) <str> *repository content architecture*
  - **threads** (*optional*) <str> *repository group id*

### A.4. ADD FILTERS

To add filters to the repository:

- **method:** POST
- **path:** /repositories/<id>/add\_filters/
- **permission:** EXECUTE
- **success response:** 200 OK
- **failure response:** 404 Not Found if the id does not match a repository
- **return:** true

- **parameters:**
  - filters <list of str> *list of filter ids*

## A.5. ADD GROUP

To add a group to the repository:

- **method:** POST
- **path:** /repositories/<id>/add\_group/
- **permission:** EXECUTE
- **success response:** 200 OK
- **failure response:** 404 Not Found if the id does not match a repository
- **return:** true
- **parameters:**
  - addgrp <list of str> *group id*

## A.6. CLONE

To create a new repository by cloning an existing one:

- **method:** POST
- **path:** /repositories/<id>/clone/
- **permission:** EXECUTE
- **success response:** 200 OK
- **failure response:**
  - 404 Not Found if the id does not match a repository
  - 409 Conflict if the parameters match an existing repository
  - 409 Conflict if the parent repository is currently syncing
- **return:** a Task object
- **parameters:**
  - clone\_id <str> *the id of the clone repository*
  - clone\_name <str> *the name of clone repository*
  - feed <str> *parent/origin/none*
  - relative\_path (optional) <str> *clone repository on disk path*
  - groupid (optional) <str> *repository groups that clone belongs to*

- **filters** (optional) <list of objects> *synchronization filters to apply to the clone*
- **publish** (optional) <bool> *sets the publish state on a repository; if not specified uses 'default\_to\_published' value from pulp.conf*

## A.7. PUBLISH

To update a repository's 'publish' state:

True means the repository is exposed through Apache.  
False stops the repository from being exposed through Apache.

- **method:** POST
- **path:** /repositories/<id>/update\_publish/
- **permission:** EXECUTE
- **success response:** 200 OK
- **failure response:** 404 Not Found if the id does not match a repository
- **return:** True on successful update, False otherwise
- **parameters:**
  - **state** <bool> *publish state*

## A.8. REPOSITORY DISTRIBUTION

To list the distributions the repository is part of:

- **method:** GET
- **path:** /repositories/<id>/distribution/
- **permission:** READ
- **success response:** 200 OK
- **failure response:** Unspecified
- **return:** list of Distribution objects

## A.9. ADD NOTE

To add a Note to the Repository:

- **method:** POST
- **path:** /repositories/<id>/notes/
- **permission:** EXECUTE
- **success response:** 200 OK

- **failure response:**
  - 404 Not found if given repository does not exist
  - 409 Conflict if given key already exists
- **return:** True
- **parameters:**
  - **key** <str> *key to be added*
  - **value** <str> *value of key*

## A.10. DELETE NOTE

To delete a Note from a Repository:

- **method:** DELETE
- **path:** /repositories/<id>/notes/<key>
- **permission:** DELETE
- **success response:** 200 OK
- **failure response:**
  - 404 Not found if given repository does not exist
  - 404 Not Found if given key does not exist
- **return:**
- **parameters:**
  - <>

## A.11. UPDATE NOTE

To change the value of an existing key in Repository Notes:

- **method:** PUT
- **path:** /repositories/<id>/add\_group/
- **permission:**
- **success response:**
- **failure response:**
  - 404 Not found if given repository does not exist
  - 404 Not Found if given key does not exist
- **return:** true



- **parameters:** new value of the key

## A.12. SYNCHRONIZATION SCHEDULES

To list all repository synchronization schedules:

- **method:** GET
- **path:** /repositories/schedules/
- **permission:** READ
- **success response:** 200 OK
- **failure response:** None
- **return:** list of object that are mappings of repository id to synchronization schedule

## A.13. DELETE SCHEDULE

To remove a repository's schedule for the given type:

- **method:** DELETE
- **path:** /repositories/<id>/schedules/<type>/
- **permission:** DELETE
- **success response:** 200 OK
- **failure response:** 404 Not Found
- **return:** (empty) Schedule object

**example:**

```
{'href': '/pulp/api/repositories/0ad/',
 'id': '0ad',
 'options': null,
 'schedule': null}
```

## A.14. GET SCHEDULE

To get the repository schedule for the given type:

- **method:** GET
- **path:** /repositories/<id>/schedules/<type>/
- **permission:** READ
- **success response:** 200 OK
- **failure response:** 404 Not Found
- **return:** Schedule object

**example:**

```
{'href': '/pulp/api/repositories/0ad/',
 'id': '0ad',
 'options': {'skip': {}},
 'schedule': '2011-12-13T13:45:00-07:00/PT5M',
 'type': 'sync'}
```

## A.15. CREATE OR REPLACE SCHEDULE

To create or replace schedule:

- **method:** PUT
- **path:** /repositories/<id>/schedules/<type>/
- **permission:** CREATE
- **success response:** 200 OK
- **failure response:** Unspecified
- **return:** Schedule object

**example:**

```
{'href': '/pulp/api/repositories/0ad/',
 'id': '0ad',
 'options': {'skip': {}},
 'schedule': '2011-12-13T13:45:00-07:00/PT5M',
 'type': 'sync'}
```

- **parameters:**
  - **schedule** <str> *schedule for given type in iso8601 format*
  - **options** <obj> *options for the scheduled action*

## A.16. GET DISTRIBUTIONS

To get a distribution object:

- **method:** GET
- **path:** /repositories/<id>/
- **permission:** READ
- **success response:** 200 OK
- **failure response:** 404 Not Found if the id does not match a distribution
- **return:** a Distribution object

**example:**

■

```
{ "id": "ks-Fedora-None-16-x86_64",
 "_id": "ks-Fedora-None-16-x86_64",
 "_ns": "distribution",
 "description": "ks-Fedora-Fedora-16-x86_64",
 "family": "Fedora",
 "relativepath": "/var/lib/pulp/distributions/ks-Fedora-Fedora-16-
x86_64",
 "variant": Fedora,
 "version": "16",
 "timestamp": "2011-11-03T07:39:38",
 "arch": "x86_64",
 "url": ["http://localhost/pulp/ks/released/F-
16/GOLD/Everything/x86_64/os/"]}
 "files": ["/var/lib/pulp/distributions/ks-Fedora-Fedora-16-
x86_64/initrd.img",
 "/var/lib/pulp/distributions/ks-Fedora-Fedora-16-
x86_64/repomd.xml",
 "/var/lib/pulp/distributions/ks-Fedora-Fedora-16-x86_64/vmlinuz",
 "/var/lib/pulp/distributions/ks-Fedora-Fedora-16-
x86_64/.treeinfo",
 "/var/lib/pulp/distributions/ks-Fedora-Fedora-16-
x86_64/efiboot.img",
 "/var/lib/pulp/distributions/ks-Fedora-Fedora-16-
x86_64/boot.iso",
 "/var/lib/pulp/distributions/ks-Fedora-Fedora-16-
x86_64/efidisk.img"]}]}
```

## A.17. LIST ALL

To get a list of all distributions managed by Pulp:

- **method:** GET
- **path:** /distributions/
- **permission:** READ
- **success response:** 200 OK
- **failure response:** None
- **return:** list of distribution objects, possibly empty

**example:**

```
[{'_id': 'ks-Fedora-Fedora-14-x86_64',
 '_ns': 'distribution',
 'description': None,
 'id': 'ks-Fedora-Fedora-14-x86_64',
 'family' : 'Fedora',
 'variant' : 'Fedora',
 'version' : '14',
 'arch' : 'x86_64',
 'timestamp' : '2011-05-13T15:44:30',
 'files': ['/var/lib/pulp/distributions/ks-Fedora-Fedora-14-
x86_64/images/install.img',
```

```

 '/var/lib/pulp/distributions/ks-Fedora-Fedora-14-
x86_64/images/efiboot.img',
 '/var/lib/pulp/distributions/ks-Fedora-Fedora-14-
x86_64/images/boot.iso',
 '/var/lib/pulp/distributions/ks-Fedora-Fedora-14-
x86_64/images/efidisk.img',
 '/var/lib/pulp/distributions/ks-Fedora-Fedora-14-
x86_64/images/pxeboot/initrd.img',
 '/var/lib/pulp/distributions/ks-Fedora-Fedora-14-
x86_64/images/pxeboot/vmlinuz'],
 'relativepath' : '/var/lib/pulp/distributions/ks-Fedora-Fedora-14-
x86_64'},]

```

## A.18. CONTENT DELIVERY SERVER (CDS)

### A.18.1. Register

Registers a Content Delivery Server (CDS) instance with the Pulp server. The CDS instance must be configured and running prior to this call. This call will attempt to initialize the CDS instance which involves making an invocation against it and negotiating the shared token. If that initialization is unsuccessful, this call will fail and the CDS will not be added to the Pulp server.

There is no explicit step for creating a CDS cluster. If the CDS is to be part of a cluster, the desired cluster ID is specified in this call and effectively creates the cluster at that point.

- **method:** POST
- **path:** /cds/
- **permission:** CREATE
- **success response:** 201 - Created
- **failure response:**
  - 409 - if there is already a CDS instance with the given hostname
  - 500 - if the CDS instance cannot be reached to be initialized or raises an error during initialization
- **return:**dictionary describing the created CDS instance
- **parameters:**dictionary describing the created CDS instance
  - **hostname** <str> *the hostname of the CDS instance being registered*
  - **name** <str> *user-friendly name for the CDS instance*
  - **description** <str> *user-friendly description of the CDS instance*
  - **sync\_schedule** <str> *schedule in iso8601 format*
  - **cluster\_id** <str> *(optional) ID of the cluster the CDS should belong to*
- **sample response:**

```
{
 "description": null,
 "_ns": "cds",
 "hostname": "pulp-cds-1",
 "last_sync": null,
 "secret": "833925b0-2732-46dd-95bc-570d532c1e31",
 "cluster_id": null,
 "sync_schedule": null,
 "repo_ids": [],
 "_id": "pulp-cds-1",
 "id": "pulp-cds-1",
 "name": "pulp-cds-1"
}
```

### A.18.2. Unregister

Removes a CDS instance from Pulp control. The CDS processes running on the instance are not stopped. The shared secret between the Pulp server and CDS is removed from the CDS, allowing it to be registered to a different Pulp server.

This call will attempt to inform the CDS it is being released. If that call to the CDS fails, this call will fail. The CDS can be forcibly removed from the Pulp server's database by passing the "force" flag to the call.

- **method:** DELETE
- **path:** /cds/<hostname>/
- **permission:** DELETE
- **success response:** 200 - OK
- **failure response:**
  - 500 - if there is no CDS for the given hostname
- **return:** true if the CDS was unregistered
- **parameters:** dictionary describing the created CDS instance
  - **hostname** <str> *the hostname of the CDS instance being registered*
  - **force** <boolean> *if true, the CDS will be removed from the Pulp database even if it cannot be contacted*

### A.18.3. List All

Returns a list of all CDS instances registered to the Pulp server.

- **method:** GET
- **path:** /cds/
- **permission:** READ
- **success response:** 200 - OK

- **failure response:** None
- **return:** list of dictionaries describing each CDS instance
- **sample response:**

```
[
 {
 "description": null,
 "_ns": "cds",
 "hostname": "pulp-cds-1",
 "last_sync": null,
 "secret": "56f3a490-cdff-435f-ae42-4bddc0526f3e",
 "cluster_id": null,
 "sync_schedule": null,
 "repo_ids": [],
 "heartbeat": [
 false,
 "2011-11-30T21:16:00.767145+00:00",
 {
 "cds": {}
 }
],
 "_id": "pulp-cds-1",
 "id": "pulp-cds-1",
 "name": "pulp-cds-1"
 }
]
```

#### A.18.4. Get CDS Instance

Retrieves metadata describing a single CDS instance.

- **method:** GET
- **path:** /cds/<hostname>/
- **permission:** READ
- **success response:** 200 - OK
- **failure response:**
  - 404 - if no CDS exists with the given hostname
- **return:** dictionary describing the CDS instance
- **parameters:**
  - **hostname** <str> *the hostname of the CDS instance to retrieve*
- **sample response:**

```
{
 "description": null,
 "_ns": "cds",
 "hostname": "pulp-cds-1",
```

```

 "last_sync": null,
 "secret": "56f3a490-cdff-435f-ae42-4bddc0526f3e",
 "cluster_id": null,
 "sync_schedule": null,
 "repo_ids": [],
 "heartbeat": [
 false,
 "2011-11-30T21:16:00.767145+00:00",
 {
 "cds": {}
 }
],
 "_id": "pulp-cds-1",
 "id": "pulp-cds-1",
 "name": "pulp-cds-1"
 }

```

### A.18.5. Associate a Repository

Configures the CDS to host the given repository. The repository is not immediately sent to the CDS but rather will be downloaded in the next CDS sync.

- **method:** POST
- **path:** /cds/<hostname>/associate/
- **permission:** EXECUTE
- **success response:** 201 - Accepted
- **failure response:**
  - 409 - if the Pulp server is currently redistributing consumers for the given repository across the current CDS host list for that repository
- **return:** task instance describing the redistribution of consumers to take the CDS into account
- **parameters:**
  - **hostname** <str> *task instance describing the redistribution of consumers to take the CDS into account*
  - **repo\_id** <str> *identifies the repository being associated with the CDS instance*
- **sample response:**

```

{
 "scheduled_time": "2011-11-30T21:18:49Z",
 "exception": null,
 "args": [
 "repo-1"
],
 "job_id": null,
 "class_name": "CdsApi",
 "start_time": null,
 "traceback": null,
 "method_name": "redistribute",

```

```

 "finish_time": null,
 "state": "waiting",
 "result": null,
 "scheduler": "immediate",
 "progress": null,
 "id": "e5496305-1b98-11e1-9c0f-00508d977dff"
 }

```

### A.18.6. Unassociate a Repository

Removes a repository from being hosted by a CDS instance. As with `associate`, this call simply updates the Pulp database. The change is not seen on the CDS instance until the next sync occurs.

- **method:** POST
- **path:** `/cds/<hostname>/unassociate/`
- **permission:** EXECUTE
- **success response:** 201 - Accepted
- **failure response:**
  - 409 - if the Pulp server is currently redistributing consumers for the given repository across the current CDS host list for that repository
- **return:** task instance describing the redistribution of consumers to take the CDS removal into account
- **parameters:**
  - **hostname** `<str>` *the hostname of the CDS instance to update*
  - **repo\_id** `<str>` *identifies the repository being removed from the CDS instance*
- **sample response:**

```

{
 "scheduled_time": "2011-11-30T21:19:16Z",
 "exception": null,
 "args": [
 "repo-1"
],
 "job_id": null,
 "class_name": "CdsApi",
 "start_time": null,
 "traceback": null,
 "method_name": "redistribute",
 "finish_time": null,
 "state": "waiting",
 "result": null,
 "scheduler": "immediate",
 "progress": null,
 "id": "f5968980-1b98-11e1-86e4-00508d977dff"
}

```



## A.18.7. Retrieve CDS History

Retrieves history entries describing events that took place on a CDS instance.

- **method:** POST
- **path:** /cds/<hostname>/history/
- **permission:** EXECUTE
- **success response:** 200 - OK
- **failure response:** None
- **return:** list of CDS history items (empty list if none match) containing the following fields:
  - **cds\_hostname** <str> *hostname of the CDS the entry refers to*
  - **originator** <str> *indicates the user who triggered the action that caused the history event*
  - **type\_name** <str> *describes the type of event that occurred; possible values: registered, unregistered, sync\_started, sync\_finished, repo\_associated, repo\_unassociated*
  - **details** <str> *arbitrary text further describing the event (will differ based on event type)*
  - **timestamp** <str> *timestamp in iso8601 format of when the event item was added*
- **parameters:** (any of the parameters can be combined for advanced queries)
  - **hostname** <str> *the hostname of the CDS instance to update*
  - **event\_type** <str> *(optional) if specified, only events of the given type are returned (see above for valid type names)*
  - **limit** <str> *(optional) if specified, only up to the given amount of items will be returned*
  - **sort** <str> *(optional) indicates the sort direction by timestamp; valid values: ascending, descending*
  - **start\_date** <str> *(optional) if specified, only events that took place after the given date (in iso8601 format) will be returned*
  - **end\_date** <str> *(optional) if specified, only events that took place before the given date (in iso8601 format) will be returned*
- **sample response:**

```
[
 {
 "originator": "admin",
 "cds_hostname": "pulp-cds-1",
 "timestamp": "2011-11-30T21:19:16Z",
 "type_name": "repo_unassociated",
 "details": {
 "repo_id": "repo-1"
 },
 "_ns": "cds_history",
 "_id": "4d813279-1fac-4f04-b9c6-28a00545e0e3",
```

```

 "id": "4d813279-1fac-4f04-b9c6-28a00545e0e3"
 },
 {
 "originator": "admin",
 "cds_hostname": "pulp-cds-1",
 "timestamp": "2011-11-30T21:18:49Z",
 "type_name": "repo_associated",
 "details": {
 "repo_id": "repo-1"
 },
 "_ns": "cds_history",
 "_id": "93d7d6dd-baf7-4ce5-b580-c134736ceeca",
 "id": "93d7d6dd-baf7-4ce5-b580-c134736ceeca"
 }
]

```

### A.18.8. Trigger a CDS Synchronization

Schedules a CDS sync to run as soon as possible (may be delayed depending on what other work the Pulp server is doing). The sync will take place asynchronously and the call will return immediately with information on how to track the sync.

- **method:** POST
- **path:** /cds/<hostname>/sync/
- **permission:** EXECUTE
- **success response:** 201 - Accepted
- **failure response:**
  - 400 - if the timeout cannot correctly be parsed
  - 404 - if no CDS exists with the given hostname
- **return:** task instance describing the sync task
- **parameters:**
  - **hostname** <str> *the hostname of the CDS instance to update*
  - **timeout** <str> *timeout value in iso8601 format that dictates how long the sync may run before the Pulp server aborts the request*
- **sample response:**

```

{
 "scheduled_time": "2011-11-30T21:21:04Z",
 "exception": null,
 "args": [
 "pulp-cds-1"
],
 "job_id": null,
 "class_name": "CdsApi",
 "start_time": null,
 "traceback": null,

```

```

 "method_name": "cds_sync",
 "finish_time": null,
 "state": "waiting",
 "result": null,
 "scheduler": "immediate",
 "progress": null,
 "id": "36203045-1b99-11e1-add9-00508d977dff"
 }

```

## A.19. FILTERS

Repository filters are ways of filtering available source content from repositories.

### A.19.1. List

Returns a list of filters in Pulp Server.

- **method:** GET
- **path:** /pulp/apifilters/
- **permission:** READ
- **success response:** 200 - OK
- **failure response:** None
- **return:** <List> Filter objects
- **sample response:**

```

[
 {
 "description": "test-filter",
 "_ns": "filters",
 "package_list": [
 "foo"
],
 "_id": "test-filter",
 "type": "blacklist",
 "id": "test-filter"
 },
 {
 "description": "test-filter1",
 "_ns": "filters",
 "package_list": [
 "foo1"
],
 "_id": "test-filter1",
 "type": "blacklist",
 "id": "test-filter1"
 }
]

```

### A.19.2. Create

Creates a blacklist/whitelist filter.

- **method:** POST
- **path:** /pulp/apifilters/
- **permission:** CREATE
- **success response:** 201 - Created
- **failure response:**
  - 409 - Conflict when filter with the same id already exists
- **return:** <dict> Newly created filter object
- **parameters:** dictionary describing the created CDS instance
  - **id** <str> Filter ID required
  - **type** <str> **blacklist OR whitelist** required
  - **description** <str> Filter description optional - defaulted to filter id
  - **package\_list** <str> List of names or full NEVRA or python re syntax compatible regexes for packages (optional)
- **sample response:**

```
{
 "description": "test-filter",
 "_ns": "filters",
 "package_list": [
 "foo"
],
 "_id": "test-filter",
 "type": "blacklist",
 "id": "test-filter"
}
```

### A.19.3. Info

Returns information about a filter.

- **method:** GET
- **path:** /pulp/api/filters/<filter-id>
- **permission:** READ
- **success response:** 200 - OK
- **failure response:**
  - 404 - Not Found if a filter with given id does not exist
- **return:** <dict> filter object

- **sample response:**

```
{
 "description": "test-filter",
 "_ns": "filters",
 "package_list": [
 "foo"
],
 "_id": "test-filter",
 "type": "blacklist",
 "id": "test-filter"
}
```

#### A.19.4. Delete

Deletes a filter and it's associations with repositories, if any.

- **method:** DELETE
- **path:** /pulp/api/filters/<filter-id>
- **permission:** DELETE
- **success response:** 200 - OK
- **failure response:**
  - 404 - Not Found if a filter with given id does not exist
- **return:** true if filter is successfully deleted

## A.20. TASKS

### A.20.1. Task object fields

- **id** <str> *unique id (usually a uuid) for the task*
- **job\_id** <str> *associate job id*
- **class\_name** <str> *name of the class, if the task's method is an instance method*
- **method\_name** <str> *name of the pulp library method that was called*
- **state** <str> *one of several valid states of the tasks lifetime: waiting, running, finished, error, timed\_out, canceled, reset, suspended*
- **start\_time** <str or nil> *time the task started running in iso8601 format, nil if the task has not yet started*
- **finish\_time** <or nil> *time the task finished running in iso8601 format, nil if the task has not yet finished*
- **result** <object or nil> *the result of the pulp library method upon return, usually nil*
- **exception** <str or nil> *a string representation of an error in the pulp library call, if any*

- **traceback** <str or nil> *a string print out of the trace back for the exception, if any*
- **progress** <object or nil> *object representing the pulp library call's progress, nil if no information is available*
- **scheduled\_time** <str or nil> *time the task is scheduled to run in iso8601 format, applicable only for scheduled tasks*
- **snapshot\_id** <str> *id of task's snapshot, if it has one*

### A.20.2. TaskSnapshot object fields

- **id** <str> *unique task id*
- **class\_name** <str> *name of the class, if the task's method is an instance method*
- **method\_name** <str> *name of the pulp library method that was called*
- **state** <str> *one of several valid states of the tasks lifetime: waiting, running, finished, error, timed\_out, canceled, reset, suspended*
- **failure\_threshold** <int> *number of failures allowed this task before it is no longer scheduled*
- **cancel\_attempts** <int> *the number of times cancel was called on this task*
- **callable** <str> *pickled task method*
- **args** <str> *pickled arguments for the task method*
- **kwargs** <str> *pickled keyword arguments for the task method*
- **progress** <object or nil> *object representing the pulp library call's progress, nil if no information is available*
- **timeout** <str> *pickled timedelta representing the time limit for the task's run*
- **schedule\_threshold** <str> *pickled timedelta representing a max difference between the scheduled\_time and start\_time before an error is logged*
- **\_progress\_callback** <str> *pickled method allowing progress information to be recorded by the task*
- **start\_time** <str> *pickled datetime showing the start time of the task*
- **finish\_time** <str> *pickled datetime showing the finish time of the task*
- **result** <str> *pickled result of the task call*
- **exception** <str> *pickled error, if one occurred*
- **traceback** <str> *pickled traceback, if one occurred*

### A.20.3. Get a Task

To get a Task object for a specific task:

- **method:** GET

- **path:** /tasks/<id>/
- **permission:** READ
- **success response:** 200 OK
- **failure response:**
  - 404 Not Found if no such task
- **return:** Task object
- **sample response:**

```
{
 "scheduled_time": "2012-01-05T06:00:00Z",
 "exception": null,
 "args": [],
 "job_id": null,
 "class_name": null,
 "start_time": null,
 "traceback": null,
 "method_name": "cull_history",
 "finish_time": null,
 "state": "waiting",
 "result": null,
 "scheduler": "interval",
 "snapshot_id": null,
 "progress": null,
 "id": "7e419e4f-3719-11e1-9ea0-f0def14122c8"
}
```

#### A.20.4. Get all Task

To get a list of all tasks currently in the tasking system:

- **method:** GET
- **path:** /tasks/
- **permission:** READ
- **success response:** 200 OK
- **failure response:** None
  - 404 Not Found if no such task
- **return:** list of task objects
- **filters:**
  - **id** <str> *task id*
  - **state** <str> *tasking system task state: waiting, running, complete, incomplete, current, archived (current is the same as waiting, running, complete, and incomplete, also the same as omitting the state filter; archived looks into the task history, will not return archived tasks without this*

*filter* )

- **sample response:**

```
[
 {
 "scheduled_time": "2012-01-05T01:00:00Z",
 "exception": null,
 "args": [],
 "job_id": null,
 "class_name": null,
 "start_time": null,
 "traceback": null,
 "method_name": "cull_audited_events",
 "finish_time": null,
 "state": "waiting",
 "result": null,
 "scheduler": "interval",
 "snapshot_id": null,
 "progress": null,
 "id": "6cd50f8f-371a-11e1-ba28-f0def14122c8"
 },
 {
 "scheduled_time": "2012-01-05T06:00:00Z",
 "exception": null,
 "args": [],
 "job_id": null,
 "class_name": null,
 "start_time": null,
 "traceback": null,
 "method_name": "cull_history",
 "finish_time": null,
 "state": "waiting",
 "result": null,
 "scheduler": "interval",
 "snapshot_id": null,
 "progress": null,
 "id": "6cd53278-371a-11e1-ba29-f0def14122c8"
 }
]
```



## APPENDIX B. REVISION HISTORY

|                                                                                                                                                                                                                     |                          |                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|-----------------------|
| <b>Revision 0-40</b>                                                                                                                                                                                                | <b>Tue Dec 16 2014</b>   | <b>Megan Lewis</b>    |
| <p>Added bug links throughout book.<br/>         BZ#1169200 Added warnings to Orphaned Packages chapter.<br/>         BZ#1145972 Corrected backup and restore procedures.</p>                                       |                          |                       |
| <b>Revision 0-39</b>                                                                                                                                                                                                | <b>Wed Dec 10 2014</b>   | <b>Megan Lewis</b>    |
| <p>Standardized the Preface to match new standards.<br/>         Standardized the Abstract to match new standards.<br/>         Standardized titles to match new standards.<br/>         Applied brand changes.</p> |                          |                       |
| <b>Revision 0-38</b>                                                                                                                                                                                                | <b>Mon Dec 16 2013</b>   | <b>Dan Macpherson</b> |
| <p>Adding clarification to update instructions</p>                                                                                                                                                                  |                          |                       |
| <b>Revision 0-37</b>                                                                                                                                                                                                | <b>Mon Dec 16 2013</b>   | <b>Dan Macpherson</b> |
| <p>Changing</p>                                                                                                                                                                                                     |                          |                       |
| <b>Revision 0-36</b>                                                                                                                                                                                                | <b>Thurs Nov 21 2013</b> | <b>Megan Lewis</b>    |
| <p>#1017410 Implemented feedback</p>                                                                                                                                                                                |                          |                       |
| <b>Revision 0-35</b>                                                                                                                                                                                                | <b>Fri Nov 1 2013</b>    | <b>Megan Lewis</b>    |
| <p>#1017409 Added details to limit old packages synchronizing with a CDS</p>                                                                                                                                        |                          |                       |
| <b>Revision 0-34</b>                                                                                                                                                                                                | <b>Thur Oct 31 2013</b>  | <b>Megan Lewis</b>    |
| <p>#1007410 Added details to view CDS logs and to create a custom log</p>                                                                                                                                           |                          |                       |
| <b>Revision 0-33</b>                                                                                                                                                                                                | <b>Tue Oct 15 2013</b>   | <b>Megan Lewis</b>    |
| <p>#1007657 Edited details of certificates generated by nss-db-gen</p>                                                                                                                                              |                          |                       |
| <b>Revision 0-32</b>                                                                                                                                                                                                | <b>Mon Apr 22 2013</b>   | <b>Julie Wu</b>       |
| <p>Fixed typos.</p>                                                                                                                                                                                                 |                          |                       |
| <b>Revision 0-31</b>                                                                                                                                                                                                | <b>Mon Apr 22 2013</b>   | <b>Julie Wu</b>       |
| <p>#953156 Added a note in Procedure 3.3</p>                                                                                                                                                                        |                          |                       |
| <b>Revision 0-30</b>                                                                                                                                                                                                | <b>Wed Apr 17 2013</b>   | <b>Julie Wu</b>       |
| <p>#950086 Edited Procedure 12.2</p>                                                                                                                                                                                |                          |                       |
| <b>Revision 0-29</b>                                                                                                                                                                                                | <b>Thu Mar 28 2013</b>   | <b>Julie Wu</b>       |
| <p>#924783 Edited note<br/>         #921250 Added new section: Updating Red Hat Update Infrastructure to 2.1.2</p>                                                                                                  |                          |                       |
| <b>Revision 0-28</b>                                                                                                                                                                                                | <b>Tue Mar 26 2013</b>   | <b>Julie Wu</b>       |
| <p>#924783 Added additional step</p>                                                                                                                                                                                |                          |                       |
| <b>Revision 0-27</b>                                                                                                                                                                                                | <b>Tue Mar 26 2013</b>   | <b>Julie Wu</b>       |
| <p>#924783 Added new chapter</p>                                                                                                                                                                                    |                          |                       |
| <b>Revision 0-26</b>                                                                                                                                                                                                | <b>Tue Mar 19 2013</b>   | <b>Julie Wu</b>       |
| <p>#908405 Removed extra line</p>                                                                                                                                                                                   |                          |                       |
| <b>Revision 0-25</b>                                                                                                                                                                                                | <b>Mon Feb 25 2013</b>   | <b>Julie Wu</b>       |
| <p>#855904 Edited Qpid-SSL chapter</p>                                                                                                                                                                              |                          |                       |
| <b>Revision 0-24</b>                                                                                                                                                                                                | <b>Fri Feb 22 2013</b>   | <b>Dan Macpherson</b> |

#855904 Added additional instructions for Qpid-SSL chapter

|                                                                                                                                                                                       |                        |                     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|---------------------|
| <b>Revision 0-23</b><br>#868828 Added procedure 2.8<br>#855904 Added in Qpid-SSL chapter                                                                                              | <b>Tue Feb 19 2013</b> | <b>Julie Wu</b>     |
| <b>Revision 0-20</b><br>#908135 fixed all prompt to #<br>#908405 Added step in procedure 11.1                                                                                         | <b>Thu Feb 07 2013</b> | <b>Julie Wu</b>     |
| <b>Revision 0-20</b><br>#893174 changed repo 2.0 to 2<br>#890079 added default username and password<br>#908135 Procedure 11.1 fixed example prompt                                   | <b>Wed Feb 06 2013</b> | <b>Julie Wu</b>     |
| <b>Revision 0-19</b><br>Fixed index terms: RHUI/RHUA abbreviation                                                                                                                     | <b>Tue Aug 21 2012</b> | <b>Julie Wu</b>     |
| <b>Revision 0-18</b><br>QE edits. BZ 844203                                                                                                                                           | <b>Tue Aug 21 2012</b> | <b>Shikha Nansi</b> |
| <b>Revision 0-17</b><br>Fixed Typo in Procedure 3.7. BZ 844205                                                                                                                        | <b>Fri Aug 17 2012</b> | <b>Shikha Nansi</b> |
| <b>Revision 0-16</b><br>Brewing                                                                                                                                                       | <b>Thu Aug 16 2012</b> | <b>Julie Wu</b>     |
| <b>Revision 0-15</b><br>Adjusted screen shots to fit "connected:rhua.example.com" in the same line.                                                                                   | <b>Thu Aug 16 2012</b> | <b>Julie Wu</b>     |
| <b>Revision 0-14</b><br>Brewing                                                                                                                                                       | <b>Tue Aug 14 2012</b> | <b>Shikha Nansi</b> |
| <b>Revision 0-13</b><br>Added note in RHUI Manager BZ#844203, BZ#844205                                                                                                               | <b>Tue Aug 14 2012</b> | <b>Shikha Nansi</b> |
| <b>Revision 0-12</b><br>Removed Table 5.1 #844203<br>Removed numbering in Revision history                                                                                            | <b>Tue Aug 14 2012</b> | <b>Julie Wu</b>     |
| <b>Revision 0-11</b><br>Updated Chapter 6 as per James' feedback<br>Edited 1.3 Status codes as per James' feedback<br>Added step 6 in Procedure11.1; moved Note from step 5 to step 6 | <b>Mon Aug 13 2012</b> | <b>Julie Wu</b>     |
| <b>Revision 0-10</b><br>small typo in repositories.xml                                                                                                                                | <b>Tue Aug 07 2012</b> | <b>Shikha Nansi</b> |
| <b>Revision 0-9</b>                                                                                                                                                                   | <b>Tue Aug 07 2012</b> | <b>Shikha Nansi</b> |

---

|                                                                                                                                                             |                        |                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|---------------------|
| Updates as per BZ# 844480                                                                                                                                   |                        |                     |
| Updates as per BZ# 844483                                                                                                                                   |                        |                     |
| Updates as per BZ# 844487                                                                                                                                   |                        |                     |
| Updates as per BZ# 843633                                                                                                                                   |                        |                     |
| Updates as per BZ# 844204                                                                                                                                   |                        |                     |
| Updates as per BZ# 844205                                                                                                                                   |                        |                     |
| Updates as per BZ# 844213                                                                                                                                   |                        |                     |
| Updates as per BZ# 844214                                                                                                                                   |                        |                     |
| <b>Revision 0-8</b>                                                                                                                                         | <b>Mon Aug 06 2012</b> | <b>Julie Wu</b>     |
| Fixed screen shots Example 1.1 and 1.2. Added i manage identity certificate                                                                                 |                        |                     |
| <b>Revision 0-7</b>                                                                                                                                         | <b>Tue Jul 31 2012</b> | <b>Julie Wu</b>     |
| Chapter 7-12 QE edits                                                                                                                                       |                        |                     |
| <b>Revision 0-6</b>                                                                                                                                         | <b>Fri Jul 27 2012</b> | <b>Shikha Nansi</b> |
| Changed all heading in Appendix to Title Case.                                                                                                              |                        |                     |
| Appendix QE edits                                                                                                                                           |                        |                     |
| Reworded some sentences in Appendix.                                                                                                                        |                        |                     |
| "Renamed Content_Distribution_Servers.xml to Content_Delivery_Servers                                                                                       |                        |                     |
| Deleted Duplicate "Delete Note" from Appendix                                                                                                               |                        |                     |
| Changed the manual identifier to Documentation-Administration_Guide                                                                                         |                        |                     |
| Chapter 1 QE edits                                                                                                                                          |                        |                     |
| Chapter 2 QE edits                                                                                                                                          |                        |                     |
| Chapter 3 QE edits                                                                                                                                          |                        |                     |
| Chapter 6 QE edits                                                                                                                                          |                        |                     |
| <b>Revision 0-5</b>                                                                                                                                         | <b>Fri Jul 20 2012</b> | <b>Shikha Nansi</b> |
| Added RHUI-2.1 APIs appendix                                                                                                                                |                        |                     |
| Repositioned CRL chapter                                                                                                                                    |                        |                     |
| <b>Revision 0-4</b>                                                                                                                                         | <b>Fri Jul 20 2012</b> | <b>Julie Wu</b>     |
| Edited screen shot in Example 1.1.                                                                                                                          |                        |                     |
| Fixed link in Procedure 6.1                                                                                                                                 |                        |                     |
| Edited CRL chapter                                                                                                                                          |                        |                     |
| Remove Chapter 8 Red Hat Entitlement Certificates                                                                                                           |                        |                     |
| Fixed links in Chap 6 and 11                                                                                                                                |                        |                     |
| <b>Revision 0-3</b>                                                                                                                                         | <b>Thu Jul 19 2012</b> | <b>Julie Wu</b>     |
| Edited author group email address.                                                                                                                          |                        |                     |
| Fixed typo in Procedure6.1                                                                                                                                  |                        |                     |
| Procedure2.6 step1, changed "type p at the prompt to list the packages in a particular repository" to "type c at the prompt to create a custom repository". |                        |                     |
| Changed screen shot in Procedure2.6 step5.                                                                                                                  |                        |                     |
| Switched Procedure 3.3 and 3.4.                                                                                                                             |                        |                     |
| Procedure4.2 step2. screen shot edited.                                                                                                                     |                        |                     |
| Fixed screen shots: Connected:teletran-1 to Connected:rhua.example.com"                                                                                     |                        |                     |
| <b>Revision 0-2</b>                                                                                                                                         | <b>Thu Jul 12 2012</b> | <b>Shikha Nansi</b> |
| Bug 810691 - RFE: add a chapter or section to show how to update the entitlement-signing CA certificate.                                                    |                        |                     |
| <b>Revision 0-1</b>                                                                                                                                         | <b>Mon Jul 2 2012</b>  | <b>Shikha Nansi</b> |
| Initial creation of book by publican                                                                                                                        |                        |                     |

---

# INDEX

## B

backup, [Backup and Restore for Red Hat Update Infrastructure](#)

## C

CDS (see [Red Hat Update Infrastructure Manager](#), [Content Delivery Server](#))

Certification Revocation List

CRL, [Implementing a Certification Revocation List](#)

Revoke, [Implementing a Certification Revocation List](#)

Content Delivery Server (CDS) Backup

Backup process for Red Hat Update Appliance, [Backup Process](#)

Content Delivery Server (CDS) Restore

Backup process for CDS, [Restore Process](#)

## L

log files, [Monitoring](#)

## M

monitoring, [Monitoring](#)

log files, [Monitoring](#)

## O

Orphaned Package

Unreferenced Package, [Deleting Orphaned Packages](#)

## Q

Qpid SSL

nss-db-gen, [Using Qpid SSL Certificates](#)

## R

Red Hat Update Appliance Backup

Backup process for Red Hat Update Appliance, [Backup Process](#)

Red Hat Update Appliance Restore

Restore process for Red Hat Update Appliance, [Restore Process](#)

Red Hat Update Infrastructure

monitoring, [Monitoring](#)

updating, [Updating Red Hat Update Infrastructure](#)

---

**Red Hat Update Infrastructure Manager, [Using the Red Hat Update Infrastructure Manager](#)**

add a CDS instance, [Using Content Delivery Server Instances](#)

add a Red Hat repository, [Using Repositories](#)

Associate a repository with a CDS cluster, [Using Content Delivery Server Instances](#)

CDS cluster, [Using Content Delivery Server Instances](#)

change user password, [Configuring Users](#)

content delivery server, [Using Content Delivery Server Instances](#)

create client configuration RPM, [Using Client Entitlements](#)

create configuration RPMs, [Using Client Entitlements](#)

create custom repository, [Using Custom Repositories](#)

custom repositories, [Using Custom Repositories](#)

delete a CDS instance, [Using Content Delivery Server Instances](#)

delete a Red Hat repository, [Using Repositories](#)

delete packages from a custom repository, [Using Custom Repositories](#)

display repository information, [Using Repositories](#)

entitlement certificates, [Using Client Entitlements](#), [Managing Entitlement Certificates](#)

exiting, [Using the Red Hat Update Infrastructure Manager](#)

first launch, [Red Hat Update Infrastructure Manager First Launch](#)

generate entitlement certificate, [Using Client Entitlements](#)

identity certificates, [Using Identity Certificates](#)

launch, [Red Hat Update Infrastructure Manager First Launch](#)

list CDS instances, [Using Content Delivery Server Instances](#)

list CDS synchronization status, [Status and Scheduling](#)

list certificate information, [Managing Entitlement Certificates](#)

list custom repository entitlements, [Managing Entitlement Certificates](#)

list packages in a repository, [Using Repositories](#)

list repositories, [Using Repositories](#)

list repository synchronization status, [Status and Scheduling](#)

logging out, [Using the Red Hat Update Infrastructure Manager](#)

move a CDS instance, [Using Content Delivery Server Instances](#)

navigation, [Using the Red Hat Update Infrastructure Manager](#)

operation, [Using the Red Hat Update Infrastructure Manager](#)

repositories, [Using Repositories](#)

screens, [Using the Red Hat Update Infrastructure Manager](#)

shell prompt, [Using the Red Hat Update Infrastructure Manager from the Shell Prompt](#)

shell prompt cert command, [Using the Red Hat Update Infrastructure Manager from the Shell Prompt](#)

shell prompt flags, [Using the Red Hat Update Infrastructure Manager from the Shell Prompt](#)

shell prompt packages command, [Using the Red Hat Update Infrastructure Manager from the Shell Prompt](#)

shell prompt repo command, [Using the Red Hat Update Infrastructure Manager from the Shell Prompt](#)

shell prompt status command, [Using the Red Hat Update Infrastructure Manager from the Shell Prompt](#)

starting, [Using the Red Hat Update Infrastructure Manager](#)

status and scheduling, [Status and Scheduling](#)

trigger CDS synchronization, [Status and Scheduling](#)

trigger repository synchronization, [Status and Scheduling](#)

Unassociate a repository from a CDS cluster, [Using Content Delivery Server Instances](#)

upload content certificate, [Managing Entitlement Certificates](#)

upload packages to a custom repository, [Using Custom Repositories](#)

users, [Configuring Users](#)

view last synchronization details, [Status and Scheduling](#)

## U

### Update Certificates

Update Process, [Updating the Entitlement-Signing CA Certificate](#)