



Red Hat Satellite 5.8

User Guide

Using and administering Red Hat Satellite

Red Hat Satellite 5.8 User Guide

Using and administering Red Hat Satellite

Red Hat Satellite Documentation Team
satellite-doc-list@redhat.com

Legal Notice

Copyright © 2017 Red Hat.

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

This guide covers the use and administration of Red Hat Satellite. For further information, see the Red Hat Satellite Getting Started Guide and the Red Hat Satellite Client Configuration Guide.

Table of Contents

CHAPTER 1. MANAGING USER ACCOUNTS	4
1.1. CREATING AND DELETING USER ACCOUNTS	4
1.2. ASSIGNING ROLES TO USER ACCOUNTS	6
1.3. CUSTOMIZING SELECTED PARTS OF RED HAT SATELLITE	7
CHAPTER 2. AUTOMATICALLY SYNCHRONIZING THE RED HAT SATELLITE SERVER REPOSITORY ...	8
CHAPTER 3. PLANNING FOR DISASTER RECOVERY	9
3.1. BACKING UP A RED HAT SATELLITE SERVER	9
3.2. BACKING UP AN EMBEDDED DATABASE	10
3.3. CLONING A RED HAT SATELLITE WITH AN EMBEDDED DATABASE	13
3.4. CREATING REDUNDANT SATELLITES WITH EXTERNAL DATABASES	14
3.5. AUTOMATING SATELLITE DATABASE BACKUPS	15
CHAPTER 4. USING COMMAND LINE CONFIGURATION MANAGEMENT TOOLS	17
4.1. USING RED HAT NETWORK ACTIONS CONTROL	17
4.2. USING THE RED HAT NETWORK CONFIGURATION CLIENT	18
4.3. USING THE RED HAT NETWORK CONFIGURATION MANAGER	21
4.4. USING THE RED HAT SATELLITE COMMAND LINE TOOL (SPACECMD)	28
4.5. USING THE RED HAT SATELLITE FINAL ARCHIVE TOOL (SPACEWALK-FINAL-ARCHIVE)	29
CHAPTER 5. CLONING SOFTWARE CHANNELS AND ERRATA	31
5.1. FEATURES	31
5.2. EXAMPLE USAGE	31
CHAPTER 6. MAINTAINING SYSTEM SECURITY USING OPENSAP	33
6.1. OPENSAP FEATURES	33
6.2. OPENSAP PREREQUISITES	33
6.3. RED HAT SATELLITE PREREQUISITES FOR USING OPENSAP	34
6.4. PERFORMING AUDIT SCANS	34
6.5. OPENSAP SATELLITE WEB INTERFACE	37
CHAPTER 7. REPORTING CLIENT SOFTWARE FAILURES	41
7.1. VIEWING SOFTWARE FAILURES FOR A SINGLE CLIENT	41
7.2. GROUPING SIMILAR SOFTWARE FAILURES	41
7.3. CHANGING ORGANIZATION-WIDE SETTINGS FOR SOFTWARE FAILURE REPORTS	41
7.4. LOG FILES OF SOFTWARE FAILURES	42
CHAPTER 8. GENERATING RED HAT SATELLITE REPORTS	43
CHAPTER 9. SCHEDULING RED HAT SATELLITE ADMINISTRATIVE TASKS	45
9.1. SCHEDULING A RUN	46
9.2. SETTING UP A SELF-SUBSCRIBED RED HAT SATELLITE	47
CHAPTER 10. TROUBLESHOOTING	50
10.1. Disk Space	50
10.2. Installing and Updating	50
10.3. Services	51
10.4. Connectivity	51
10.5. Logging and Reporting	53
10.6. Errors	58
10.7. Web Interface	63
10.8. Anaconda	63
10.9. Tracebacks	65

10.10. Registration	66
10.11. Kickstarts and Snippets	66
10.12. Multi-Organization Satellites and Satellite Certificate	67
10.13. Proxy Installation and Configuration	68
APPENDIX A. REVISION HISTORY	73

CHAPTER 1. MANAGING USER ACCOUNTS

1.1. CREATING AND DELETING USER ACCOUNTS

The **Users** page on the Red Hat Satellite web server provides suitable tools to manage Satellite users. You can use this page to create, delete, activate, and deactivate user accounts, as well as assign roles and their associated permissions.

Creating User Accounts

Before Satellite users can register with the Satellite server to request product updates or to perform other maintenance, they need a suitable user account. Only certain Satellite **Administrators** can create user accounts.

Procedure 1.1. Creating User Accounts

To create a user account:

1. Navigate to the Satellite web server page, and click the **Users** tab on the navigation bar.
2. On the right side of the page, click **create new user** to open the **Create User** page.
3. Complete all of the required fields.



NOTE

The login value must be at least five characters long, and may only contain alphanumeric, hyphen, underscore, comma, period, and commercial at (@) characters.

4. Click **Create Login** to create the new user. An email will be sent to the user, using the address specified during creation, to inform them of the new account details. This will include the password in plain text.
5. When the account has been successfully created, you will be redirected to the **User List** page. To change permissions and set options for the new user, select their name from the displayed list to display the **User Details** page, and navigate to the appropriate tabs to make your changes.

Deleting User Accounts

Only Satellite **Administrators** can delete user accounts. Deleted accounts cannot be used to log in to the Satellite server interface, or to schedule actions.



WARNING

You cannot retrieve deleted user accounts. Consider deactivating the user account before deleting it, in order to assess the consequences.

Procedure 1.2. Deleting User Accounts

To delete a user account:

1. Navigate to the Satellite web server page, and click the **Users** tab on the navigation bar.
2. Click the user name of the account that you want to delete from the **Username** list. The **User Details** page displays.
3. Ensure that the user account is not a Satellite administrator.

If the user is a Satellite administrator, clear the associated check box, and click **Submit**.

If the user is not a Satellite administrator, continue to the next step.

4. Click **Delete User**. The **Confirm User Deletion** page displays.
5. Ensure that you want to completely delete this user account, and click **Delete User**.

After the user account has been successfully deleted, you will be returned to the **Active Users** page. The user's name will no longer appear in the **Active Users** list.

Procedure 1.3. Activating and Deactivating Users

User accounts are automatically activated when they are created. They can be deactivated by administrators, or users can deactivate their own accounts. Deactivated user accounts cannot log in to the Satellite server interface, or schedule any actions. Any actions that were scheduled before the account was deactivated remain in the action queue until they are completed. Deactivated user accounts can only be reactivated by administrators.



NOTE

Administrator accounts can only be deactivated after the Administrator role has been removed from the account.

To deactivate a user account:

1. Select the user's name from the list in the **Users** tab, to display the **User Details** page.
2. Check to see if the user is a Satellite administrator.

If the user is a Satellite administrator, uncheck the box next to that role, and click **Submit**.

If the user is not a Satellite administrator, continue to the next step.

3. Click **Deactivate User**.

You will be asked to confirm this action, by clicking it again. Check the details, and then click **Deactivate User** again to confirm.

4. Once the account has been successfully deactivated, the user's name will not appear in the **Active Users** list. Click the **Deactivated** link from the **User List** menu to view deactivated user accounts.

5. To reactivate the user account, view the **Deactivated** list, check the box next to the user to be reactivate, and click **Reactivate**.

1.2. ASSIGNING ROLES TO USER ACCOUNTS

User accounts can be managed through the **Users** tab at the top of the Satellite Server navigation bar. To change permissions and set options for a user, select their name from the displayed list to display the **User Details** page, and navigate to the appropriate tabs to make your changes. Modify account details by making the changes and clicking **Submit**.

User Roles

User roles are used to delegate responsibilities to user accounts. Each user role has a different level of responsibility and access.

To assign a user a new role, select the appropriate checkbox on the **User Details** page. Modify roles by making the changes and clicking **Submit**.

The user roles to choose from are

Satellite Administrator

A special role for Satellite administrative tasks such as creating organizations, managing subscriptions, and configuring global Satellite Server settings.

This role cannot be assigned on the **User Details** page. A user that already has the Satellite Server administrator role can assign the role to another user by going to **Admin** → **Users**.

Organization Administrator

Performs management functions such as managing users, systems, and channels within the context of their organization. Organization administrators are automatically granted administration access to all other roles, which are signified by the checkboxes for the other roles being selected and grayed-out.

Activation Key Administrator

Performs activation key functions for such as creating, modifying, and deleting keys within the account.

Channel Administrator

Provides complete access to the software channels and related associations within the organization. Performs functions such as making channels globally subscribable, and creating new channels, and managing the packages within channels.

Configuration Administrator

Has complete access to the configuration channels and related associations within the organization. Also has complete access to the kickstart profiles and associated items within the organization. Performs kickstart profile, channel and file management configuration functions in the organization.

System Group Administrator

This role has complete authority over the systems and system groups to which it is granted access. Performs administrative functions such as creating new system groups, deleting assigned system groups, adding systems to groups, and managing user access to groups.

Satellite administrators can remove Satellite administrator rights from user accounts, including their own, but there must always be at least one Satellite administrator.

1.3. CUSTOMIZING SELECTED PARTS OF RED HAT SATELLITE

Selected parts of the Red Hat Satellite web interface can be customized. These include the headers, footers and login page.

1. Open the **rhncnf** file of Red Hat Satellite in a text editor.
2. Edit the file with the required content. To enter content that spans multiple lines escape every new line with a backslash character. Backslashes themselves can be escaped but HTML is not escaped.



NOTE

Red Hat Satellite does not currently support UTF-8 encoding for **rhncnf**.

- To customize the header edit **java.custom_header** with the required content.
 - To customize the footer edit **java.custom_footer** with the required content.
 - To customize the login banner edit **java.login_banner** with the required content.
3. Restart Satellite for the changes to take effect.

CHAPTER 2. AUTOMATICALLY SYNCHRONIZING THE RED HAT SATELLITE SERVER REPOSITORY

Manually synchronizing the Red Hat Satellite server repository with the Red Hat Content Delivery Network (CDN) can be an arduous task. The synchronization can be automated to occur randomly in a designated off peak window for best performance. You can use the **cron** utility to effectively automate synchronization.

Procedure 2.1. To Use the cron Utility to Automate Synchronization:

1. Switch to the root user, and run the following command to open the **crontab** in a text editor:

```
# crontab -e
```

2. Create a suitable job definition to schedule the synchronization. To create a random synchronization time, use the following entry:

```
0 1 * * * perl -le 'sleep rand 9000' && cdn-sync --email >/dev/null  
2>1
```

This entry runs the synchronization job randomly between 01:00 and 03:30, and discards **stdout** and **stderr** messages from the **cron** utility. This prevents duplicating messages from the **cdn-sync** command. Other options can be included as needed. See the **crontab** manual page **man crontab** for more information.

3. Exit the text editor to save the updated **crontab** file. The new rules take effect immediately.



NOTE

The **crontab** file opens in **vi** by default. To change this behavior, change the **EDITOR** variable to the name of the text editor you prefer.

CHAPTER 3. PLANNING FOR DISASTER RECOVERY

This chapter describes the recommended methods for backing up, verifying, and restoring Red Hat Satellite and embedded databases. If you are using an external database, consult your organization's database administrator. If you are using an embedded database, see [Section 3.2, “Backing up an Embedded Database”](#) for a complete description of this process and the options available.

You should create backups either nightly or weekly, depending on the amount of data being stored, and how much data can potentially be lost in the case of a system outage.

If you plan on performing offline, or "cold", backups, Red Hat recommends that you schedule these backups to occur during scheduled Satellite Server maintenance outages, because all website and client connection services will be unavailable during the backup. Satellite 5.6 and later include online, or "hot", backup functionality; it is not necessary to perform offline backups.

3.1. BACKING UP A RED HAT SATELLITE SERVER

Several methods exist for backing up your Red Hat Satellite system. The following methods are those recommended by Red Hat.

Minimal Backup

Red Hat recommends that you back up at least the following files and directories:

- `/var/opt/rh/rh-postgresql95/lib/pgsql/` (Embedded database only)
- `/etc/sysconfig/rhn/`
- `/etc/rhn/`
- `/etc/sudoers`
- `/var/www/html/pub/`
- `/var/satellite/redhat/[0-9]*/` (This is the location of any custom RPMs)
- `/root/.gnupg/`
- `/root/ssl-build/`
- `/etc/dhcp.conf`
- `/etc/httpd`
- `/var/lib/tftpboot/` (In Red Hat Enterprise Linux 6)
- `/var/lib/cobbler/`
- `/var/lib/rhn/kickstarts/`
- `/var/www/`
- `/var/lib/nocpulse/`
- `/etc/tomcat*/`

- `/etc/jabberd/`
- `/etc/cobbler/`

If possible, back up `/var/satellite/` as well. In case of failure, this will save lengthy download times. The `/var/satellite/` directory (specifically `/var/satellite/redhat/NULL/`) is primarily a duplicate of Red Hat's RPM repository, and can be regenerated using the `cdn-sync` command. Red Hat recommends that the entire `/var/satellite/` tree be backed up. In the case of disconnected satellites, `/var/satellite/` *must* be backed up.

Backing up only these files and directories does have some drawbacks. As part of the failure recovery process, you need to:

- Reinstall the Red Hat Satellite ISO RPMs.
- Reregister the server.
- Use the `cdn-sync` command to resynchronize Red Hat packages.
- Reinstall the `/root/ssl-build/rhn-org-httpd-ssl-key-pair-MACHINE_NAME-VER-REL.noarch.rpm` file.

Backup without Reregistration

Another method is to back up all of the files and directories mentioned above but reinstall the Satellite server without reregistering it. During the installation, cancel or skip the Red Hat Network registration and SSL certificate generation sections.

Full Machine Backup

The final and most comprehensive method is to back up the entire machine. This saves download and reinstallation time but requires additional disk space and back-up time.



IMPORTANT

Regardless of the back-up method used, when you restore the Satellite server from a backup, you must run the following command to schedule the recreation of search indexes the next time the `rhn-search` service is started:

```
# service rhn-search cleanindex
```

3.2. BACKING UP AN EMBEDDED DATABASE

Red Hat Satellite provides a specialized command-line utility to assist with embedded database management tasks. The `db-control` command provides features to create, verify, and restore backups, to obtain database status information and to restart the database when necessary. See the `db-control` manual page (`man db-control`) for a full listing of the features available.

The following sections demonstrate how to create, verify, and restore Red Hat Satellite embedded and managed databases.

3.2.1. Performing Online Database Backups

Red Hat Satellite Server 5 contains functionality that enables online backups of your database, without the need to stop the Satellite Server. Additions to the existing **db-control** command make this functionality possible.

Three new options have been added to the **db-control** command:

- **online-backup *FILENAME***: Performs an online backup of the Satellite database (embedded PostgreSQL only).
- **reset-password**: Resets the user password and unlocks the account.
- **restore *DIRECTORY* | *FILENAME***: Restores the database from either:
 - An offline backup taken by **db-control backup** and saved in the *DIRECTORY* directory. The database must be stopped for both the **backup** and **restore** operations in order to run successfully.
 - An online backup taken by **db-control online-backup** and saved as *FILENAME*. The database itself must be running for both the **online-backup** and **restore** operations in order to run successfully, but all other Satellite services must be stopped.

3.2.1.1. Performing an Online Backup

To create an online backup of an embedded Red Hat Satellite 5 server database, change to the root user, and run the following command. Replace the ***FILENAME*** option with the full path to the backup file that you want to create. This location needs to be writable by the PostgreSQL user:

```
# db-control online-backup FILENAME
```



NOTE

There is no need to stop either the database or the Satellite services to perform an online backup.

3.2.1.2. Restoring a Database from an Online Backup

Use the **db-control restore *FILENAME*** command to restore an embedded database from a backup created using the **db-control online-backup** command. Before you restore a database, you need to shut down all Satellite services except the database itself.

Procedure 3.1. To Restore a Database from an Online Backup:

1. Change to the root user, and run the following command to stop all Satellite services except the database:

```
# rhn-satellite stop --exclude=rh-postgresql95-postgresql
```

2. Run the following command to restore the database. Replace the ***FILENAME*** option with the full path to the backup file created with the **db-control online-backup** command:

```
# db-control restore FILENAME
```

3. After the restoration is complete, run the following command to restart the database and all related services:

```
# rhn-satellite start
```

3.2.2. Performing Offline Database Backups

Red Hat Satellite Server 5 provides the ability to perform online backup and restore operations. Red Hat recommends that you continue to perform offline backups during monthly or quarterly maintenance windows.

3.2.2.1. Performing an Offline Backup

The following procedure describes how to back up an embedded Red Hat Satellite server database.

Procedure 3.2. To Create an Offline Backup:

1. Change to the root user, and run the following command to stop the Satellite server:

```
# rhn-satellite stop
```

2. Run the following command to create the backup:

```
# db-control backup DIRECTORY
```

Replace *DIRECTORY* with the absolute path to the location where you want to store your database backup. This process will take several minutes.

3. When the backup is complete, run the following command to restart the Satellite server:

```
# rhn-satellite start
```

4. Copy the backup to another system using **rsync** or another file-transfer utility. Red Hat strongly recommends scheduling the backup process automatically using cron jobs. For instance, back up the system at 03:00 and then copy the backup to the separate repository (partition, disk, or system) at 06:00.

3.2.2.2. Verifying the Backup

Backing up the embedded database is useful only if you can ensure the integrity of the resulting backup. There are two approaches to this integrity check; you can *examine* the backup to check its time stamp and identify any missing files, or you can *verify* the backup, which involves a more thorough inspection and validation of the md5sum of each file in the backup. The first approach is quicker, but the second provides more thorough validation.

To examine the backup, run the following command as root:

```
# db-control examine BACKUP_FILE
```

To verify the backup, run the following command as root:

```
# db-control verify BACKUP_FILE
```

If the verification is successful, you can safely restore the database from the `BACKUP_FILE` directory.



NOTE

Users of external databases should also perform periodic backups. Consult your external database administrator for more information about supported backup procedures.

3.2.2.3. Restoring the Database

Use the `db-control restore` command to restore embedded databases from backup. Before you attempt to restore a database, you need to shut down the database and any related services.

Procedure 3.3. To Restore an Embedded Database from a Backup:

1. Run the following command to stop all of the Red Hat Satellite services:

```
# rhn-satellite stop
```

2. Run the following command, including the directory containing the backup, to begin the restoration. Ensure that you replace *directory* with the absolute path to the location that contains the backup. This process will verify the contents of the backup before restoring the database. The process will take several minutes.

```
# db-control restore directory
```

This not only restores the embedded database but first verifies the contents of the backup directory using checksums.

3. After the restoration is complete, restart the database and related services:

```
# rhn-satellite start
```

4. Regardless of whether you are backing up an external or embedded database, when the database is restored from a backup, you should schedule the restoration of search indexes the next time the `rhn-search` service is started:

```
# service rhn-search cleanindex
```

3.3. CLONING A RED HAT SATELLITE WITH AN EMBEDDED DATABASE

You can limit outages caused by hardware or other failures by cloning the Red Hat Satellite server with an embedded database in its entirety. The cloned server can be prepared for use if the primary server fails.

Procedure 3.4. To Clone a Satellite Server with an Embedded Database:

1. Install Red Hat Satellite with an embedded database on a base install of Red Hat Enterprise Linux on a separate machine. That is, a machine separate from your primary Red Hat Satellite server. Omit the SSL Certificate generation step.

2. Back up the primary server's database daily using the commands described in [Section 3.2.2.1, "Performing an Offline Backup"](#). This ensures that only changes made the day of the failure are lost.
3. Establish a mechanism to copy the backup to the secondary server. Keep these repositories synchronized using a file transfer program such as **rsync**. Copying is not necessary if using a *Storage Area Network (SAN)*.
4. Use the **db-control restore** command to import duplicate data.
5. If the primary server fails, transfer the SSL key pair RPM package in **/root/ssl-build** from the primary to the secondary server, and install that package. This ensures that Red Hat Satellite clients can authenticate with and securely connect to the secondary server.
6. Update your DNS to reference the secondary server, or configure your load balancer appropriately.

3.4. CREATING REDUNDANT SATELLITES WITH EXTERNAL DATABASES

In keeping with the cloning option available to Red Hat Satellite with an embedded database, you can limit outages on Satellite servers with external databases by preparing redundant Satellite servers. Unlike clones, you can run redundant Satellite servers with external databases in either *active* or *standby* mode. This is entirely up to your network topology and is independent of the steps listed here.



IMPORTANT

Before you begin the following procedure, prepare the external database for failover using suitable recommendations for building a fault-tolerant database. Consult your database administrator.

Procedure 3.5. To Create a Redundant Satellite with an External Database:

1. Install Red Hat Satellite on a separate machine, but omit the database configuration, database schema, SSL certificate, and bootstrap script generation steps. Include the same Red Hat Network account and database connection information provided during the initial Satellite installation.
2. Register the new Satellite server. See the Red Hat Satellite *Installation Guide* for more information.
3. If your original SSL certificate does not take your high-availability solution into account, create a new one with a more appropriate **Common Name** value (see *The SSL Maintenance Tool* in the Red Hat Satellite *Client Configuration Guide*). In this case, generate a new bootstrap script (as defined in *Generating Bootstrap Scripts* in the *Red Hat Satellite Client Configuration Guide*) that captures this new value. Ensure the **Common Name** value represents the combined Satellite solution, not a single machine's host name.
4. After installation, copy the following files from the primary server to the secondary:
 - o **/etc/rhn/rhn.conf**
 - o **/etc/tnsnames.ora** (Oracle database only.)

- Copy the server-side SSL certificate RPMs from the primary server and install them on the secondary server.

If, during the installation process, you generated a new SSL certificate that included a new Common Name value, copy the SSL certificate RPMs from the secondary to the primary server and redistribute the client-side certificate. If you also created another bootstrap script, use it to install the certificate on all client systems.

- If you created a new bootstrap script, copy the contents of `/var/www/html/pub/bootstrap/` to the primary server.
 - If you did not create a new bootstrap script, copy the contents of `/var/www/html/pub/bootstrap/` from the primary server to the secondary server.
- Run the following command on the secondary server to stop the **Red Hat Network Task Engine** service:

```
# service taskomatic stop
```

You can use custom scripting or other means to establish automatic start-up/failover of the **Red Hat Network Task Engine** on the secondary server. Regardless, you need to ensure that it starts in the event of a failure.

- Share channel package data (by default located in `/var/satellite`) and cache data (by default located in `/var/cache/rhn`) between the primary and secondary servers over some type of networked storage device. This eliminates data replication and ensures a consistent store of data for each server.
- Make the various servers available on your network using a suitable Common Name and a method that suits your infrastructure. Options include round-robin DNS, a network load balancer, and a reverse-proxy setup.

3.5. AUTOMATING SATELLITE DATABASE BACKUPS

Backup tasks can be automated so that they occur in non-peak times, such as the late evening or early morning. This also ensures they are performed regularly, and are not forgotten. The most effective way to automate backups is using **cron**.

Procedure 3.6. To Automate Satellite Server Database Backups:

Create a new file called **backup-db.sh** containing the following script. This script will stop the satellite, perform a database backup, and restart the satellite:

```
#!/bin/bash
{
/usr/sbin/rhn-satellite stop
d=db-backup-$(date "+%F");
mkdir -p /tmp/$d;
db-control backup /tmp/$d
/usr/sbin/rhn-satellite start
} &> /dev/null
```

- Create a new file called **move-files.sh** containing the following script. This script will use **rsync** to move the backup files to a directory to be stored:

```
#!/bin/bash
rsync -avz /tmp/db-backup-$(date "+%F") <destination> &> /dev/null
```

Replace *<destination>* with the path to the backup directory.

Alternatively, use the following script to achieve the same goal:

```
#!/bin/bash
scp -r /tmp/db-backup-$(date "+%F") <destination> &> /dev/null
```

2. Switch to the root user, and open the **crontab** file in a text editor:

```
# crontab -e
```



NOTE

The **crontab** file opens in vi by default. To change this behavior, change the **EDITOR** variable to the name of the text editor you prefer.

3. Create a suitable job definition to schedule the backup scripts to run:

```
0 3 * * * backup-db.sh
0 6 * * * move-files.sh
```

This **crontab** entry will run the backup at 03:00, and transfer the backup files at 06:00. Other options can be included as needed. You can also include a clean up script to remove older backup directories and prevent the backup storage from filling up.

4. Exit the editor to save the **crontab** file. The new rules take effect immediately.

CHAPTER 4. USING COMMAND LINE CONFIGURATION MANAGEMENT TOOLS

In addition to the options provided in the Red Hat Satellite website, there are two command line tools for managing a system's configuration files: the **Red Hat Network Configuration Client** and the **Red Hat Network Configuration Manager**. There is a complementary **Red Hat Network Actions Control** tool that is used to enable and disable configuration management on client systems. If you do not yet have these these tools installed, they can be found within the **Red Hat Network Tools** child channel for your operating system.



NOTE

Whenever a configuration file is deployed via the website, a backup of the previous file including its full path is made in the `/var/lib/rhncfg/backups/` directory on the affected system. The backup retains its filename but has a `.rhn-cfg-backup` extension appended.

4.1. USING RED HAT NETWORK ACTIONS CONTROL

The **Red Hat Network Actions Control** (`rhncfg-actions-control`) application is used to enable and disable configuration management of a system. Client systems cannot be managed in this fashion by default. This tool allows System Administrators to enable or disable specific modes of allowable actions such as: *deploying* a configuration file onto the system, *uploading* a file from the system, using *diff* to find out what is currently managed on a system and what is available, or allowing running arbitrary *remote commands*. These various modes are enabled/disabled by placing/removing files and directories in the `/etc/sysconfig/rhn/allowed-actions/` directory. Due to the default permissions on the `/etc/sysconfig/rhn/` directory, Red Hat Network Actions Control have to be run by someone with root access.

4.1.1. Using General Command Line Options

There is a `man` page available, as there are for most command line tools. Simply decide what Red Hat Network scheduled actions should be enabled for use by system administrators. These options enable the various scheduled action modes:

Table 4.1. `rhncfg-actions-control` options

Option	Description
<code>--enable-deploy</code>	Allow <code>rhncfg-client</code> to deploy files.
<code>--enable-diff</code>	Allow <code>rhncfg-client</code> to diff files.
<code>--enable-upload</code>	Allow <code>rhncfg-client</code> to upload files.
<code>--enable-mtime-upload</code>	Allow <code>rhncfg-client</code> to upload mtime.
<code>--enable-all</code>	Allow <code>rhncfg-client</code> to do everything.
<code>--enable-run</code>	Enable <code>script.run</code>

Option	Description
--disable-deploy	Disable deployment.
--disable-diff	Disable diff
--disable-upload	Disable upload
--disable-mtime-upload	Disable mtime upload
--disable-all	Disable all options
--disable-run	Disable script.run
--report	Report whether the modes are enabled or disabled
-f, --force	Force the operation without asking first
-h, --help	show help message and exit

Once a mode is set, your system is now ready for config management through Red Hat Satellite. **rhncfg-client --enable-all** is a common option.

4.2. USING THE RED HAT NETWORK CONFIGURATION CLIENT

As the name implies, the **Red Hat Network Configuration Client (rhncfg-client)** is installed and run from an individual client system. From there you may use it to gain knowledge about how Red Hat Network deploys configuration files to the client.

The **Red Hat Network Configuration Client** offers these primary modes: list, get, channels, diff, and verify.

4.2.1. Listing Configuration Files

To list the configuration files for the machine and the labels of the config channels containing them, issue the command:

```
rhncfg-client list
```

The output resembles the following list:

```
Config Channel      File
config-channel-17  /etc/example-config.txt
config-channel-17  /var/spool/aalib.rpm
config-channel-14  /etc/rhn/rhn.conf
```

These are the configuration files that apply to your system. However, there may be duplicate files present in the other channels. For example, issue the following command:

```
rhncfg-manager list config-channel-14
```

and observe the following output:

```
Files in config channel 'config-channel-14' /etc/example-config.txt
/etc/rhn/rhn.conf
```

You may then wonder where the second version of `/etc/example-config.txt` went. The rank of the `/etc/example-config.txt` file in `config-channel-17` was higher than that of the same file in `config-channel-14`. As a result, the version of the configuration file in `config-channel-14` is not deployed for this system, although the file still resides in the channel. The `rhncfg-client` command does not list the file because it will not be deployed on this system.

4.2.2. Getting a Configuration File

To download the most relevant configuration file for the machine, issue the command:

```
rhncfg-client get /etc/example-config.txt
```

You should see output resembling:

```
Deploying /etc/example-config.txt
```

View the contents of the file with `less` or another pager. Note that the file is selected as the most relevant based upon the rank of the config channel containing it. This is accomplished within the **Configuration** tab of the **System Details** page.

4.2.3. Viewing Configuration Channels

To view the labels and names of the config channels that apply to the system, issue the command:

```
rhncfg-client channels
```

You should see output resembling:

```
Config channels: Label Name ----- ---- config-channel-17 config chan 2
config-channel-14 config chan 1
```

The following table lists the options available for `rhncfg-client get`:

Table 4.2. `rhncfg-client get` options

Option	Description
<code>--topdir=TOPDIR</code>	Make all file operations relative to this string.
<code>--exclude=EXCLUDE</code>	Excludes a file from being deployed with 'get/'. May be used multiple times.
<code>-h, --help</code>	Show help message and exit

4.2.4. Differentiating between Configuration Files

To view the differences between the config files deployed on the system and those stored by Red Hat Network, issue the command:

```
rhncfg-client diff
```

The output resembles the following:

```
[root@testsatellite root]# rhncfg-client diff
--- /etc/test
+++ /etc/test 2013-08-28 00:14:49.405152824 +1000
@@ -1 +1,2 @@
   This is the first line
+This is the second line added
```

In addition, you may include the `--topdir` option to compare config files in Red Hat Network with those located in an arbitrary (and unused) location on the client system, like so:

```
[root@ root]# rhncfg-client diff --topdir /home/test/blah/ /usr/bin/diff:
/home/test/blah/etc/example-config.txt: No such file or directory
/usr/bin/diff: /home/test/blah/var/spool/aalib.rpm: No such file or
directory
```

4.2.5. Verifying Configuration Files

To quickly determine if client configuration files are different than those associated with it via Red Hat Network, issue the command:

```
rhncfg-client verify
```

The output resembles the following:

```
modified /etc/example-config.txt /var/spool/aalib.rpm
```

The file `example-config.txt` is locally modified, while `aalib.rpm` is not.

The following table lists the options available for `rhncfg-client verify`:

Table 4.3. rhncfg-client verify options

Option	Description
<code>-v, --verbose</code>	Increase the amount of output detail. Displays differences in the mode, owner, and group permissions for the specified config file.
<code>-o, --only</code>	Only show files that differ.
<code>-h, --help</code>	Show help message and exit

4.3. USING THE RED HAT NETWORK CONFIGURATION MANAGER

Unlike the **Red Hat Network Configuration Client**, the **Red Hat Network Configuration Manager** (**rhncfg-manager**) is designed to maintain Red Hat Network's central repository of config files and channels, not those located on client systems. This tool offers a command line alternative to the configuration management features within the Red Hat Network website, as well as the ability to script some or all of the related maintenance.

It is intended for use by Config Administrators and requires an Red Hat Network username and password that has the appropriate permission set. The username may be specified in `/etc/sysconfig/rhn/rhncfg-manager.conf` or in the `[rhncfg-manager]` section of `~/.rhncfgrc`.

When the **Red Hat Network Configuration Manager** is run as root, it attempts to pull in needed configuration values from the **Red Hat Update Agent**. When run as a user other than root, you may have to make configuration changes within the `~/.rhncfgrc` file. The session file is cached in `~/.rhncfg-manager-session` to prevent logging in for every command.

The default timeout for the **Red Hat Network Configuration Manager** is 30 minutes. To alter this, add the `server.session_lifetime` option and new value to the `/etc/rhn/rhn.conf` file on the server running the manager, like so:

```
server.session_lifetime = 120
```

The **Red Hat Network Configuration Manager** offers these primary modes: `add`, `create-channel`, `diff`, `diff-revisions`, `download-channel`, `get`, `list`, `list-channels`, `remove`, `remove-channel`, `revisions`, `update`, and `upload-channel`.

Each mode offers its own set of options, which can be seen by issuing the following command:

```
rhncfg-manager mode --help
```

Replace `mode` with the name of the mode to be inspected:

```
rhncfg-manager diff-revisions --help
```

You can see such a list of options for the `add` mode at [Section 4.3.2, “Adding Files to a Configuration Channel”](#).

4.3.1. Creating a Configuration Channel

To create a config channel for your organization, issue the command:

```
rhncfg-manager create-channel channel-label
```

If prompted for your Red Hat Satellite username and password, provide them. The output resembles the following:

```
Red Hat Network username: rhn-user
Password:
Creating config channel channel-label Config channel channel-label created
```

Once you have created a config channel, use the remaining modes listed above to populate and maintain that channel.

4.3.2. Adding Files to a Configuration Channel

To add a file to a config channel, specify the channel label as well as the local file to be uploaded, such as:

```
rhncfg-manager add --channel=channel-label /path/to/file
```

In addition to the required channel label and the path to the file, you may use the available options for modifying the file during its addition. For instance, you may alter the path and file name by including the **--dest-file** option in the command, like:

```
rhncfg-manager add --channel=channel-label --dest-  
file=/new/path/to/file.txt/path/to/file
```

The output resembles the following:

```
Pushing to channel example-channel  
Local file >/path/to/file -> remote file /new/path/to/file.txt
```

The following table lists the options available for **rhncfg-manager add**:

Table 4.4. rhncfg-manager add options

Option	Description
-c CHANNEL --channel=CHANNEL	Upload files in this config channel
-d DEST_FILE --dest-file=DEST_FILE	Upload the file as this path
--delim-start=DELIM_START	Start delimiter for variable interpolation
--delim-end=DELIM_END	End delimiter for variable interpolation
-i, --ignore-missing	Ignore missing local files
--selinux-context=SELINUX_CONTEXT	Overwrite the SELinux context
-h, --help	show help message and exit

**NOTE**

By default, the maximum file size for configuration files is 128KB. If you need to change that value, find or create the following lines in the following files:

In `/usr/share/rhn/config-defaults/rhn_web.conf` (in bytes):

```
maximum_config_file_size = 131072
```

In `/usr/share/rhn/config-defaults/rhn_server.conf` (in bytes):

```
maximum_config_file_size = 131072
```

In `/usr/share/rhn/config-defaults/rhn_java.conf` (in kilobytes):

```
java.config_file_edit_size = 128
```

4.3.3. Differentiating between Latest Configuration Files

To view the differences between the configuration files on disk and the latest revisions in a channel, issue the command:

```
rhncfg-manager diff --channel=channel-label --dest-file=/path/to/file.txt
\ /local/path/to/file
```

You should see output resembling:

```
--- /tmp/dest_path/example-config.txt config_channel: example-channel
revision: 1
+++ /home/test/blah/hello_world.txt 2003-12-14 19:08:59.000000000 -0500
@@ -1 +1 @@
-foo
+hello, world
```

The following table lists the options available for `rhncfg-manager diff`:

Table 4.5. rhncfg-manager diff options

Option	Description
-c CHANNEL, --channel=CHANNEL	Get file(s) from this config channel
-r REVISION, --revision=REVISION	Use this revision
-d DEST_FILE, --dest-file=DEST_FILE	Upload the file as this path
-t TOPDIR, --topdir=TOPDIR	Make all files relative to this string
-h, --help	Show help message and exit

4.3.4. Differentiating between Various Versions

To compare different versions of a file across channels and revisions, use the `-r` flag to indicate which revision of the file should be compared and the `-n` flag to identify the two channels to be checked. See [Section 4.3.11, “Determining the Number of File Revisions”](#) for related instructions. Specify only one file name here, since you are comparing the file against another version of itself. For example:

```
rhncfg-manager diff-revisions -n=channel-label1 -r=1 -n=channel-label2 -
r=1 /path/to/file.txt
```

The output resembles the following:

```
--- /tmp/dest_path/example-config.txt 2004-01-13 14:36:41 \ config
channel: example-channel2 revision: 1
--- /tmp/dest_path/example-config.txt 2004-01-13 14:42:42 \ config
channel: example-channel3 revision: 1
@@ -1 +1,20 @@
-foo
+blah
+-----BEGIN PGP SIGNATURE-----
+Version: GnuPG v1.0.6 (GNU/Linux)
+Comment: For info see http://www.gnupg.org
+
+iD8DBQA9ZY6vse4XmfJPGwgRAsHcAJ9ud9dabUcdscdcqB8AZP7e0Fua0NmKsdhQCe0WHX
+VsDTfen2NwdwwPaTM+S+Cow=
+=Ltp2
+-----END PGP SIGNATURE-----
```

The following table lists the options available for `rhncfg-manager diff-revisions`:

Table 4.6. rhncfg-manager diff-revisions options

Option	Description
<code>-c CHANNEL, --channel=CHANNEL</code>	Use this config channel
<code>-r REVISION, --revision=REVISION</code>	Use this revision
<code>-h, --help</code>	Show help message and exit

4.3.5. Downloading All Files in a Channel

To download all the files in a channel to disk, create a directory and issue the following command:

```
rhncfg-manager download-channel channel-label --topdir .
```

The output resembles the following:

```
Copying /tmp/dest_path/example-config.txt -> \
blah2/tmp/dest_path/example-config.txt
```

The following table lists the options available for `rhncfg-manager download-channel`:

Table 4.7. rhncfg-manager download-channel options

Option	Description
-t TOPDIR, --topdir=TOPDIR	Directory all the file paths are relative to. This option must be set.
-h, --help	Show help message and exit

4.3.6. Getting the Contents of a File

To direct the contents of a particular file to stdout, issue the command:

```
rhncfg-manager get --channel=channel-label \ /tmp/dest_path/example-  
config.txt
```

You should see the contents of the file as output.

4.3.7. Listing All Files in a Channel

To list all the files in a channel, issue the command:

```
rhncfg-manager list channel-label
```

You should see output resembling:

```
Files in config channel `example-channel13': /tmp/dest_path/example-  
config.txt
```

The following table lists the options available for **rhncfg-manager get**:

Table 4.8. rhncfg-manager get options

Option	Description
-c CHANNEL, --channel=CHANNEL	Get file(s) from this config channel
-t TOPDIR, --topdir=TOPDIR	Make all files relative to this string
-r REVISION, --revision=REVISION	Get this file revision
-h, --help	Show help message and exit

4.3.8. Listing All Configuration Channels

To list all of your organization's configuration channels, issue the command:

```
rhncfg-manager list-channels
```

The output resembles the following:

```
Available config channels: example-channel example-channel2 example-
channel3 config-channel-14 config-channel-17
```

Note that this does not list **local_override** or **server_import** channels.

4.3.9. Removing a File from a Channel

To remove a file from a channel, issue the command:

```
rhncfg-manager remove --channel=channel-label /tmp/dest_path/example-
config.txt
```

If prompted for your Red Hat Network username and password, provide them. You should see output resembling:

```
Red Hat Network username: rhn-user Password: Removing from config channel
example-channel3 /tmp/dest_path/example-config.txt removed
```

The following table lists the options available for **rhncfg-manager remove**:

Table 4.9. rhncfg-manager remove options

Option	Description
-c CHANNEL, --channel=CHANNEL	Remove files from this config channel
-t TOPDIR, --topdir=TOPDIR	Make all files relative to this string
-h, --help	Show help message and exit

4.3.10. Deleting a Configuration Channel

To destroy a configuration channel in your organization, issue the command:

```
rhncfg-manager remove-channel channel-label
```

The output resembles the following:

```
Removing config channel example-channel Config channel example-channel
removed
```

4.3.11. Determining the Number of File Revisions

To find out how many revisions (revisions go from 1 to N where N is an integer greater than 0) of a file/path are in a channel, issue the following command:

```
rhncfg-manager revisions channel-label /tmp/dest_path/example-config.txt
```

The output resembles the following:

```
Analyzing files in config channel example-channel \
/tmp/dest_path/example-config.txt: 1
```

4.3.12. Updating a File in a Channel

To create a new revision of a file in a channel (or add the first revision to that channel if none existed before for the given path), issue the following command:

```
rhncfg-manager update \ --channel=channel-label --dest-
file=/path/to/file.txt /local/path/to/file
```

The output resembles the following:

```
Pushing to channel example-channel: Local file example-
channel/tmp/dest_path/example-config.txt -> \ remote file
/tmp/dest_path/example-config.txt
```

The following table lists the options available for **rhncfg-manager update**:

Table 4.10. rhncfg-manager update options

Option	Description
-c CHANNEL, --channel=CHANNEL	Upload files in this config channel
-d DEST_FILE, --dest-file=DEST_FILE	Upload the file as this path
-t TOPDIR, --topdir=TOPDIR	Make all files relative to this string
--delim-start=DELIM_START	Start delimiter for variable interpolation
--delim-end=DELIM_END	End delimiter for variable interpolation
-h, --help	Show help message and exit

4.3.13. Uploading Multiple Files at Once

To upload multiple files to a config channel from local disk at once, issue the command:

```
rhncfg-manager upload-channel --topdir=topdir channel-label
```

The output resembles the following:

```
Using config channel example-channel4 Uploading /tmp/ola_world.txt from
blah4/tmp/ola_world.txt
```

The following table lists the options available for **rhncfg-manager upload-channel**:

Table 4.11. rhncfg-manager upload-channel options

Option	Description
-t TOPDIR, --topdir=TOPDIR	Directory all the file paths are relative to
-c CHANNEL, --channel=CHANNEL	List of channels the config info will be uploaded into. Channels delimited by ','. Example: --channel=foo,bar,baz
-h, --help	Show help message and exit

4.4. USING THE RED HAT SATELLITE COMMAND LINE TOOL (SPACECMD)

The **spacecmd** tool interacts with Red Hat Satellite's XML-RPC API. This provides users with a simple way of executing Satellite functionality from the command line.



NOTE

An initial run of **spacecmd** requires your username and password. This opens a session ticket for the chosen user and all subsequent usage of **spacecmd** uses this session until it expires, after one hour. Change the user and password using the **-u USERNAME** and **-p PASSWORD** options.

spacecmd uses two methods of execution.

From the Interactive Shell

Run **spacecmd** alone to start the interactive shell.

```
[root@satellite57 ~]# spacecmd
Welcome to spacecmd, a command-line interface to Spacewalk.

Type: 'help' for a list of commands
      'help <cmd>' for command-specific help
      'quit' to quit

INFO: Connected to https://localhost/rpc/api as admin
spacecmd {SSM:0}>
```

This displays the **spacecmd** prompt, which also indicates the number of system attached to the System Set Manager (SSM).

To run commands, enter them into the shell. For example, to list all systems, run **system_list**:

```
spacecmd {SSM:0}> system_list
system001.example.com
system002.example.com
system003.example.com
system004.example.com
...
```

To list the base channel for a system, run **system_list** followed by the name of the system:

```
spacecmd {SSM:0}> system_listbasechannel system001.example.com
rhel-x86_64-server-6
```

List all commands using the **help** command.

From the Terminal

Execute **spacecmd** commands directly from the your Linux terminal. For example, use **spacecmd system_list** to list all your systems:

```
[root@satellite57 ~]# spacecmd system_list
INFO: Connected to https://localhost/rpc/api as admin
system001.example.com
system002.example.com
system003.example.com
system004.example.com
...
```

Or list the base channel for a system with **spacecmd system_listbasechannel systemname**:

```
[root@satellite57 ~]# spacecmd system_listbasechannel
system001.example.com
INFO: Connected to https://localhost/rpc/api as admin
rhel-x86_64-server-6
```

List all commands with **spacecmd help**.

4.5. USING THE RED HAT SATELLITE FINAL ARCHIVE TOOL (SPACEWALK-FINAL-ARCHIVE)

The **spacewalk-final-archive** is used to generate a final archive of your Red Hat Satellite 5 server before decommissioning it. The command generates an archive file found at **/tmp/spacewalk-final/final-archive.tar.bz2**. This archive includes:

- A backup of the database stored in the **archive/db_backup** directory. This backup is created with the **db-control** command.
- A copy of all relevant system files stored in the **archive/debug** directory. This backup is created with the **spacewalk-debug** command.
- A final copy of all reports in CSV format stored in the **archive/reports** directory. This backup is created with the **spacewalk-report** command.
- Transition data in CSV format for use with Red Hat Satellite 6 stored in the **archive/transition** directory. This backup is created with the **spacewalk-export** command.

To start a full archive process, run the command on its own:

```
[root@satellite57 ~]# spacewalk-final-archive
```

Use the **-h** option to see other options to restrict certain content from the archive or to change the archive export directory.

CHAPTER 5. CLONING SOFTWARE CHANNELS AND ERRATA

Use the **spacewalk-clone-by-date** command to create custom cloned Red Hat Enterprise Linux channels based on the date an erratum was made available to the Red Hat Enterprise Linux system.

5.1. FEATURES

The following features are available with **spacewalk-clone-by-date**:

- Cloning the channel errata and associated package states as they were on a specific date
- Automating the cloning by scripts and template files
- Removing or blocking packages from channels
- Resolving package dependencies within the parent and child channels
- Filtering and acting on specific errata while ignoring others. For example, acting only on security errata and ignoring bugfixes and enhancements.



NOTE

You need to run the **spacewalk-clone-by-date** command as the *root user* and the **username** needs to be either an Organizational Administrator or Channel Administrator.



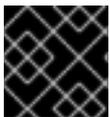
IMPORTANT

Use of **spacewalk-clone-by-date** is limited to Red Hat Enterprise Linux 5 and higher versions because **spacewalk-clone-by-date** uses **yum** metadata to complete dependency resolution.

5.2. EXAMPLE USAGE

The example below clones the **rhel-i386-server-5** channel errata as it is on January 1st, 2012, into the channel named **my-clone-RHEL-5**.

```
# spacewalk-clone-by-date --username=your_username --
password=your_password --server=satellite_server_url --channels=rhel-i386-
server-5 my-clone-RHEL-5 --to_date=2012-01-01
```



IMPORTANT

Ensure the cloned channel name contains no spaces.

The example below will only clone security errata released on or before January 1st, 2012, ignoring any kernel updates or vim-extended packages. The command will also run the cloning process in the background on the Satellite.

```
# spacewalk-clone-by-date --username=your_username --
password=your_password --server=satellite_server_url --channels=rhel-i386-
server-5 my-clone-RHEL-5 --to_date=2012-01-01 --security_only --background
--blacklist=kernel,vim-extended --assumeeyes
```



See the manual page for **spacewalk-clone-by-date** for more information about the available options and how to use them.

CHAPTER 6. MAINTAINING SYSTEM SECURITY USING OPENS CAP

The *Security Certification and Authorization Package (SCAP)* is a standardized compliance checking solution for enterprise-level Linux infrastructures. It is a line of specifications maintained by the National Institute of Standards and Technology (NIST) for maintaining system security for enterprise systems.

Red Hat Satellite Server 5.5 and later use **OpenSCAP** to implement the SCAP specifications. **OpenSCAP** is an auditing tool that utilizes the Extensible Configuration Checklist Description Format (XCCDF). XCCDF is a standard way of expressing checklist content and defines security checklists. It also combines with other specifications such as Common Platform Enumeration (CPE), Common Configuration Enumeration (CCE), and Open Vulnerability and Assessment Language (OVAL), to create a SCAP-expressed checklist that can be processed by SCAP-validated products.

6.1. OPENS CAP FEATURES

OpenSCAP verifies the presence of patches by using content produced by the [Red Hat Security Response Team \(SRT\)](#), it checks system security configuration settings, and examines systems for signs of compromise by using rules based on standards and specifications.

6.2. OPENS CAP PREREQUISITES

To effectively use OpenSCAP, the following must be available:

- A tool to verify that a system conforms to a standard.

Satellite Server 5.5 and later use OpenSCAP as an auditing feature. This allows you to use the web interface to schedule and view compliance scans for any system.

- SCAP content.

You can generate your own SCAP content if you have an understanding of at least XCCDF or OVAL. XCCDF content is also frequently published online under open source licenses, and you can customize this content to suit your needs instead.



NOTE

Red Hat supports the use of templates to evaluate your systems. However, custom content authoring of these templates is not supported.

Some examples of bodies that publish XCCDF content are:

- [The United States Government Configuration Baseline \(USGCB\)](#): Official SCAP content for desktops within federal agencies that has been developed at NIST in collaboration with Red Hat, Inc. and the United States Department of Defense (DoD) using OVAL.
- Community-provided content:
 - [SCAP Security Guide](#): Active community-run content that sources from the USGCB requirements and widely-accepted policies and contains profiles for desktop, server, and FTP server. Suitable for **Red Hat Enterprise Linux 6** and **JBoss Enterprise Application Server 5**.

- OpenSCAP Content for **Red Hat Enterprise Linux 6**: The openscap-content package from the **Red Hat Enterprise Linux 6** Optional Channel also provides default content guidance by means of a template.

SCAP was created to provide a standardized approach to maintaining system security, and the standards that are used will therefore continually change to meet the needs of the community and enterprise businesses. New specifications are governed by [NIST's SCAP Release cycle](#) in order to provide a consistent and repeatable revision workflow.

6.3. RED HAT SATELLITE PREREQUISITES FOR USING OPENSAP

The following sections describe the prerequisites for using OpenSCAP on **Red Hat Satellite Servers** and **Satellite Clients**.

Package Requirements

- **Satellite Server: Satellite 5.5** or later.
- **Satellite Client:** spacewalk-oscsp package (available from the Red Hat Network Tools Child Channel).

Entitlement Requirements

A Management entitlement is required for scheduling scans.

Other Requirements

Satellite Client: Distribution of the XCCDF content to all client machines.

You can distribute the XCCDF content to client machines using any of the following methods:

- Traditional methods, such as CD, USB, NFS, SCP, FTP.
- Satellite scripts.
- RPM packages.

Custom RPMs are the recommended way to distribute SCAP content to other machines. RPM packages can be signed and verified to ensure their integrity. Installation, removal, and verification of RPM packages can be managed from the user interface.

6.4. PERFORMING AUDIT SCANS

OpenSCAP integration in **Red Hat Satellite Server** provides the ability to perform audit scans on client systems. This section describes the methods available for performing these scans.

6.4.1. Using the Web Interface to Perform Audit Scans

This section describes how to use the **Satellite** web interface to perform audit scans.

Procedure 6.1. To Perform an Audit Scan Using the Web Interface:

1. Log in to the Satellite web interface.
2. Click **Systems** → **system_name**.

3. Click **Audit** → **Schedule**.
4. Complete the **Schedule New XCCDF Scan** page. See [Section 6.5.2.3, “Schedule Page”](#) for information about the fields on this page.

**WARNING**

The XCCDF content is validated before it is run on the remote system. Specifying invalid command-line arguments can cause **spacewalk-oscapp** to fail to validate or run. Due to security concerns the **oscapp xccdf eval** command only accepts a limited set of parameters.

NOTE

You can run the **rhn_check** command to ensure that the action is being picked up by the client system.

```
# rhn_check -vv
```

If **rhnsd** or **osad** are running on the client system, the action will be picked up by these services. To check if they are running, run *one* of the following commands.

For Red Hat Enterprise Linux 5 and 6:

```
# service rhnsd start
# chkconfig rhnsd on
OR
# service osad start
# chkconfig osad on
```

For Red Hat Enterprise Linux 7:

```
# systemctl enable rhnsd
# systemctl start rhnsd
OR
# systemctl enable osad
# systemctl start osad
```

To view the results of the scan, see [Section 6.4.3, “Viewing the Results of SCAP Audits”](#).

6.4.2. Using the API to Perform Audit Scans

This section describes how to use the **Satellite** API to perform audit scans.

Procedure 6.2. To Perform an Audit Scan Using the API:

1. Choose an existing script or create a script for scheduling a system scan through **system.scap.scheduleXccdfScan**, the front-end API, for example:

```
#!/usr/bin/python
import xmlrpclib
client = xmlrpclib.Server('https://satellite.example.com/rpc/api')
key = client.auth.login('username', 'password')
client.system.scap.scheduleXccdfScan(key, 1000010001,
    '/usr/local/share/scap/usgcb-rhel5desktop-xccdf.xml',
    '--profile united_states_government_configuration_baseline')
```

Where:

- o 1000010001 is the **system ID (sid)**.
 - o **/usr/local/share/scap/usgcb-rhel5desktop-xccdf.xml** is the path to the content location on the client system. In this case, it assumes USGCB content in the **/usr/local/share/scap** directory.
 - o **--profile united_states_government_configuration_baseline** is an additional argument to the **oscap** command. In this case, it is using the USGCB.
2. Run the script on the command-line interface of any system. The system needs the appropriate Python and XML-RPC libraries installed.

NOTE

You can run the **rhn_check** command to ensure that the action is being picked up by the client system.

```
# rhn_check -vv
```

If **rhnsd** or **osad** are running on the client system, the action will be picked up by these services. To check if they are running, run *one* of the following commands:

For Red Hat Enterprise Linux 5 and 6:

```
# service rhnsd start
# chkconfig rhnsd on
OR
# service osad start
# chkconfig osad on
```

For Red Hat Enterprise Linux 7:

```
# systemctl enable rhnsd
# systemctl start rhnsd
OR
# systemctl enable osad
# systemctl start osad
```

6.4.3. Viewing the Results of SCAP Audits

There are three methods of viewing the results of finished scans:

- Using the web interface. After the scan has finished, the results are available on the **Audit** page of specific system. See [Section 6.5, “OpenSCAP Satellite Web Interface”](#).
- Using the API functions in handler `system.scap`.
- Using the `spacewalk-report` command, as follows:

```
# spacewalk-report system-history-scap
# spacewalk-report scap-scan
# spacewalk-report scap-scan-results
```

6.5. OPENS CAP SATELLITE WEB INTERFACE

The following sections describe the pages in the Red Hat Satellite web interface that provide access to OpenSCAP and its features.

6.5.1. OpenSCAP Scans Page

Click the **Audit** tab on the top navigation bar to display the OpenSCAP Scans page. This is the "overview" page for all OpenSCAP functionality in Satellite Server. Use this page to view, search for, and compare completed scans.

6.5.1.1. All Scans

The **All Scans** page is the default page that appears on the **Audit** tab. This page displays all the completed OpenSCAP scans that the viewer has permission to see. Permissions for scans are derived from system permissions.

For each scan, the following information is displayed:

- System: the system that was scanned.
- XCCDF Profile: the evaluated profile.
- Completed: the time the scan was completed.
- Satisfied: the number of rules that were satisfied. A rule is considered to be Satisfied if the result of the evaluation is either Pass or Fixed.
- Dissatisfied: the number of rules that were not satisfied. A rule is considered to be Dissatisfied if the result of the evaluation is Fail.
- Unknown: the number of rules that failed to evaluate. A rule is considered to be Unknown if the result of the evaluation is Error, Unknown or Not Checked.

The evaluation of XCCDF rules may also return status results such as **Informational**, **Not Applicable**, or **not Selected**. In such cases, the given rule is not included in the statistics on this page. See **System Details** → **Audit** for information about these types of results.

6.5.1.2. XCCDF Diff

XCCDF Diff is an application which visualizes the comparison of two XCCDF scans. It shows metadata for two scans as well as the lists of results.

Click the appropriate icon on the **List Scans** page to access the **diff** output of similar scans.

Alternatively, or you can specify the ID of arbitrary scans.

Items that show up in only one of the compared scans are considered to be "varying". Varying items are always highlighted in beige. There are three possible comparison modes: **Full Comparison** which shows all the scan items, **Only Changed Items** which shows items that have changed, and finally **Only Invariant Items** which shows unchanged or similar items.

6.5.1.3. Advanced Search

Use the Advanced Search page to search through your scans according to specified criteria, including:

- Rule results.
- Targeted machine.
- Time frame of the scan.

The search either returns a list of results or a list of scans which are included in the results.

6.5.2. Systems Audit Page

Use the **Systems Audit** page to schedule and view compliance scans for a particular system. Scans are performed by the OpenSCAP tool, which implements NIST's standard **Security Content Automation Protocol (SCAP)**. Before you scan a system, ensure that the SCAP content is prepared and all prerequisites are met.

To display the **Systems Audit** page, click **Systems** → **system_name** → **Audit**.

6.5.2.1. List Scans

This page displays a summary of all scans completed on the selected system. The following columns are displayed:

Table 6.1. OpenSCAP Scan Labels

Column Label	Definition
XCCDF Test Result	The scan test result name. This is also a link to the detailed results of the scan.
Completed	The exact time the scan finished.
Compliance	The unweighted pass:fail ratio of compliance based on the standard that was used.
P	The number of checks that passed.
F	The number of checks that failed.
E	The number of errors that occurred during the scan.
U	Unknown

Column Label	Definition
N	Not applicable to the machine.
K	Not checked.
S	Not selected.
I	Informational
X	Fixed
Total	Total number of checks.

Each entry begins with an icon indicating the results of a comparison to a previous similar scan. The icons indicate the following:

-  No difference between the compared scans.
-  Arbitrary differences between the compared scans.
-  Major differences between the compared scans. Either there are more failures than the previous scan or less passes.
-  No comparable scan was found, and therefore no comparison was made.

6.5.2.2. Scan Details

The **Scan Details** page contains the results of a single scan. This page is divided into two sections:

Details of the XCCDF Scan

This section displays various details about the scan, including:

- **File System Path:** The path to the XCCDF file used for the scan.
- **Command-line Arguments:** Any additional command-line arguments that were used.
- **Profile Identifier:** The profile identifier used for the scan.
- **Profile Title:** The title of the profile used for the scan.
- **Scan's Error output:** Any errors encountered during the scan.

XCCDF Rule Results

The rule results provide the full list of XCCDF rule identifiers, identifying tags, and the result for each of these rule checks. This list can be filtered by a specific result.

6.5.2.3. Schedule Page

Use the **Schedule New XCCDF Scan** page to schedule new scans for specific machines. Scans occur at the system's next scheduled check-in that occurs *after* the date and time specified.

The following fields can be configured:

- **Command-line Arguments:** Optional arguments to the **oscap** command, either:
 - **--profile PROFILE:** Specifies a particular profile from the XCCDF document.

Profiles are determined by the **Profile** tag in the XCCDF XML file. Use the **oscap** command to see a list of profiles within a given XCCDF file, for example:

```
$ oscap info /usr/share/openscap/scap-rhel6-xccdf.xml
Document type: XCCDF Checklist
Checklist version: 1.1
Status: draft
Generated: 2011-10-12
Imported: 2012-11-15T22:10:41
Resolved: false
Profiles:
    RHEL6-Default
```

If not specified, the default profile is used.



NOTE

Some early versions of OpenSCAP in Red Hat Enterprise Linux 5 require that you use the **--profile** option or the scan will fail.

- **--skip-valid:** Do not validate input and output files. You can use this option to bypass the file validation process if you do not have well-formed XCCDF content.
- **Path to XCCDF Document:** This is a required field. The **path** parameter points to the XCCDF content location on the client system. For example:
/usr/local/scap/dist_rhel6_scap-rhel6-oval.xml



WARNING

The XCCDF content is validated before it is run on the remote system. Specifying invalid arguments can cause **spacewalk-oscaps** to fail to validate or run. Due to security concerns, the **oscap xccdf eval** command only accepts a limited set of parameters.

For information about how to schedule scans using the Satellite web interface, see [Section 6.4.1, “Using the Web Interface to Perform Audit Scans”](#).

CHAPTER 7. REPORTING CLIENT SOFTWARE FAILURES

You can take advantage of Red Hat Satellite's software failure reporting capabilities and the Automatic Bug Reporting Tool (ABRT) to extend the overall reporting functionality of your systems. This extended functionality allows your clients to automatically report software failures captured by ABRT to the Satellite server, and also to process the captured failures in a centralized fashion. You can use either the webUI or the API to process these failure reports.

For information about setting up Red Hat Satellite tools for ABRT on client systems, see the *Red Hat Satellite Client Configuration Guide*.

7.1. VIEWING SOFTWARE FAILURES FOR A SINGLE CLIENT

The following procedure describes how to view software failure reports for a single client system with Satellite's ABRT tools installed.

Procedure 7.1. To View Software Failures for a Single Client:

1. Log in to the Red Hat Satellite Web UI.
2. Click **Systems** → **system_name** → **Software** → **Software Crashes** to see the list of software failures that occurred on the registered system.
3. Click the required failure to display its details and the files captured for this software failure report.

7.2. GROUPING SIMILAR SOFTWARE FAILURES

The Red Hat Satellite Web UI provides a page to group software failures across all systems by **Crash UUID**. This helps with identifying similar software crashes on your clients.

Procedure 7.2. To view similar software failures across clients

1. Log into your Red Hat Satellite Web UI.
2. Click **Systems** → **Software Crashes** to see a list of all software failures across all registered systems.
3. Click the on a **Crash UUID** to see the systems affected by the software failure.
4. Click on a specific system to see details and the files captured for the individual software failure report.

The software failure report from the client system displays.

7.3. CHANGING ORGANIZATION-WIDE SETTINGS FOR SOFTWARE FAILURE REPORTS

Red hat Satellite provides the ability to change the organization-wide settings for software failure reports. For example, with every software failure, clients upload the files captured by ABRT during the failure to your Satellite server. Because these files may be of arbitrary length, you can configure an organization-wide size limit for the upload of a single crash file.

The following procedure shows how to modify organization-wide settings for software failures.

Procedure 7.3. To Change the Organization-wide Settings for Software Failures:

1. In the Satellite Web UI, click **Admin** → *<organization_name>* → **Configuration**.
2. Modify the desired organization-wide and upload size settings, and then click **Update Organization**.

7.4. LOG FILES OF SOFTWARE FAILURES

The log files captured by ABRT as a result of software failures are uploaded to your Satellite server for every failure report. You can download these files using either the Web UI or the API. On the Satellite server, these log files are physically stored in the `/var/satellite/systems/$org_id/$system_id/crashes/$crash_name/` directory.

CHAPTER 8. GENERATING RED HAT SATELLITE REPORTS

This chapter is designed to help you generate reports from Red Hat Satellite.

Red Hat Satellite contains a number of command-line reports:

- **channel-packages** - Packages in channels
- **channels** - Channel report
- **custom-info** - Display system custom info
- **entitlements** - Entitlement and channel list and usage
- **errata-channels** - List of errata in channels
- **errata-list** - Errata information based upon compliance checks against systems
- **errata-list-all** - List of all erratas
- **errata-systems** - Listing of each errata applicable to each affected system
- **inactive-systems** - Inactive systems in Satellite
- **inventory** - Inventory report
- **kickstartable-trees** - List of kickstartable trees
- **packages-updates-all** - List of packages that can be upgraded
- **packages-updates-newest** - List of packages that can be upgraded
- **scap-scan** - Results of OpenSCAP xccdf evaluation
- **scap-scan-results** - Results of OpenSCAP xccdf evaluation
- **system-crash-count** - Crash count for systems
- **system-crash-details** - Crash details for systems
- **system-currency** - System currency list
- **system-groups** - System groups in Satellite
- **system-groups-keys** - Activation keys for system groups
- **system-groups-systems** - Systems in system groups
- **system-groups-users** - System groups users report
- **system-history** - System event history
- **system-history-channels** - Channel event history
- **system-history-configuration** - Configuration event history

- **system-history-entitlements** - System entitlement event history
- **system-history-errata** - Errata event history
- **system-history-kickstart** - Kickstart event history
- **system-history-packages** - Package event history
- **system-history-scap** - OpenSCAP event history
- **system-packages-installed** - Packages installed on systems
- **users** - Users in the system
- **users-systems** - Systems administered by individual users

To generate a report, use the **spacewalk-report** command as follows:

```
# spacewalk-report report-name
```

This command generates the selected report as comma-separated value (CSV) output.



NOTE

The logging feature of Satellite is added by default in fresh installations of Satellite version 5.6 and later. If the Satellite is upgraded from a version below 5.6 the logging feature will be turned on at the time of upgrade and from that point all events will be audited.

This means that all users created before the upgrade will get logged from the time of upgrade. The past creation of a user and any past events will not appear in the log but all future events will be logged.

To more information, run the **spacewalk-report** command with the **-h** option.

CHAPTER 9. SCHEDULING RED HAT SATELLITE ADMINISTRATIVE TASKS

Red Hat Satellite allows organization administrators to regularly perform long-term operations using the **taskomatic** service. These operations are segregated into individual tasks and grouped logically into a *bunch* that is defined by schedules. You can modify these schedules to execute at specific time intervals. Satellite schedules are used to:

- Remove the administrative burden from the organizational administrator by automating tasks.
- Schedule operational tasks for time frames that will not tax the organization's daily network traffic.

Red Hat Satellite provides default schedules that trigger specific task bunches.

Table 9.1. Default Schedules in Red Hat Satellite 5.8

Schedule Name	Bunch Name	Bunch Function
channel-repodata-default	channel-repodata-bunch	Generates channel repository data.
cleanup-data-default	cleanup-data-bunch	Cleans up orphaned and outdated data.
clear-taskologs-default	clear-taskologs-bunch	Clears taskomatic run log history.
cobbler-sync-default	cobbler-sync-bunch	Applies any cobbler configuration changes.
compare-configs-default	compare-configs-bunch	Schedules a comparison of configuration files on all systems.
daily-status-queue	daily-status-bunch	Sends daily report.
errata-cache-default	errata-cache-bunch	Recalculates errata cache for a given server or channel.
errata-queue-default	errata-queue-bunch	Processes errata.
kickstart-cleanup-default	kickstart-cleanup-bunch	Cleans up stale kickstart files.
kickstartfile-sync-default	kickstartfile-sync-bunch	Synchronizes kickstart profiles that were generated using the wizard.
package-cleanup-default	package-cleanup-bunch	Cleans up orphaned packages.
sandbox-cleanup-default	sandbox-cleanup-bunch	Cleans up sandbox.

Schedule Name	Bunch Name	Bunch Function
satcert-check-default	satcert-check-bunch	Determines expiration status of Satellite certificate.
session-cleanup-default	session-cleanup-bunch	Deletes expired rows from the PXTSessions table to prevent it from growing too large.

9.1. SCHEDULING A RUN

A *run* is a single execution of a bunch according to a configured schedule. You can schedule a run based on the default template provided by Red Hat Satellite, or you can create an entirely new schedule.

Procedure 9.1. Creating a Schedule Template

1. Log in to Satellite as the Organization Administrator.
2. Click **Admin** → **Task Schedules** → **Create Schedule**.
3. Complete the following fields:
 - Schedule Name: must begin with a letter and contain only lowercase characters, hyphens, periods, underscores, or numerals.
 - Bunch: the default bunch of administrative tasks the administrator can choose from.
 - Frequency

The following frequency options are available:

- **Disable Schedule:** only recommended for administrators who have advanced knowledge of the scheduled tasks and their consequences. Disabling schedules can change Satellite behavior.
 - **Daily:** creates a daily schedule for a specific time of day.
 - **Weekly:** creates a weekly schedule for a specific day and time of day.
 - **Monthly:** creates a monthly schedule for a specific day and time of day.
 - **Custom Quartz Format:** this format relies on cron expressions to define the schedule. For more information about this format, see the crontab man page (**man 5 crontab**.)
4. Click **Create Schedule**.

Procedure 9.2. Editing Schedule Templates

As an alternative to creating a new schedule, you can edit the default templates. To edit one of the existing templates:

1. Log in to Satellite as the Organization Administrator.
2. Click **Admin** → **Task Schedules**.

3. Click the schedule that you want to modify.
4. Change the Frequency type as required.
5. Click **Edit Schedule**.

9.2. SETTING UP A SELF-SUBSCRIBED RED HAT SATELLITE

A self-subscribed Red Satellite 5 Server is registered to itself rather than the Red Hat Content Delivery Network (CDN). A Satellite server that is not registered to itself is registered to Red Hat CDN, then activated as a Satellite server. The Satellite can then use the **cdn-sync** command get new packages and content from the Red Hat CDN.

Once a self-subscribed Satellite server is set up, it gets content in the same way but through a base channel hosted on the Satellite server itself rather than through the Red Hat CDN. This process allows control of the Red Hat Enterprise Linux packages on the Satellite server, in the same manner as clients registered to the Satellite.



IMPORTANT

Self-subscribed Satellites have several limitations. These are:

- A self-subscribed Satellite treats the self-registration as it does any other client system registration. To prevent accidental changes to your self-subscribed Satellite lock the self-subscribed Satellite's system profile using **Lock system** in the system profile.
- A self-subscribed Satellite cannot use **osad**. Installing the client-side **osad** package will break the provisioning feature of Satellite.

9.2.1. Installing and Configuring a Self-Subscribed Satellite

Procedure 9.3. Installing and Configuring a Self-Subscribed Satellite

1. Install Red Hat Enterprise Linux following the instructions provided in *Scenario 1: Installing Satellite with Embedded Database* in the *Red Hat Satellite 5 Installation Guide*. The installation has to be run in disconnected mode.
2. After the installation, change the value of the **disconnected** parameter back to **0** in the **/etc/rhn/rhn.conf** configuration file so that you are able to use the **cdn-sync** command in the next step.

```
disconnected=0
```

3. Use the **cdn-sync** command to download and import the base channel that matches the version of Red Hat Enterprise Linux installed on the Satellite server. The **cdn-sync** command can import the necessary files from the Red Hat CDN. Optionally, use the **satellite-sync** command to import the base channel content ISOs available for download.
4. Use the Satellite 5 web interface to create a cloned channel of the imported base channel. See [Chapter 5, Cloning Software Channels and Errata](#) for more information.
5. Install the client side Satellite SSL certificate onto the Satellite server using the following command.

```
# rpm -Uvh /var/www/html/pub/rhn-org-trusted-ssl-cert-1.0-1.noarch.rpm
```

6. Reconfigure **Red Hat Update Agent** to use the Satellite hostname and SSL certificate by editing the `/etc/sysconfig/rhn/up2date`. Change the following options:

```
sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
noSSLServerURL=http://satellite-server-hostname/XMLRPC
serverURL=https://satellite-server-hostname/XMLRPC
```

7. Register the Satellite server:

```
# rhnreg_ks --username satellite_username --password satellite_password
```

8. Log into the Satellite web interface. Go to **System Details** → **Channels** then select the cloned base channel from the drop-down menu. Click **Modify Base Channel**.

9.2.2. Testing Self-Subscribed Satellite Functionality

Procedure 9.4. Testing Self-Subscribed Satellite Functionality

1. Verify that synchronization from the Red Hat CDN is working.

```
# cdn-sync -l
```

Running the `cdn-sync` command should list all the channels available to be synchronized. The output should resemble the following:

```
17:34:02 p = previously imported/synced channel
17:34:02 . = channel not yet imported/synced
```

2. Verify the source of package updates.

- o Red Hat Enterprise Linux 6:

```
yum check-update
```

This should display information indicating that packages are downloaded from the Satellite rather than from the Red Hat CDN.

9.2.3. Client-Side Application Functionality with a Self-Subscribed Satellite

Red Hat provides various client-side tools for interaction with various aspects of a Red Hat Satellite. The list below outlines whether or not a client-side application functions on a Self-Subscribed Satellite server.



IMPORTANT

There are several important things to note about a self-subscribed Satellite:

- If a client-side application is not listed here it has not been tested.
- Red Hat recommends that Administrators lock the registered Self-Subscribed Satellite within the Satellite web interface. This prevents any scheduled event from executing. Before unlocking the Satellite review the pending events and delete those you do not want to run.
- Red Hat recommends Administrators entitle the Self-Subscribed Satellite to the Management level but not with the Provisioning entitlement. This helps to avoid possible harmful or accidental changes to the Satellite server.
- If the self-subscribed Satellite has been granted a Provisioning entitlement do not attempt to use the Satellite to re-provision itself. The Satellite will attempt to perform the re-installation of the Red Hat Enterprise Linux operating system but on reboot the Red Hat installation program will be unable to download the necessary packages from the Satellite to perform the installation. There is a high risk of data loss and service interruption for your Satellite, especially if external kickstart trees are used.

- *Red Hat Update Agent Tools*

The **rhncfg-check**, **rhnsd** and **yum** packages will all function normally on a self-subscribed Satellite.

- *Push*

The **osad** package will not install. The **osad** package is used to push packages to client systems but it conflicts with the server-side **osa-dispatcher** package. Do not attempt to force the installation of **osad** on a self-subscribed Satellite.

- *Configuration Client Tool*

The **rhncfg-client** package will function normally.

- *Configuration Management Tool*

The **rhncfg-manager** package will function normally.

- *Custom Info*

The **rhncfg-custom-info** package will function normally.

CHAPTER 10. TROUBLESHOOTING

This chapter provides tips for determining the cause of and resolving the most common errors associated with Red Hat Satellite. If you need additional help, contact Red Hat Network support at <https://access.redhat.com/support/>. Log in using your Satellite-entitled account to see the full list of options.

To begin troubleshooting general problems, examine the log file or files related to the component exhibiting failures. A useful exercise is to issue the `tail -f` command for all log files and then run `yum list`. You should then examine all new log entries for potential clues.

10.1. Disk Space

Q: My disk space filled up fast. What happened and what should I do?

A: A common issue is full disk space. An almost sure sign of this is the appearance of halted writing in the log files. If logging stopped during a write, such as mid-word, the hard disks may be full. To confirm this, run this command and check the percentages in the **Use%** column:

```
# df -h
```

In addition to log files, you can obtain valuable information by retrieving the status of your Red Hat Satellite and its various components. This can be done with the command:

```
# /usr/sbin/rhn-satellite status
```

In addition, you can obtain the status of components such as the Apache Web server and the **Red Hat Network Task Engine** individually. For instance, to view the status of the Apache Web server, run the command:

```
# service httpd status
```

10.2. Installing and Updating

Q: SELinux keeps giving me messages when I'm trying to install. Why?

A: If you encounter any issues with SELinux messages (such as AVC denial messages) while installing Red Hat Satellite, be sure to have the `audit.log` files available so that Red Hat Support personnel can assist you. You can find the file in `/var/log/audit/audit.log` and can attach the file to your Support ticket for engineers to assist you.

Q: I changed /var/satellite to an NFS mount, and now SELinux is stopping it from working properly. What do I need to do?

A: SELinux parameters need to be changed based on the new NFS mount in order for SELinux to allow that traffic. Do this with the command:

```
# /usr/sbin/setsebool -P spacewalk_nfs_mountpoint on
```

If you are using Red Hat Enterprise Linux 6, you will also need to run the command:

```
# /usr/sbin/setsebool -P cobbler_use_nfs on
```

■

Q: My Satellite is failing. Any idea why?

A: Do not subscribe the Red Hat Satellite to any of the following child channels available from Red Hat Network's central servers:

Red Hat Developer Suite

Red Hat Application Server

Red Hat Extras

JBoss product channels

Subscribing to these channels and updating the Satellite might install newer, incompatible versions of critical software components, causing the Satellite to fail.

10.3. Services**Q: Why isn't the Apache Web server running?**

A: If the Apache Web server isn't running, entries in your `/etc/hosts` file may be incorrect.

Q: How do I find out what the status of the Red Hat Network Task Engine is?

A: To obtain the status of the **Red Hat Network Task Engine**, run the command:

```
# service taskomatic status
```

Q: How do I find out what the status of the Satellite's Embedded Database is?

A: To view the status of the Satellite's Embedded Database, if it exists, run the command:

```
# db-control status
```

Q: What do I do if the push capability of the Red Hat Satellite stops working?

A: If the push capability of the Red Hat Satellite ceases to function, it is possible that old log files may be at fault. Stop the jabberd daemon before removing these files. To do so, issue the following commands as root:

```
# service jabberd stop
# rm -f /var/lib/jabberd/db/_db*
# service jabberd start
```

10.4. Connectivity**Q: I can't connect! How do I work out what is wrong?**

A: The following measures can be used to troubleshoot general connection errors:

Attempt to connect to the Red Hat Satellite's database at the command line using the

```
# sqlplus username/password@sid
```

Make sure that Red Hat Satellite is using Network Time Protocol (NTP) and set to the appropriate time zone. This also applies to all client systems and the separate database machine in Red Hat Satellite with Stand-Alone Database.

Confirm the correct package:

```
rhn-org-httpd-ssl-key-pair-MACHINE_NAME-VER-REL.noarch.rpm
```

is installed on the Red Hat Satellite and the corresponding **rhn-org-trusted-ssl-cert-*.noarch.rpm** or raw CA SSL public (client) certificate is installed on all client systems.

Verify the client systems are configured to use the appropriate certificate.

If also using one or more Red Hat Satellite Proxy Servers, ensure each Proxy's SSL certificates are prepared correctly. The Proxy should have both its own server SSL key-pair and CA SSL public (client) certificate installed, since it will serve in both capacities. See the SSL Certificates chapter of the *Red Hat Satellite Client Configuration Guide* for specific instructions.

Make sure client systems are not using firewalls of their own, blocking required ports as identified in the *Red Hat Satellite Installation Guide's Additional Requirements* section.

Q: What do I do if importing or synchronizing a channel fails and I can't recover it?

A: If importing/synchronizing a channel fails and you can't recover it in any other way, run this command to delete the cache:

```
# rm -rf temporary-directory
```



NOTE

The *Red Hat Satellite Installation Guide* section on *Preparing for Import from Local Media* specifies `/var/rhn-sat-import/` as the temporary directory.

Next, restart the importation or synchronization.

Q: I'm getting "SSL_CONNECT" errors. What do I do now?

A: A common connection problem, indicated by **SSL_CONNECT** errors, is the result of a Satellite being installed on a machine whose time had been improperly set. During the Satellite installation process, SSL certificates are created with inaccurate times. If the Satellite's time is then corrected, the certificate start date and time may be set in the future, making it invalid.

To troubleshoot this, check the date and time on the clients and the Satellite with the following command:

```
# date
```

The results should be nearly identical for all machines and within the "notBefore" and "notAfter" validity windows of the certificates. Check the client certificate dates and times with the following command:

```
# openssl x509 -dates -noout -in /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
```

Check the Satellite server certificate dates and times with the following command:

```
# openssl x509 -dates -noout -in /etc/httpd/conf/ssl.crt/server.crt
```

By default, the server certificate has a one-year life while client certificates are good for 10 years. If you find the certificates are incorrect, you can either wait for the valid start time, if possible, or create new certificates, preferably with all system times set to GMT.

10.5. Logging and Reporting

Q: What are the different log files?

A: Virtually every troubleshooting step should start with a look at the associated log file or files. These provide invaluable information about the activity that has taken place on the device or within the application that can be used to monitor performance and ensure proper configuration. See [Table 10.1, “Log Files”](#) for the paths to all relevant log files:

There may be numbered log files (such as `/var/log/rhn/rhn_satellite_install.log.1`, `/var/log/rhn/rhn_satellite_install.log.2`, etc.) within the `/var/log/rhn/` directory. These are *rotated* logs, which are log files created with a `<NUMBER>` extension when the current `rhn_satellite_install.log` file fills up to a size as specified by the `logrotate(8)` daemon and the contents written to a rotated log file. For example, the `rhn_satellite_install.log.1` contains the oldest rotated log file, while `rhn_satellite_install.log.4` contains the most recently rotated log.

Table 10.1. Log Files

Component/Task	Log File Location
Apache Web server	<code>/var/log/httpd/</code> directory
Red Hat Satellite	<code>/var/log/rhn/</code> directory
Red Hat Satellite Installation Program	<code>/var/log/rhn/rhn_satellite_install.log</code>
Database installation - <i>Embedded Database</i>	<code>/var/log/rhn/install_db.log</code>
Database population	<code>/var/log/rhn/populate_db.log</code>

Component/Task	Log File Location
Red Hat Satellite Synchronization Tool	/var/log/rhn/rhn_server_satellite.log
Red Hat Network DB Control - Embedded Database	/var/log/rhn/rhn_database.log
Red Hat Network Task Engine (taskomatic)	/var/log/messages
yum	/var/log/yum.log
XML-RPC transactions	/var/log/rhn/rhn_server_xmlrpc.log

Q: How do I use `spacewalk-report`?

A: There are instances where administrators may need a concise, formatted summary of their Red Hat Satellite resources, whether it is to take inventory of their entitlements, subscribed systems, or users and organizations. Rather than gathering such information manually from the Satellite interface, Red Hat Satellite includes the **`spacewalk-report`** command to gather and display vital Satellite information at once.



NOTE

To use **`spacewalk-report`** you must have the **`spacewalk-reports`** package installed.

The logging feature of Satellite is added by default in fresh installations of Satellite version 5.6 and above. If the Satellite is upgraded from a version below 5.6 the logging feature will be turned on at the time of upgrade and from that point all events will be audited. This means that all users created before the upgrade will get logged from the time of upgrade. The past creation of a user and any past events will not appear in the log but all future events will be logged.

`spacewalk-report` allows administrators to organize and display reports about content, errata, systems, system event history, and user resources across the Satellite. The **`spacewalk-report`** command is used to generate reports on:

System Inventory - Lists all of the systems registered to the Satellite.

Entitlements - Lists all organizations on the Satellite, sorted by system or channel entitlements.

Errata - Lists all the errata relevant to the registered systems, sorts errata by severity as well as the systems that apply to a particular erratum.

Users - Lists all the users registered to the Satellite, and lists any systems associated with a particular user.

System History - Lists all, or a subset, of the system events that have occurred.

To get a report in CSV format, run the following at the command prompt of your Satellite server.

```
# spacewalk-report report_name
```

The following reports are available:

Table 10.2. spacewalk-report Reports

Report	Invoked as	Description
Group Audit	audit-server-groups	Audit of user changes in group
Server Audit	audit-servers	Audit of server changes
User Audit	audit-users	Audit of user changes
Packages Report	channel-packages	Lists the packages, as well as the channels they are in
Channels	channels	Lists the channels available on the server
Cloned Channels	cloned-channels	Lists channels that have been cloned
Custom Information	custom-info	Displays any custom information about the system
Entitlements	entitlements	Lists all organizations on the Satellite with their system or channel entitlements
Errata in channels	errata-channels	Lists errata in channels
Errata Compliance	errata-list	Lists the details of errata out of compliance information
All Errata	errata-list-all	Complete list of all errata
Errata for systems	errata-systems	Lists applicable errata and any registered systems that are affected
Relationship Mapping	host-guests	Provides host-guest mapping details
Inactive Systems	inactive-systems	

Report	Invoked as	Description
System Inventory	inventory	List of systems registered to the server, together with hardware and software information
Kickstart Trees	kickstartable-trees	Lists trees able to be kickstarted
Package Update	packages-updates-all	List of all packages that can be updated
Newest Package Update	package-updates-newest	Lists the newest updates to packages
SCAP Scans	scap-scan scap-scan-results	Displays the results of an OpenSCAP xccdf evaluation
Splice Reporting	splice-export	Displays system data needed for splice integration for enhanced reporting
Crash Count	system-crash-count	Displays the number of times systems have crashed
Crash Details	system-crash-details	Lists the systems' crash details
System Currency	system-currency	Lists system currency values
System Groups	system-groups	Lists system groups in the Satellite server
Group Activation keys	system-groups-keys	Lists all existing activation keys for the system groups
Systems in System Groups	system-groups-systems	Lists all system groups and systems within each group
Users in System Groups	system-groups-users	Lists all system groups and their affiliated users

Report	Invoked as	Description
System history	system-history	Lists system event history
System history channels	system-history-channels	Lists system event history
System history configuration	system-history-configuration	Lists system configuration event history
System history entitlements	system-history-entitlements	Lists system entitlement event history
System history errata	system-history-errata	Lists system errata event history
System history kickstart	system-history-kickstart	Lists system kickstart and provisioning event history
System history packages	system-history-packages	Lists system package event history
SCAP Event History	system-history-scap	Lists systems' OpenSCAP event history
Installed Packages	system-packages-installed	Lists all packages installed on the systems
Users in the system	users	Lists all users registered to the Satellite
Systems administered	users-systems	Lists systems that can be administered by individual users

For more information about an individual report, run **spacewalk-report** with the **--info** or **--list-fields-info** and the report name. The description and list of possible fields in the report will be shown.

For further information, the **spacewalk-report(8)** manpage as well as the **--help** parameter of the **spacewalk-report** program can be used to get additional information about the program invocations and their options.

Q: How do I work out what version of the database schema I have?

A: To determine the version of your database schema, run the command:

```
# rhn-schema-version
```

Q: How do I work out what character set types I have?

A: To derive the character set types of your Satellite's database, run the command:

```
# rhn-charsets
```

Q: Why isn't the administrator getting email?

A: If the administrator is not getting email from the Red Hat Satellite, confirm the correct email addresses have been set for `traceback_mail` in `/etc/rhn/rhn.conf`.

Q: How do I change the sender of the traceback mail?

A: If the traceback mail is marked from `dev-null@rhn.redhat.com` and you would like the address to be valid for your organization, include the `web.default_mail_from` option and appropriate value in `/etc/rhn/rhn.conf`.

10.6. Errors

Q: I'm getting an "Error validating satellite certificate" error during a Red Hat Satellite installation. How do I fix it?

A: An "Error validating satellite certificate" error message during a Red Hat Satellite installation is caused by having an HTTP proxy in the environment. This can be confirmed by looking at the `install.log` file, and locating the following error:

```
ERROR: unhandled exception occurred:
Traceback (most recent call last):
  File "/usr/bin/rhn-satellite-activate", line 45, in ?
    sys.exit(abs(mod.main() or 0))
  File "/usr/share/rhn/satellite_tools/rhn_satellite_activate.py",
line 585, in main
    activateSatellite_remote(options)
  File "/usr/share/rhn/satellite_tools/rhn_satellite_activate.py",
line 291, in activateSatellite_remote
    ret = s.satellite.deactivate_satellite(systemid, rhn_cert)
  File "/usr/lib/python2.4/site-packages/rhn/rpplib.py", line 603, in
__call__
    return self._send(self._name, args)
  File "/usr/lib/python2.4/site-packages/rhn/rpplib.py", line 326, in
_request
    self._handler, request, verbose=self._verbose)
  File "/usr/lib/python2.4/site-packages/rhn/transports.py", line 171,
in request
```

```

headers, fd = req.send_http(host, handler)
File "/usr/lib/python2.4/site-packages/rhn/transport.py", line 698,
in send_http
self._connection.connect()
File "/usr/lib/python2.4/site-packages/rhn/connections.py", line
193, in connect
sock.connect((self.host, self.port))
File "<string>", line 1, in connect
socket.timeout: timed out

```

To resolve the issue:

1. Run the install script in disconnected mode, and skip the database installation which has already been done:

```
# ./install.pl --disconnected --skip-db-install
```

2. Open `/etc/rhn/rhn.conf` with your preferred text editor, and add or modify the following line:

```
server.satellite.rhn_parent = satellite.rhn.redhat.com
```

Remove the following line:

```
disconnected=1
```

If you are using a proxy for the connection to Red Hat Network, you will also need to add or modify the following lines to reflect the proxy settings.

```
server.satellite.http_proxy = <hostname>:<port>
server.satellite.http_proxy_username = <username>
server.satellite.http_proxy_password = <password>
```

3. Re-activate Satellite in connected mode, using the **rhn-satellite-activate** command as the root user, including the path and filename of the Satellite manifest:

```
# rhn-satellite-activate --manifest=manifest.zip --verbose
```

Alternatively, try running the **install.pl** script in connected mode, but with the **--answer-file=*answer file*** option. Ensure the answer file has the HTTP proxy information specified as follows:

```
rhn-http-proxy = <hostname>:<port>
rhn-http-proxy-username = <username>
rhn-http-proxy-password = <password>
```

Q: I'm getting an "ERROR: server.mount_point not set in the configuration file" error when I try to activate or synchronize the Red Hat Satellite. How do I fix it?

A: An "ERROR: server.mount_point not set in the configuration file" error during Red Hat Satellite activation or synchronization can occur if the **mount_point** configuration parameter in `/etc/rhn/rhn.conf` does not point to a directory path, or the directory path it points to is not

present or does not have permission to access the directory.

To resolve the issue, check the value of the *mount_point* configuration parameter in `/etc/rhn/rhn.conf`. If it is set to the default value of `/var/satellite`, verify that the `/var/satellite` and `/var/satellite/redhat` directories exist. For all values, check that path to the file is accurate, and that the permissions are set correctly.

Q: Why does `cobbler check` give an error saying that it needs a different version of `yum-utils`?

A: Sometimes, running the `cobbler check` command can give an error similar to the following:

```
# cobbler check
The following potential problems were detected:
#0: yum-utils need to be at least version 1.1.17 for reposync -l,
current version is 1.1.16
```

This is a known issue in Cobbler's `reposync` package. The error is spurious and can be safely ignored. This error will be resolved in future versions of Red Hat Satellite.

Q: I'm getting an "unsupported version" error when I try to activate the Red Hat Satellite certificate. How do I fix it?

A: If your Red Hat Satellite certificate has become corrupted, you could get one of the following errors:

```
ERROR: <Fault -2: 'unhandled internal exception: unsupported version:
96'>
```

```
RHN_PARENT: satellite.rhn.redhat.com
Error reported from RHN: <Fault -2: 'unhandled internal
exception: unsupported version: 115'>
ERROR: unhandled XMLRPC fault upon remote activation: <Fault -2:
'unhandled internal exception: unsupported version: 115'>
ERROR: <Fault -2: 'unhandled internal exception: unsupported
version: 115'>
```

```
Invalid satellite certificate
```

To resolve this issue, contact Red Hat support services for a new certificate.

Q: I'm getting an "Internal Server Error" complaining about ASCII when I try to edit the kickstart profile. What's going on?

A: If you have recently added some kernel parameters to your kickstart profile, you might find that when you attempt to **View a List of Kickstart Profiles** that you get the following Internal Server Error:

```
'ascii' codec can't encode character u'\u2013'
```

This error occurs because some text in the profile is not being recognized correctly.

To resolve the issue:

1. Ssh directly onto the Satellite server as the root user:

```
# ssh root@satellite.fqdn.com
```

2. Find the kickstart profile that is causing the problem by looking at the dates of the files in `/var/lib/cobbler/config/profiles.d` and locating the one that was edited most recently:

```
# ls -l /var/lib/cobbler/config/profiles.d/
```

3. Open the profile in your preferred text editor, and locate the following text:

```
\u2013hostname
```

Change the entry to read:

```
--hostname
```

4. Save changes to the profile and close the file.
5. Restart the Red Hat Satellite services to pick up the updated profile:

```
# rhn-satellite restart
Shutting down rhn-satellite...
Stopping RHN Taskomatic...
Stopped RHN Taskomatic.
Stopping cobbler daemon: [
OK ]
Stopping rhn-search...
Stopped rhn-search.
Stopping MonitoringScout ... [
OK ]
Stopping Monitoring ... [
OK ]
Stopping httpd: [
OK ]
Stopping tomcat5: [
OK ]
Shutting down osa-dispatcher: [
OK ]
Shutting down Oracle Net Listener ... [
OK ]
Shutting down Oracle DB instance "rhnsat" ... [
OK ]
Shutting down Jabber router: [
OK ]
Done.
Starting rhn-satellite...
Starting Jabber services [
OK ]
Starting Oracle Net Listener ... [
OK ]
```

```

Starting Oracle DB instance "rhnsat" ... [
OK ]
Starting osa-dispatcher: [
OK ]
Starting tomcat5: [
OK ]
Starting httpd: [
OK ]
Starting Monitoring ... [
OK ]
Starting MonitoringScout ... [
OK ]
Starting rhn-search...
Starting cobbler daemon: [
OK ]
Starting RHN Taskomatic...
Done.

```

6. Return to the web interface. Note that the interface can take some time to resolve the services. It should return to normal after some time.

Q: I'm getting "Host Not Found" or "Could Not Determine FQDN" errors. What do I do now?

A: Because Red Hat Network configuration files rely exclusively on fully qualified domain names (FQDNs), it is imperative that key applications are able to resolve the name of the Red Hat Satellite into an IP address. **Red Hat Update Agent**, **Red Hat Network Registration Client**, and the Apache Web server are particularly prone to this problem with the Red Hat Network applications issuing errors of "host not found" and the Web server stating "Could not determine the server's fully qualified domain name" upon failing to start.

This problem typically originates from the `/etc/hosts` file. You may confirm this by examining `/etc/nsswitch.conf`, which defines the methods and the order by which domain names are resolved. Usually, the `/etc/hosts` file is checked first, followed by Network Information Service (NIS) if used, followed by DNS. One of these has to succeed for the Apache Web server to start and the Red Hat Network client applications to work.

To resolve this problem, identify the contents of the `/etc/hosts` file. It may look like this:

```

127.0.0.1 this_machine.example.com this_machine localhost.localdomain
\ localhost

```

First, in a text editor, remove the offending machine information, like so:

```

127.0.0.1 localhost.localdomain.com localhost

```

Then, save the file and attempt to re-run the Red Hat Network client applications or the Apache Web server. If they still fail, explicitly identify the IP address of the Satellite in the file, such as:

```

127.0.0.1 localhost.localdomain.com localhost
123.45.67.8 this_machine.example.com this_machine

```

Replace the value here with the actual IP address of the Satellite. This should resolve the problem. Keep in mind, if the specific IP address is stipulated, the file will need to be updated when the machine obtains a new address.

Q: I'm getting a "This server is not an entitled Satellite" when I try to synchronize the Red Hat Satellite server. How do I fix it?

A: If `cdn-sync` reports that the server is not activated as a Red Hat Satellite, it isn't subscribed to the respective Red Hat Satellite channel. If this is a newly installed system, make sure that the satellite manifest is activated on the system. If it was activated earlier, then it has become deactivated.

Check the system's child channels to discover if it is subscribed to any Red Hat Network Red Hat Satellite channel. View subscribed channels with the following command:

```
# yum repolist
```

Activate the same Satellite manifest again on your Satellite using this command as the root user:

```
# rhn-satellite-activate --manifest=manifest.zip --verbose
```

10.7. Web Interface

Q: I'm having problems with the Red Hat Satellite user interface. Which log files should I check?

A: If you experience errors viewing, scheduling, or working with kickstarts in the Red Hat Satellite user interface, check the `/var/log/tomcat6/catalina.out` log file.

For all other user interface errors, check the `/var/log/httpd/error_log` log file.

10.8. Anaconda

Q: I'm getting an error that says `Error downloading kickstart file`. What is the problem and how do I fix it?

A: This error is usually the result of a network issue. To locate the problem, run the `cobbler check` command, and read the output, which should look something like this:

```
# cobbler check
The following potential problems were detected:
#0: reposync is not installed, need for cobbler reposync,
install/upgrade yum-utils?
#1: yumdownloader is not installed, needed for cobbler repo add with -
-rpm-list parameter, install/upgrade yum-utils?
#2: The default password used by the sample templates for newly
installed machines (default_password_crypted in /etc/cobbler/settings)
is still set to 'cobbler' and should be changed
#3: fencing tools were not found, and are required to use the
(optional) power management features. install cman to use them
```

If `cobbler check` does not provide any answers, check the following:

Verify `httpd` is running: `service httpd status`

Verify `cobblerd` is running: `service cobblerd status`

Verify that you can fetch the kickstart file using **wget** from a different host:

```
wget http://satellite.example.com/cblr/svc/op/ks/profile/rhel5-
i386-u3:1:Example-Org
```

Q: I'm getting a package installation error that says The file `chkconfig-1.3.30.1-2.i386.rpm` cannot be opened. What is the problem and how do I fix it?

A: Clients will fetch content from Red Hat Satellite based on the `--url` parameter in the kickstart. For example:

```
url --url http://satellite.example.com/ks/dist/ks-rhel-i386-server-5-
u3
```

If you receive errors from Anaconda stating it can't find images or packages, check that the URL in the kickstart will generate a **200 OK** response. You can do this by attempting to **wget** the file located at that URL:

```
wget http://satellite.example.com/ks/dist/ks-rhel-i386-server-5-u3
--2011-08-19 15:06:55-- http://satellite.example.com/ks/dist/ks-rhel-
i386-server-5-u3
Resolving satellite.example.com... 10.10.77.131
Connecting to satellite.example.com|10.10.77.131|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/plain]
Saving to: `ks-rhel-i386-server-5-u3.1'
2011-08-19 15:06:55 (0.00 B/s) - `ks-rhel-i386-server-5-u3.1' saved
[0/0]
```

If you get a response other than **200 OK**, check the error logs to find out what the problem is. You can also check the actual file Anaconda tried to download by searching the **access_log** file:

```
# grep chkconfig /var/log/httpd/access_log
10.10.77.131 - - [19/Aug/2011:15:12:36 -0400] "GET
/rhn/common/DownloadFile.do?url=/ks/dist/ks-rhel-i386-server-
5-u3/Server /chkconfig-1.3.30.1-2.i386.rpm HTTP/1.1" 206 24744 "-"
"urlgrabber/3.1.0 yum/3.2.19"
10.10.76.143 - - [19/Aug/2011:15:12:36 -0400] "GET /ks/dist/ks-rhel-
i386-server-5-u3/Server/chkconfig-
1.3.30.1-2.i386.rpm HTTP/1.1" 206 24744 "-" "urlgrabber/3.1.0
yum/3.2.19"
10.10.76.143 - - [19/Aug/2011:15:14:20 -0400] "GET /ks/dist/ks-rhel-
i386-server-5-u3/Server/chkconfig-
1.3.30.1-2.i386.rpm HTTP/1.1" 200 162580 "-" "urlgrabber/3.1.0
yum/3.2.19"
10.10.77.131 - - [19/Aug/2011:15:14:20 -0400] "GET
/rhn/common/DownloadFile.do?url=/ks/dist/ks-rhel-i386-server-
5-u3/Server/chkconfig-1.3.30.1-2.i386.rpm HTTP/1.1" 200 162580 "-"
"urlgrabber/3.1.0 yum/3.2.19"
```

If the requests are not appearing in the **access_log** file, the system might be having trouble with the networking setup. If the requests are appearing but are generating errors, check the error logs.

You can also try manually downloading the files to see if the package is available:

```
wget http://satellite.example.com/ks/dist/ks-rhel-i386-server-5-
u3/Server/chkconfig-1.3.30.1-2.i386.rpm
```

10.9. Tracebacks

Q: I'm getting emails with "WEB TRACEBACK" in the subject. What should I do about them?

A: A typical traceback email might look something like this:

```
Subject: WEB TRACEBACK from satellite.example.com
Date: Wed, 19 Aug 2011 20:28:01 -0400
From: Red Hat Satellite <dev-null@redhat.com>
To: admin@example.com

java.lang.RuntimeException: XmlRpcException calling cobbler.
  at
  com.redhat.rhn.manager.kickstart.cobbler.CobblerXMLRPCHelper.invokeMet
  hod(CobblerXMLRPCHelper.java:72)
  at
  com.redhat.rhn.taskomatic.task.CobblerSyncTask.execute(CobblerSyncTask
  .java:76)
  at
  com.redhat.rhn.taskomatic.task.SingleThreadedTestableTask.execute(Sing
  leThreadedTestableTask.java:54)
  at org.quartz.core.JobRunShell.run(JobRunShell.java:203)
  at
  org.quartz.simpl.SimpleThreadPool$WorkerThread.run(SimpleThreadPool.ja
  va:520)
Caused by: redstone.xmlrpc.XmlRpcException: The response could not be
  parsed.
  at redstone.xmlrpc.XmlRpcClient.handleResponse(XmlRpcClient.java:434)
  at redstone.xmlrpc.XmlRpcClient.endCall(XmlRpcClient.java:376)
  at redstone.xmlrpc.XmlRpcClient.invoke(XmlRpcClient.java:165)
  at
  com.redhat.rhn.manager.kickstart.cobbler.CobblerXMLRPCHelper.invokeMet
  hod(CobblerXMLRPCHelper.java:69)
  ... 4 more
Caused by: java.io.IOException: Server returned HTTP response code:
  503 for URL: http://someserver.example.com:80/cobbler_api
  at
  sun.net.www.protocol.http.HttpURLConnection.getInputStream(HttpURLConn
  ection.java:1236)
  at redstone.xmlrpc.XmlRpcClient.handleResponse(XmlRpcClient.java:420)
  ... 7 more
```

This indicates that there has been a problem with Cobbler communicating with the **taskomatic** service. Try checking the following:

Verify **httpd** is running: # **service httpd status**

Verify **cobblerd** is running: # **service cobblerd status**

Verify that there are no firewall rules that would prevent **localhost** connections

10.10. Registration

Q: The `rhnreg_ks` command is failing when I run it, saying **ERROR: unable to read system id**. What is the problem?

A: At the end of the kickstart file, there is a **%post** section that registers the machine to the Red Hat Satellite:

```
# begin Red Hat management server registration
mkdir -p /usr/share/rhn/
wget http://satellite.example.com/pub/RHN-ORG-TRUSTED-SSL-CERT -O
/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
perl -npe 's/RHNS-CA-CERT/RHN-ORG-TRUSTED-SSL-CERT/g' -i
/etc/sysconfig/rhn/*
rhnreg_ks --serverUrl=https://satellite.example.com/XMLRPC --
sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT --activationkey=1-
c8d01e2f23c6bbaedd0f6507e9ac079d
# end Red Hat management server registration
```

Interpreting this in the order it was added in, this will:

Create a directory to house the custom SSL cert used by the Red Hat Satellite.

Fetch the SSL certificate to use during registration.

Search and replace the SSL certificate strings from the **rhn_register** configuration files, and then register to the Red Hat Satellite using the SSL certificate and an activation key. Every kickstart profile includes an activation key that assures that the system is assigned the correct base and child channels, and gets the correct system entitlements. If it is a reprovisioning of an existing system, the activation key will also ensure it is associated with the previous system profile.

If the **rhnreg_ks** command fails, you might see errors like this in the **ks-post.log** log file:

```
ERROR: unable to read system id.
```

These errors will also occur if an attempt is made to perform an **rhn_check** and the system has not registered to the Red Hat Satellite.

The best way to troubleshoot this is to view the kickstart file and copy and paste the four steps directly at the command prompt after the kickstart has completed. This will produce error messages that are more detailed to help locate the problem.

10.11. Kickstarts and Snippets

Q: What is the directory structure for kickstarts?

A: The base path where the kickstart files are stored is **/var/lib/rhn/kickstarts/**. Within this directory, raw kickstarts are in the **upload** subdirectory, and wizard-generated kickstarts are in the **wizard** subdirectory:

```
Raw Kickstarts: /var/lib/rhn/kickstarts/upload/$profile_name--
$org_id.cfg
Wizard Kickstarts: /var/lib/rhn/kickstarts/wizard/$profile_name--
$org_id.cfg
```

Q: What is the directory structure for Cobbler snippets?

A: Cobbler snippets are stored in `/var/lib/rhn/kickstarts/snippets`. Cobbler accesses snippets using the symbolic link `/var/lib/cobbler/snippets/spacewalk`.

```
Snippets: /var/lib/rhn/kickstarts/snippets/$org_id/$snippet_name
```



IMPORTANT

Red Hat Satellite RPMs expect the Cobbler kickstart and snippet directories to be in their default locations, do not change them.

10.12. Multi-Organization Satellites and Satellite Certificate

Q: How do I register my systems in a Multiple Organization environment when I do not have enough entitlements in my Satellite Certificate?

A: There are some situations in which you need to free entitlements and do not have a lot of time to do so, and may not have access to each organization in order to do this yourself. There is an option in Multi-Org Satellites that allows the Satellite administrator to reduce an organization's entitlement count below their usage. This method must be done logged into the administrative organization.

For example, logged into the administrative organization, if your certificate is 5 system management entitlements shy of being able to cover all registered systems on your Satellite, the 5 systems that were most recently registered to that organization will be unentitled. This process is described below:

1. In the `/etc/rhn/rhn.conf` file, set ***web.force_unentitlement*** to 1.
2. Restart the Satellite.
3. Reduce the allocated entitlements to the desired organizations either via each organization's **Subscriptions** tab or via individual entitlement's **Organizations** tabs.
4. A number of systems in the organization should now be in an **unentitled** state. The number of systems unentitled in the organization will be equal to the difference between the total number of entitlements you removed from the organization and the number of entitlements the organization did not have applied to the systems.

For example, if you removed 10 entitlements from the organization in step 3, and the organization has 4 entitlements that were not in use by systems, then 6 systems in the organization will be unentitled.

After you have the sufficient number of entitlements required, you should then be able to activate your new Satellite certificate. Note that modifying the ***web.force_unentitlement*** variable is only necessary to reduce an organization's allocated entitlements below what they are using. If an

organization has more entitlements than are being actively used, you do not need to set this variable to remove them.

Q: I have extra entitlements on my Satellite Certificate that are not being used. What happens to these entitlements?

A: If you are issued a new Satellite certificate and it has more entitlements than are being consumed on your Satellite, any extra entitlements will be assigned to the administrative organization. If you log into the web interface as the Satellite administrator, you will be able to allocate these entitlements to other organizations. The previously-allocated entitlements to other organizations will be unaffected.

10.13. Proxy Installation and Configuration

Q: After configuring the Red Hat Network Package Manager how can I determine if the local packages were successfully added to the private Red Hat Network channel?

A: Use the command `rhn_package_manager -l -c "name_of_private_channel"` to list the private channel packages known to the Satellite. Or visit the Satellite interface.

After subscribing a registered system to the private channel, you can also execute the command `yum --disablerepo="*" --enablerepo="your_repo_name" list available` on the registered system and look for the packages from the private Satellite channel.

Q: How can I determine whether the clients are connecting to the Squid server?

A: The `/var/log/squid/access.log` file logs all connections to the Squid server.

Q: The Red Hat Update Agent on the client systems does not connect through the Red Hat Satellite Proxy. How can I resolve this error?

A: Make sure that the latest version of the Red Hat Update Agent is installed on the client systems. The latest version contains features necessary to connect through a Red Hat Satellite Proxy. The latest version can be obtained through the Red Hat Network by issuing the command `yum update yum` as root or from <http://www.redhat.com/support/errata/>.

The Red Hat Satellite Proxy is an extension of Apache. See the *Log Files* section of the *Red Hat Satellite Proxy Installation Guide* for its log file location.

Q: My Red Hat Satellite Proxy configuration does not work. Where do I begin troubleshooting it?

A: Make sure `/etc/sysconfig/rhn/systemid` is owned by root.apache with the permissions 0640.

Read the log files. A list is available on the *Log Files* section of the *Red Hat Satellite Proxy Installation Guide*.

Q: How do I troubleshoot general problems in the Red Hat Satellite Proxy?

- A:** To begin troubleshooting general problems, examine the log file or files related to the component exhibiting failures.

A common issue is full disk space. An almost sure sign of this is the appearance of halted writing in the log files. If logging stops during a write, such as mid-word, you likely have filled disks. To confirm this, run this command and check the percentages in the Use% column:

```
df -h
```

In addition to log files, you can obtain valuable information by retrieving the status of your various components. This can be done for the Apache Web server and Squid.

To obtain the status of the Apache Web server, run the command:

```
service httpd status
```

To obtain the status of Squid, run the command:

```
service squid status
```

If the administrator is not getting email from the Red Hat Satellite Proxy, confirm the correct email addresses have been set for `traceback_mail` in `/etc/rhn/rhn.conf`.

Q: My Red Hat Satellite Proxy encountered the error "Host Not Found"/"Could not Determine FQDN". What should I do?

- A:** Because Red Hat Network configuration files rely exclusively on fully qualified domain names (FQDN), it is imperative that key applications are able to resolve the name of the Red Hat Satellite Proxy into an IP address. Red Hat Update Agent, Red Hat Network Registration Client, and the Apache Web server are particularly prone to this problem with the Red Hat Network applications issuing errors of "host not found" and the Web server stating "Could not determine the server's fully qualified domain name" upon failing to start.

This problem originates from the `/etc/hosts` file. Confirm this by examining the `/etc/nsswitch.conf` file, which defines the methods and the order by which domain names are resolved. Usually, the `/etc/hosts` file is checked first, followed by Network Information Service (NIS) if it is being used, followed by DNS. One of these has to succeed for the Apache Web server to start and the Red Hat Network client applications to work.

To resolve this problem, identify the contents of the `/etc/hosts` file. It may look like this:

```
127.0.0.1 this_machine.example.com this_machine localhost.localdomain
\ localhost
```

In a text editor, remove the machine host information from the file, it should look like so:

```
127.0.0.1 localhost.localdomain.com localhost
```

Save the file and attempt to re-run the Red Hat Network client applications or the Apache Web server. If they still fail, explicitly identify the IP address of the Proxy in the file, such as:

```
127.0.0.1 localhost.localdomain.com localhost
123.45.67.8 this_machine.example.com this_machine
```

Replace the value here with the actual IP address of the Proxy. This should resolve the problem. Keep in mind, if the specific IP address is stipulated, the file will need to be updated when the machine obtains a new address.

Q: I am having issues with Red Hat Satellite Proxy and network connection errors. What should I do?

A: If you are experiencing problems that you believe to be related to failed connections, follow these measures:

Confirm the correct package:

```
rhncertutil-MACHINE_NAME-VER-REL.noarch.rpm
```

is installed on the Red Hat Satellite Proxy and the corresponding **rhncertutil-ssl-cert-*.noarch.rpm** or raw CA SSL public (client) certificate is installed on all client systems.

Verify the client systems are configured to use the appropriate certificate.

If using one or more Red Hat Satellite Proxies, ensure each Proxy's SSL certificate is prepared correctly. If using the Red Hat Satellite Proxy in conjunction with a Red Hat Satellite, the Proxy should have both its own server SSL key-pair and CA SSL public (client) certificate installed, since it will serve in both capacities. See the SSL Certificates chapter of the *Red Hat Satellite Client Configuration Guide* for specific instructions.

If the Red Hat Satellite Proxy is connecting through an HTTP Proxy, make sure the URL listed is valid. For instance, the HTTP Proxy URL field should not contain references to protocols, such as `http://` or `https://`. Only the hostname and port should be included in the form `hostname:port`, such as **your-gateway.example.com:8080**.

Make sure client systems are not using firewalls of their own, blocking required ports, as identified in the *Additional Requirements* section of the *Red Hat Satellite Proxy Installation Guide*.

Q: I am having issues with package delivery errors and object corruption. What should I check for?

A: If package delivery fails or an object appears to be corrupt, and it is not related to connection errors, you should consider clearing the caches. The Red Hat Satellite Proxy has two caches you should be concerned with: one for Squid and the other for authentication.

The Squid cache is located in `/var/spool/squid/`. To clear it:

1. Stop the Apache Web server: **service httpd stop**
2. Stop the Squid server: **service squid stop**
3. Delete the contents of that directory: **rm -fv /var/spool/squid/***
4. Restart both services:

```
service squid start
service httpd start
```

The same task can be accomplished quicker by just clearing the directory and restarting squid, but this method will most likely result in a number of Red Hat Network traceback messages.

The internal caching mechanism used for authentication by the Proxy may also need its cache cleared. To do this, issue the following command:

```
rm -fv /var/spool/squid/*
```

NOTE

If you have exhausted these troubleshooting steps or want to defer them to Red Hat Network professionals, Red Hat recommends you take advantage of the strong support that comes with Red Hat Satellite. The most efficient way to do this is to aggregate your Satellite's configuration parameters, log files, and database information and send this package directly to Red Hat.

Red Hat Network provides a command line tool explicitly for this purpose: The **Satellite Diagnostic Info Gatherer**, commonly known by its command **satellite-debug**. To use this tool, issue the command as root. You will see the pieces of information collected and the single tarball created, like so:

```
# satellite-debug
Collecting and packaging relevant diagnostic information.
Warning: this may take some time...
  * copying configuration information
  * copying logs
  * querying RPM database (versioning of Red Hat Satellite,
etc.)
  * querying schema version and database character sets
  * get diskspace available
  * timestamping
  * creating tarball (may take some time): /tmp/satellite-
debug.tar.bz2
  * removing temporary debug tree
```

Debug dump created, stored in /tmp/satellite-debug.tar.bz2
Deliver the generated tarball to your Red Hat Network contact or support channel.

Once finished, email the new file from the **/tmp/** directory to your Red Hat representative for immediate diagnosis.

Additionally, Red Hat provides a command line tool called the **SoS Report**, commonly known by its command **sosreport**. This tool collects your Proxy's configuration parameters, log files, and database information and sends it directly to Red Hat.

To use this tool for Red Hat Satellite information, you must have the **sos** package installed. Type **sosreport -o satellite** as root on the Satellite server to create a report. For example:

```
[root@satserver ~]# sosreport -o satellite
```



sosreport (version 3.2)

This command will collect diagnostic and configuration information from this Red Hat Enterprise Linux system and installed applications.

An archive containing the collected information will be generated in /tmp and may be provided to a Red Hat support representative.

Any information provided to Red Hat will be treated in accordance with the published support policies at:

<https://access.redhat.com/support/>

The generated archive may contain data considered sensitive and its content should be reviewed by the originating organization before being passed to any third party.

No changes will be made to system configuration.

Press ENTER to continue, or CTRL-C to quit.

You are then prompted for your first initial and last name, then a support case number.

It may take several minutes for the system to generate and archive the report to a compressed file. Once finished, email the new file from the /tmp/ directory to your Red Hat representative for immediate diagnosis.

APPENDIX A. REVISION HISTORY

Revision 1.1-0

Wed Feb 1 2017

Satellite Documentation Team

Initial revision for the Red Hat Satellite 5.8 release.