Red Hat Satellite 5.7

User Guide

Using and administering Red Hat Satellite

Last Updated: 2017-09-26

Red Hat Satellite 5.7 User Guide

Using and administering Red Hat Satellite

Red Hat Satellite Documentation Team

Legal Notice

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

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.

Monitoring Administrator

Performs scheduling of probes and oversight of other monitoring infrastructure. This role is available only on Satellite Servers with monitoring enabled.

`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 **`rhn.conf`** 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 **`rhn.conf`**.

   - 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 Red Hat Network 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' && satellite-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 **satellite-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:

- **`/opt/rh/postgresql92/root/var/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 **satellite-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 **satellite-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 are 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=postgresql92-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.

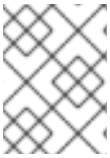
To examine the backup, run the following command as root:

```
# 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:

   - **`/etc/rhn/rhn.conf`**
   - **`/etc/tnsnames.ora`** (Oracle database only.)

5. 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.

6. - 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.

7. 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.

8. 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.

9. 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
```

1. 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:

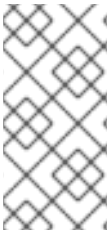
```
0 3 * * * backup-db.sh
0 6 * * * move-files.sh
```

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** (**`rhn-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. `rhn-actions-control` options

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

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. **`rhn-actions-control --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
--topdir=TOPDIR	Make all file operations relative to this string.
--exclude=EXCLUDE	Excludes a file from being deployed with 'get'/ May be used multiple times.
-h, --help	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
-v, --verbose	Increase the amount of output detail. Displays differences in the mode, owner, and group permissions for the specified config file.
-o, --only	Only show files that differ.
-h, --help	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
+
+iD8DBQA9ZY6vse4XmfJPGwgRAsHcAJ9ud9dabUcdscdcqB8AZP7e0Fua0NmKsdhQCeOWHX
+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
-c CHANNEL, --channel=CHANNEL	Use this config channel
-r REVISION, --revision=REVISION	Use this revision
-h, --help	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-channel3': /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, in 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 decomissioning 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. Red Hat Enterprise Linux 4 and lower versions use the **up2date** command to install and update packages and does not provide the metadata **spacewalk-clone-by-date** requires.

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 --assumeyes
```

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

CHAPTER 6. MONITORING

The Red Hat Network Monitoring entitlement allows you to perform a whole host of actions designed to keep your systems running properly and efficiently. With it, you can keep close watch on system resources, network services, databases, and both standard and custom applications.

Monitoring provides both real time and historical state change information, as well as specific metric data. It provides notifications of system failures and performance degradation before it becomes critical. It also provides information that assists in capacity planning and event correlation. For example, the results of a probe recording CPU usage across systems would assist in balancing loads on those systems.

There are two components to the monitoring system: the monitoring system and the *monitoring scout*. The monitoring system is installed in the Satellite and performs backend functions such as storing monitoring data and acting on it. The monitoring scout runs all the probes and collects monitoring data. The monitoring scout can be enabled to run on a Satellite or Red Hat Satellite Proxy system. Using monitoring scout on Proxy allows you to offload work from the Satellite, providing scalability for probes.

Monitoring entails establishing notification methods, installing probes on systems, regularly reviewing the status of all probes, and generating reports displaying historical data for a system or service. This section seeks to identify common tasks associated with the Monitoring entitlement. Remember, virtually all changes affecting your Monitoring infrastructure must be finalized by updating your configuration, through the **`Scout Config Push`** page.

6.1. PREREQUISITES

Before attempting to implement Red Hat Network Monitoring within your infrastructure, ensure you have all of the necessary tools in place. At a minimum, you need:

- Monitoring entitlements - These entitlements are required for all systems that are to be monitored. Monitoring is only supported on Red Hat Enterprise Linux systems.
- Red Hat Satellite with monitoring - monitoring systems must be connected to a Satellite with a base operating system of Red Hat Enterprise Linux 5 or later.
- Monitoring Administrator - This role must be granted to users installing probes, creating notification methods, or altering the monitoring infrastructure in any way. (Remember, the Satellite Administrator automatically inherits the abilities of all other roles within an organization and can therefore conduct these tasks.). Assign this role through the **`User Details`** page for the user.
- Red Hat Network monitoring daemon - This daemon, along with the SSH key for the scout, is required on systems that are monitored in order for the internal process monitors to be executed. You may, however, be able to run these probes using the systems' existing SSH daemon (**`sshd`**). See Section 6.2, “Configuring the Red Hat Network Monitoring Daemon (rhnmd)” for the complete list of available probes.

Enabling Monitoring

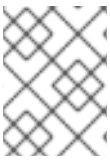
Monitoring is disabled by default, and will need to be enabled before it can be used.

1. Log in as a user with Satellite Administrator privileges and navigate to **Admin → Red Hat Satellite Configuration**. Click the **`Enable Monitoring`** checkbox, then click **`Update`** to save.

2. Restart services to pick up the changes. Go to the **restart** tab to restart the Satellite. This will take the Satellite offline for a few minutes.

3. Check if the **Monitoring** tab is available under **Red Hat Satellite Configuration** to confirm that monitoring is enabled.

4. Navigate to **Admin** → **Red Hat Satellite Configuration** → **Monitoring**. Click the **`Enable Monitoring Scout`** checkbox to enable the scout. Click **`Update Config`** to save.



NOTE

It is recommended that you leave the monitoring configuration values as the default values. **Sendmail** needs to be configured to use notifications.

6.2. CONFIGURING THE RED HAT NETWORK MONITORING DAEMON (RHNMD)

To make the most out of your monitoring entitlement, Red Hat suggests installing the Red Hat Network monitoring daemon on your client systems. Based upon **OpenSSH**, **`rhnmd`** enables the Satellite to communicate securely with the client system to access internal processes and retrieve probe status.

Please note that the Red Hat Network monitoring daemon requires that monitored systems allow connections on port 4545. You may avoid opening this port and installing the daemon altogether by using **`sshd`** instead. See Section 6.2.2, “Configuring SSH” for details.

Some probes require the daemon. An encrypted connection, either through the Red Hat Network monitoring daemon or **`sshd`**, is required on client systems for the following probes to run:

- Linux::CPU Usage
- Linux::Disk IO Throughput
- Linux::Disk Usage
- Linux::Inodes
- Linux::Interface Traffic
- Linux::Load
- Linux::Memory Usage
- Linux::Process Counts by State
- Linux::Process Count Total
- Linux::Process Health
- Linux::Process Running
- Linux::Swap Usage
- Linux::TCP Connections by State
- Linux::Users

- Linux::Virtual Memory
- LogAgent::Log Pattern Match
- LogAgent::Log Size
- Network Services::Remote Ping
- Oracle::Client Connectivity
- General::Remote Program
- General::Remote Program with Data

Note that all probes in the Linux group have this requirement.

6.2.1. Installing the Red Hat Network Monitoring Daemon

Install the Red Hat Network monitoring daemon to prepare systems for monitoring with the probes identified by **rhnmd**. Note that the steps in this section are optional if you intend to use **sshd** to allow secure connections between the Red Hat Network monitoring infrastructure and the monitored systems. See Section 6.2.2, “Configuring SSH” for instructions.

The **rhnmd** package can be found in the Red Hat Network Tools channel for all Red Hat Enterprise Linux distributions. To install it:

1. Subscribe the systems to be monitored to the Red Hat Network Tools channel associated with the system. This can be done individually through the **System Details** → **Channels** → **Software** subtab or for multiple systems at once through the **Channel Details** → **Target Systems** tab.
2. Once subscribed, open the **Channel Details** → **Packages** tab and find the **rhnmd** package (under 'R').
3. Click the package name to open the **Package Details** page. Go to the **Target Systems** tab, select the desired systems, and click **Install Packages**.
4. Install the SSH public key on all client systems to be monitored, as described in Section 6.2.3, “Installing the SSH key”.
5. Start the Red Hat Network monitoring daemon on all client systems using the command:

```
service rhnmd start
```

6. When adding probes requiring the daemon, accept the default values for **RHNMD User** and **RHNMD Port**: **nocpulse** and **4545**, respectively.

6.2.2. Configuring SSH

If you wish to avoid installing the Red Hat Network monitoring daemon and opening port 4545 on client systems, you may configure **sshd** to provide the encrypted connection required between the systems and Red Hat Network. This may be especially desirable if you already have **sshd** running. To configure the daemon for monitoring use:

1. Ensure the SSH package is installed on the systems to be monitored:

```
rpm -qi openssh-server
```

2. Identify the user to be associated with the daemon. This can be any user available on the system, as long as the required SSH key can be put in the user's **~/.ssh/authorized_keys** file.

3. Identify the port used by the daemon, as identified in its **/etc/ssh/sshd_config** configuration file. The default is port 22.

4. Install the SSH public key on all client systems to be monitored, as described in Section 6.2.3, “Installing the SSH key”.

5. Start the **sshd** on all client systems using the command:

```
service sshd start
```

6. When adding probes requiring the daemon, insert the values derived from steps 2 and 3 in the **RHNMD User** and **RHNMD Port** fields.

6.2.3. Installing the SSH key

Whether you use **rhnmd** or **sshd**, you must install the Red Hat Network monitoring daemon public SSH key on the systems to be monitored to complete the secure connection. To install it:

1. Navigate to the **Monitoring** → **Scout Config Push** page on the Satellite interface and click the name of the scout that will monitor the client system. The SSH **id_dsa.pub** key is visible on the resulting page.

2. Copy the character string (beginning with ssh-dss and ending with the hostname of the Satellite).

3. Select **Systems** from the left menu, and click the checkbox next to the systems you want to send the SSH key to. Click the **Manage** button at the top to finish.

4. From the **System Set Manager**, click **Run remote commands**, then in the **Script** text box, type the following line:

```
#!/bin/sh
cat <<EOF >> ~nocpulse/.ssh/authorized_keys
```

Then, press **Enter**, paste the SSH Key and add EOF. The result should look similar to the following:

```
#!/bin/sh
cat <<EOF>> ~nocpulse/.ssh/authorized_keys
ssh-dss AABBAB3NzaC3kc3MABCCBAJ4cmyf5jt/ihdtFbNE1YHsT0np0SYJz7xk
hzoKUUWnZmOUqJ7eXoTbGEcZjZLppOZgzAepw1vUHXfa/L9XiXvsV8K5Qmcu70h0
1gohBIder/1I1QbHMCgfDVFPtfV5eedau4AAACAc99dHbWhk/dMPiWXgHxdI0vT2
SnuozIox2klmfbTeO4Ajn/Ecfxqgs5diat/NIaeoItuGUYepXFoVv8DVL3wpp45E
02hjmp4j2MYNpc6Pc3nPOVntu6YBv+whB0VrsVzeqX89u23FFjTLGbfYrmMQflNi
j8yynGRePIMFhI= root@satellite.example.com
EOF
```

5. Set the date and time you want for the action to take place, then click **Schedule Remote Command**.

Once the key is in place and accessible, all probes that require it should allow **ssh** connections between the monitoring infrastructure and the monitored system. You may then schedule probes requiring the monitoring daemon to run against the newly configured systems.

6.3. CONFIGURING THE MYSQL PACKAGE FOR PROBES

If your Red Hat Satellite will serve monitoring-entitled client systems against which you wish to run **MySQL** probes, you must configure the **mysql** package on the Red Hat Satellite. See Appendix A, *Probes* for a listing of all available probes.

Subscribe the Satellite to the Red Hat Enterprise Linux Base channel and install the **mysql** package with either the **yum** (Red Hat Enterprise Linux 5, 6, and 7) command or the **up2date** command (Red Hat Enterprise Linux 3 and 4).

Once finished, your Satellite may be used to schedule MySQL probes.

6.4. ENABLING NOTIFICATIONS

In addition to viewing probe status within the Red Hat Network interface, you may be notified whenever a probe changes state. This is especially important when monitoring mission-critical production systems. For this reason, Red Hat recommends taking advantage of this feature.

To enable probe notifications within Red Hat Network, you must have identified a mail exchange server and mail domain during installation of your Red Hat Satellite and configured **sendmail** to properly handle incoming mail. See the *Installation* section of the *Red Hat Satellite Installation Guide* for details.

6.4.1. Creating Notification Methods

Notifications are sent via a *notification method*, an email or pager address associated with a specific Red Hat Network user. Although the address is tied to a particular user account, it may serve multiple administrators through an alias or mailing list. Each user account can contain multiple notification methods. To create a notification method:

1. Log into the Satellite as either a Satellite Administrator or Monitoring Administrator.

2. Navigate to **Users** and select the username. On the **User Details** page, click on **Notification Methods → create new method**.

3. Enter an intuitive, descriptive label for the method name, such as **DBA day email**, and provide the correct email address. Remember, the labels for all notification methods are available in a single list during probe creation, so they should be unique to your organization.

4. Select the checkbox if you desire abbreviated messages to be sent to the email address. This shorter format contains only the probe state, system hostname, probe name, time of message, and Send ID. The standard, longer format displays additional message headers, system and probe details, and instructions for response.

5. When finished, click **Create Method**. The new method shows up in the **User Details → Notification Methods** tab and the **Notification** page under the top **Monitoring** category. Click its name to edit or delete it.

6. While adding probes, select the **`Probe Notifications`** checkbox and select the new notification method from the resulting dropdown menu. Notification methods assigned to probes cannot be deleted until they are dis-associated from the probe.

6.4.2. Receiving Notifications

If you create notification methods and associate them with probes, you must be prepared to receive them. These notifications come in the form of brief text messages sent to the specified email address. Here is an example of an email notification:

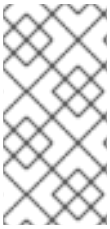
```
Subject: CRITICAL: [hostname]: Satellite: Users at 1
From: "Monitoring Satellite Notification" (rogerthat01@redhat.com)
Date: Mon, 26 Aug 2013 13:42:28 -0800
To: user@organization.com

This is Red Hat Monitoring Satellite notification 01dc8hqw.

Time: Mon Aug 26, 21:42:25 PST
State: CRITICAL
System: [hostname] ([IP address])
Probe: Satellite: Users
Message: Users 6 (above critical threshold of 2)
Notification #116 for Users

Run from: Red Hat Monitoring Satellite
```

As you can see, the longer email notifications contain virtually everything you would need to know about the associated probe. In addition to the probe command, run time, system monitored, and state, the message contains the *Send ID*, which is a unique character string representing the precise message and probe. In the above message, the Send ID is 01dc8hqw.



NOTE

Since notifications can be generated whenever a probe changes state, simple changes in your network can result in a flood of notifications. Notifications maybe redirected to a specific inbox meant for notifications to avoid issues with priority mail. The next section discusses redirecting notifications.

6.4.3. Redirecting Notifications

Upon receiving a notification, you may redirect it by including advanced notification rules within an acknowledgment email. Enable email reply redirects by opening **`/etc/aliases`** and adding the following line:

```
rogerthat01:    "| /etc/smrsh/ack_queuer.pl"
```

Once the parameter has been set, reply to the notification email and include the desired option. These are the possible redirect options, or *filter types*:

- ACK METOO - Sends the notification to the redirect destination(s) *in addition to* the default destination.
- ACK SUSPEND - Suspends the notification method for a specified time period.

- ACK AUTOACK - Does not change the destination of the notification, but automatically acknowledges matching alerts as soon as they are sent.
- ACK REDIR - Sends the notification to the redirect destination(s) *instead of* the default destination.

The format of the rule should be *filter_type probe_type duration email_address* where *filter_type* indicates one of the previous advanced commands, *probe_type* indicates **check** or **host**, *duration* indicates the length of time for the redirect, and *email_address* indicates the intended recipient. For example:

```
ACK METOO host 1h boss@domain.com
```

Capitalization is not required. Duration can be listed in minutes (m), hours (h), or days (d). Email addresses are needed only for redirects (REDIR) and supplemental (METOO) notifications.

The description of the action contained in the resulting email defaults to the command entered by the user. The reason listed is a summary of the action, such as email ack redirect by user@domain.com where user equals the sender of the email.

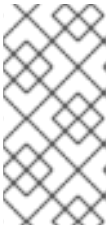
NOTE

You can halt or redirect almost all probe notifications by replying to a notification emails with a variation of the command **`ack suspend host`**. However, you cannot halt Satellite probe notifications by responding to a probe with **`ack suspend host`** or other redirect responses. These probes require you to change the notifications within the web interface of the Satellite.

6.4.4. Deleting Notification Methods

Existing relationships between methods and probes can complicate the process of deleting notification methods. Follow these steps to remove a notification method:

1. Log into the Satellite as a Satellite Administrator or Monitoring Administrator.
2. Navigate to the **Monitoring** → **Notifications** page and click the name of the method to be removed.
3. On the **User** → **User Details** → **Notification Methods** tab, click **`delete method`**. If the method is not associated with any probes, you are presented with a confirmation page. Click **`Confirm Deletion`**. The notification method is removed.



NOTE

Since both the notification method name and address can be edited, consider updating the method rather than deleting it. This redirects notifications from all probes using the method without having to edit each probe and create a new notification method.

4. If the method is associated with one or more probes, you are presented with a list of the probes using the method and the systems to which the probes are attached instead of a confirmation page. Click the probe name to go directly to the **System Details** → **Probes** tab.
5. Select another notification method and click **`Update Probe`**.

6. Return to the **Monitoring** → **Notifications** page and delete the notification method.

6.5. PROBES

Now that the Red Hat Network monitoring daemon has been installed and notification methods have been created, you may begin installing probes on your monitoring-entitled systems. If a system is entitled to monitoring, a **Probes** tab appears within its **System Details** page. This is where you will conduct most probe-related work.

6.5.1. Managing Probes

Probes are created through the Red Hat Satellite server. Once the probe has been created, the probes are propagated to the specified monitoring-entitled systems registered to the Satellite. Follow the steps below to add a probe in the Satellite server:

1. Log into the Satellite as either a Satellite Administrator or the System Group Administrator for the system.

2. Navigate to the **System Details** → **Probes** tab and click **create new probe**.

3. On the **System Probe Creation** page, complete all required fields. First, select the Probe Command Group. This alters the list of available probes and other fields and requirements. See Appendix A, *Probes* for the complete list of probes by command group. Remember that some probes require the Red Hat Network monitoring daemon to be installed on the client system.

4. Select the desired Probe Command and the monitoring Scout, typically **Red Hat Monitoring Satellite** but possibly a Red Hat Satellite Proxy Server. Enter a brief but unique description for the probe.

5. Select the **Probe Notifications** checkbox to receive notifications when the probe changes state. Use the **Probe Check Interval** dropdown menu to determine how often notifications should be sent. Selecting **1 minute** (and the **Probe Notification** checkbox) means you will receive notifications every minute the probe surpasses its CRITICAL or WARNING thresholds. See Section 6.4, “Enabling Notifications” to find out how to create notification methods and acknowledge their messages.

6. Use the **RHNMD User** and **RHNMD Port** fields, if they appear, to force the probe to communicate via **sshd**, rather than the Red Hat Network monitoring daemon. See Section 6.2.2, “Configuring SSH” for details. Otherwise, accept the default values of **nocpulse** and **4545**, respectively.

7. If the **Timeout** field appears, review the default value and adjust to meet your needs. Most but not all timeouts result in an UNKNOWN state. If the probe's metrics are time-based, ensure the timeout is not less than the time allotted to thresholds. Otherwise, the metrics serve no purpose, as the probe will time out before any thresholds are crossed.

8. Use the remaining fields to establish the probe's alert thresholds, if applicable. These CRITICAL and WARNING values determine at what point the probe has changed state. See Section 6.5.2, “Establishing Thresholds” for best practices regarding these thresholds.

9. When finished, click **Create Probe**. Remember, you must commit your monitoring configuration change on the **Scout Config Push** page for this to take effect.

To delete a probe, navigate to its **Current State** page (by clicking the name of the probe from the **System Details** → **Probes** tab), and click **delete probe**. Finally, confirm the deletion.

6.5.2. Establishing Thresholds

Many of the probes offered by Red Hat Satellite contain alert thresholds that, when crossed, indicate a change in state for the probe. For instance, the Linux::CPU Usage probe allows you to set CRITICAL and WARNING thresholds for the percent of CPU used. If the monitored system reports 75 percent of its CPU used, and the WARNING threshold is set to 70 percent, the probe will go into a WARNING state. Some probes offer a multitude of such thresholds.

In order to get the most out of your monitoring entitlement and avoid false notifications, Red Hat recommends running your probes without notifications for a time to establish baseline performance for each of your systems. Although the default values provided for probes may suit you, every organization has a different environment that may require altering thresholds.

6.5.3. Monitoring the Satellite Server

In addition to monitoring all of your client systems, you may also use Red Hat Network to monitor your Satellite or Proxy. To monitor your Satellite or Proxy, find a system monitored by the server, and go to that system's **System Details** → **Probes** tab.

Click **`create new probe`** and select the **`Satellite`** Probe Command Group. Next, complete the remaining fields as you would for any other probe. See Section 6.5.1, “Managing Probes” for instructions.

Although the Satellite or Proxy appears to be monitored by the client system, the probe is actually run from the server on itself. Thresholds and notifications work normally.

NOTE

Any probes that require Red Hat Network monitoring daemon connections cannot be used against a Red Hat Satellite or Red Hat Satellite Proxy Server on which monitoring software is running. This includes most probes in the Linux command group as well as the Log Agent probes and the Remote Program probes. Use the Satellite command group probes to monitor Red Hat Satellites and Red Hat Satellite Proxy Servers. In the case of Proxy scouts, the probes are listed under the system for which they are reporting data.

6.6. MONITORING

If you click the **`Monitoring`** tab on the top navigation bar, the **`Monitoring`** category and links appear. These pages, which require Monitoring entitlements, enable you to view the results of probes you have set to run against Monitoring-entitled systems and manage the configuration of your monitoring infrastructure.

Initiate the monitoring of a system through the **`Probes`** tab of the **`System Details`** page. See Appendix A, *Probes* for the complete list of available probes.

6.6.1. Probe Status

IMPORTANT

The Monitoring entitlement is required to view this tab.

The **`Probe Status`** page is shown by default when you click **`Monitoring`** in the top navigation bar.

The **Probe Status** page displays the summary count of probes in the various states and provides a simple interface to find problematic probes quickly. Note that the probe totals in the tabs at the top of the page may not match the numbers of probes displayed in the tables below. The counts at the top include probes for all systems in your organization, while the tables display probes on only those systems to which you have access through the System Group Administrator role. Also, the probe counts displayed here may be out of sync by as much as one minute.

The following list describes each state and identifies the icons associated with them:

-  - *Critical* - The probe has crossed a CRITICAL threshold.
-  - *Warning* - The probe has crossed a WARNING threshold.
-  - *Unknown* - The probe is not able to accurately report metric or state data.
-  - *Pending* - The probe has been scheduled but has not yet run or is unable to run.
-  - *OK* - The probe is running successfully.

The **Probe Status** page contains tabs for each of the possible states, as well as one that lists all probes. Each table contains columns indicating probe state, the monitored system, the probes used, and the date and time the status was last updated.

In these tables, clicking the name of the system takes you to the **Monitoring** tab of the **System Details** page. Clicking the name of the probe takes you to its**Current State** page. From there, you may edit the probe, delete it, and generate reports based upon its results.

Monitoring data and probe status information that was previously available only through the web interface of the Satellite can now be exported as a CSV file. Click on the **Download CSV** links throughout the Monitoring pages to download CSV files of relevant information. The exported data may include, but is not limited to:

- Probe status
- All probes in a given state (OK, WARN, UNKNOWN, CRITICAL, PENDING)
- A Probe Event history

6.6.1.1. Probe Status ⇒ Critical

IMPORTANT

The Monitoring entitlement is required to view this tab.

The probes that have crossed their CRITICAL thresholds or reached a critical status by some other means. For instance, some probes become critical (rather than unknown) when exceeding their timeout period.

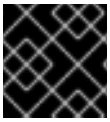
6.6.1.2. Probe Status ⇒ Warning

IMPORTANT

The Monitoring entitlement is required to view this tab.

The probes that have crossed their WARNING thresholds.

6.6.1.3. Probe Status ⇒ Unknown



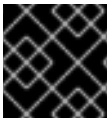
IMPORTANT

The Monitoring entitlement is required for this feature.

The probes that cannot collect the metrics needed to determine probe state. Most but not all probes enter an unknown state when exceeding their timeout period. This may mean that the timeout period should be increased, or the connection cannot be established to the monitored system.

It is also possible the probes' configuration parameters are not correct and their data cannot be found. Finally, this state may indicate that a software error has occurred.

6.6.1.4. Probe Status ⇒ Pending

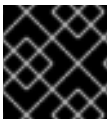


IMPORTANT

The Monitoring entitlement is required to view this tab.

The probes whose data have not been received by Red Hat Network. This state is expected for a probe that has just been scheduled but has not yet run. If all probes go into a pending state, your monitoring infrastructure may be failing.

6.6.1.5. Probe Status ⇒ OK



IMPORTANT

The Monitoring entitlement is required to view this tab.

The probes that have run successfully without exception. This is the state desired for all probes.

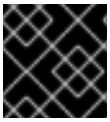
6.6.1.6. Probe Status ⇒ All

IMPORTANT

The Monitoring entitlement is required to view this tab.

All probes scheduled on systems in your account, listed in alphabetical order by the name of system.

6.6.1.7. Current State



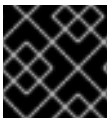
IMPORTANT

The Monitoring entitlement is required to view this tab.

Identifies the selected probe's status and when it last ran, while providing the ability to generate a report on the probe. Although this page is integral to monitoring, it is found under the **Probes** tab within the **System Details** page since its configuration is specific to the system being monitored.

To view a report of the probe's results, choose a relevant duration using the **date** fields and decide whether you would like to see metric data, the state change history or both. To obtain metric data, select the metric(s) on which you wish to see a report, and decide (using the checkboxes) whether the results should be shown in a graph, an event log, or both. Then click **Generate report** at the bottom of the page. If no data exist for the probe's metrics, you are presented with the following message: NO DATA FOR SELECTED TIME PERIOD AND METRIC.

6.6.2. Notification



IMPORTANT

The Monitoring entitlement is required to view this tab.

Identifies the contact methods that have been established for your organization. These methods contain email or pager addresses designated to receive alerts from probes.

The various notification methods available to your organization are listed here on the default **Notification** screen. The methods are listed according to the user to which they apply.

To create a new notification method, click on the name of the user to whom the notification will apply. The user's User Details ⇒ Notification Methods page appears. Click on the title of the notification method to edit the properties of the method.

6.6.2.1. Notification ⇒ Filters

Notification filters allow you to create long-term rules that suspend, redirect, or automatically acknowledge standard notifications or send supplemental notifications. This can be helpful in managing verbose or frequent probe communication.

6.6.2.1.1. Notification ⇒ Notification Filters ⇒ Active Filters

This is the default screen for the Notification Filters tab. It lists all active filters available for your organization. Click the name of the filter to edit the properties of the filter.

To create a notification filter, click the **create new notification filter** link in the upper right of the screen. Configure each option listed below and click the **Save Filter** button to create the filter.

1. *Description*: Enter a value that allows you to distinguish this filter from others.

2. *Type*: Determine what action the filter should take: redirect, acknowledge, suspend, or supplement the incoming notification.

3. *Send to*: The **Redirect Notification** and **Supplemental Notification** options in step two require an email address to which to send the notifications. The remaining options require no email address.

4. *Scope*: Determine which monitoring components are subject to the filter.

5. *Organization/Scout/Probe*: This option allows you to select the organization, scout(s), or probe(s) to which this filter applies. To select multiple items from the list, hold the **`Ctrl`** key while clicking the names of the items. To select a range of items, hold the **`Shift`** key while clicking on the first and last items in the range.

6. *Probes in State*: Select which probe state(s) relate to the filter. For example, you may choose to create a supplemental notification for critical probes only. Uncheck the box to the left of any state you want the filter to ignore.

7. *Notifications sent to*: This is the method to which the notification would be sent if no filter were in place. You may, for example, redirect notifications that would normally go to a user should that individual go on vacation, leaving all other notifications from the probe unchanged.

8. *Match Output*: Select precise notification results by entering a regular expression here. If the "Message:" portion of the notification does not match the regular expression, the filter is not applied.

9. *Recurring*: Select whether a filter runs continuously or on a recurring basis. A recurring filter runs multiple times for a period of time smaller than the duration of the filter. For example, a recurring filter could run for 10 minutes of every hour between the start and end times of the filter. A non-recurring filter runs continuously between the start and end times of the filter.

10. *Beginning*: Enter a date and time for the filter to begin operation.

11. *Ending*: Enter an end date and time for the filter.

12. *Recurring Duration*: How long a recurring filter instance is active. This field, applicable to recurring filters only, begins at the **`Beginning`** time specified above. Any notification generated outside of the specified duration is not filtered.

13. *Recurring Frequency*: How often the filter activates.

Notification filters cannot be deleted. However, a filter may be canceled by setting the end date to some time in the past. (Note that the end date must be equal to or later than the start date, or the change fails.) Another method is to select a set of filters from the **`Active`** page and click the **`Expire Notification Filters`** button in the lower right. These filters are then canceled and appear in the **`Expired Filters`** tab.

6.6.2.1.2. Notification ⇒ Notification Filters ⇒ Expired Filters

This tab lists all notification filters whose end date has passed. Expired filters are stored indefinitely; this allows an organization to recycle useful filters as needed and provides a historical record for troubleshooting.

6.6.3. Probe Suites

Probe Suites allow you to configure and apply one or more probes to a system or systems. Probe Suites may be configured once and then applied to any number of systems in a batch. This results in time savings and consistency for Monitoring customers.

To create and apply a Probe Suite, first create an empty Probe Suite, then configure member probes, and finally apply the Suite to selected systems.

1. From the Monitoring ⇒ Probe Suites page, select the **create probe suite** link. Enter an easily distinguishable name for the Probe Suite. You may also choose to add a brief description of the Suite. Click the **Create Probe Suite** button to continue.

2. Add and configure the probes that comprise the Suite. Click the **create new probe** link in the upper right.

3. Configure the probe and click the **Create Probe** button in the lower right. Repeat this process until all desired probes have been added.

NOTE

Sendmail must be configured correctly on your Red Hat Satellite and each client system to which the Probe Suite is applied must have the **rhnmd** daemon installed and running. See the *Red Hat Satellite Installation Guide* for additional information.

4. On the "Systems" tab, add the systems to which the Probe Suite applies. Click the **add systems to probe suite** link in the upper right of the screen to continue.

5. The next page displays a list of all systems with Monitoring entitlements. Check the box to the left of the system(s) to which you wish to apply the Probe Suite, select the monitoring scout you wish to use, and click the **Add systems to probe suite** button to complete the creation of the Probe Suite.

You can either delete or detach probes from the suite. Detaching a probe disassociates the probes from the suite and converts them to system-specific probes for the specified system. This means that changes to the detached probes only effect that system. Deleting a probe removes it from the Suite for all systems.

To remove probes from the Probe Suite:

1. From the Monitoring ⇒ Probe Suites page, click on the title of the Probe Suite you wish to alter.

2. Select the **Probes** sub-tab.

3. Check the box next to the probe you wish to remove.

4. Click the **Delete probes from Probe Suites** button.

You may also remove a system from the Probe Suite. There are two ways to accomplish this. The first method is to detach the system from the Probe Suite. When you do so, the system still has the same probes assigned to it. However, you now have the ability to configure these probes individually without affecting any other systems.

To detach a system from the suite:

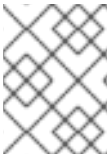
1. From the **Monitoring** ⇒ **Probe Suites** page, click on the title of the Probe Suite you wish to alter.

2. Select the **Systems** sub-tab.

3. Check the box next to the system(s) you wish to remove from the Probe Suite.

4. Click the **Detach System(s) from Probe Suite** button

The second method is to remove the system from the suite. This removes the system from the suite and deletes all running probes from the system.



NOTE

This action deletes all of the Probe Suites' probes from the system as well as all of the historical Time Series and Event Log data. This action is irreversible.

To remove a system from the Probe Suite and delete all associated probes from the system:

1. From the Monitoring ⇒ Probe Suites page, click on the title of the Probe Suite you wish to alter.
2. Select the **`Systems`** sub-tab.
3. Check the box next to the system(s) you wish to remove from the Probe Suite.
4. Click the **`Remove System(s) from Probe Suite`** button.

Finally, as with single Probes, you may download a CSV file containing information about Probe Suites. Click the **`Download CSV`** link at the bottom of the **`Monitoring`** ⇒ **`Probe Suites`** page to download the file.

6.6.4. Scout Config Push

IMPORTANT

The Monitoring entitlement is required to view this tab.

Displays the status of your monitoring infrastructure. Anytime you make a change to your monitoring configuration, such as adding a probe to a system or editing a probe's thresholds, you must reconfigure your monitoring infrastructure. Do this by selecting the Red Hat Network Server's checkbox and clicking **`Push Scout Configs`**. The table on this page identifies the date and time of requested and completed pushes.

Clicking the name of the server opens its Red Hat Network Monitoring Daemon SSH Public Key. This allows you to copy and paste the SSH key to the systems that are monitored by the scout. This is required in order for the Red Hat Network Monitoring Daemon to connect to the Satellite.

6.6.5. General Monitoring Config

IMPORTANT

The Monitoring entitlement is required to view this tab.

The General Monitoring Config page is in **Admin → Red Hat Satellite Configuration → Monitoring**. It collects information that is universally applicable to your Monitoring infrastructure. Modifying anything on this page causes the Monitoring services on the Red Hat Satellite to reset. It also schedules restart events for the Monitoring services on all Monitoring-enabled Red Hat Satellite Proxy Servers that connect to this Satellite. This is done so that the Monitoring services on these servers immediately reload their configuration.

Typically, the defaults provided in other fields are acceptable, since they are derived from your Satellite

installation. Nevertheless, you may use the fields on this page to alter your Monitoring configuration. For instance, you may change your mail exchange server here. This page also allows you to alter the destination of all administrative emails from the Satellite. When finished, click **`Update Config`**.

6.7. MONITORING TABLESPACES

Satellite Server 5.6 and later uses an embedded PostgreSQL database as the default database configuration. PostgreSQL does not require manual allocation of more space for the **Satellite** schema as it grows. PostgreSQL expands the schema of its own accord, provided that free disk space is available.

To check the database's size, run the **`db-control report`** command as the **`root`** user.

To check the size of each table in the database, run the **`db-control tablesizes`** command as the **`root`** user.

If you are using an *external* Oracle database, consult your database administrator for information about how to monitor tablespaces.

6.8. MONITORING RED HAT SATELLITE SERVER PROCESSES

Use the **`rhn-satellite status`** command to verify that all services related to the **Satellite Server** are running:

```
# rhn-satellite status
```

CHAPTER 7. MAINTAINING SYSTEM SECURITY USING OPENSCAP

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.

7.1. OPENSCAP 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.

7.2. OPENSCAP 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.

7.3. RED HAT SATELLITE PREREQUISITES FOR USING OPENSCAP

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-oscap 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.

7.4. 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.

7.4.1. Using the Web Interface to Perform Audit Scans

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

Procedure 7.1. To Perform an Audit Scan Using the Web Interface:

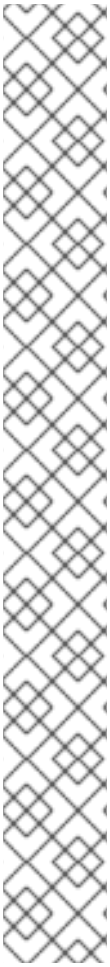
1. Log in to the Satellite web interface.
2. Click **Systems** → **system_name.**

3. Click **Audit** → **Schedule.**

4. Complete the ***`Schedule New XCCDF Scan`*** page. See Section 7.5.2.3, “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-oscap`** to fail to validate or run. Due to security concerns the **`oscap 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 7.4.3, “Viewing the Results of SCAP Audits”.

7.4.2. Using the API to Perform Audit Scans

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

Procedure 7.2. To Perform an Audit Scan Using the API:

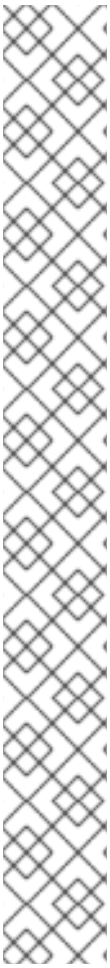
1. Choose an existing script or create a script for scheduling a system scan through **`system.scap.scheduleXccdfScan`**, the front-end API, for example:

```
#!/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:

- 1000010001 is the ***system ID (sid)***.
- **/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.
- **--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
```

7.4.3. 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 7.5, “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
```

7.5. OPENSCAP SATELLITE WEB INTERFACE

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

7.5.1. 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.

7.5.1.1. 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.

7.5.1.2. 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.

7.5.1.3. 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.

7.5.2. 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.**

7.5.2.1. List Scans

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

Table 7.1. 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.

7.5.2.2. 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.

7.5.2.3. 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-oscap** 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 7.4.1, “Using the Web Interface to Perform Audit Scans”.

CHAPTER 8. 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*.

8.1. 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 8.1. 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.

8.2. 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 8.2. 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.

8.3. 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 8.3. 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`**.

8.4. 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 9. 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 10. 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 10.1. Default Schedules in Red Hat Satellite 5.7

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.
sync-probe-default	sync-probe-bunch	Synchronizes probe state.

10.1. 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 10.1. 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 10.2. Editing Schedule Templates

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

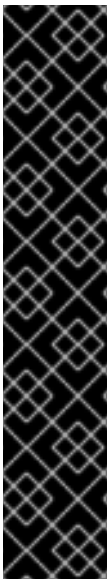
1. Log in to Satellite as the Organization Administrator.

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`**.

10.2. SETTING UP A SELF-SUBSCRIBED RED HAT SATELLITE

A self-subscribed Red Satellite 5 Server is registered to itself rather than the central Red Hat Network Classic Hosted servers. A Satellite server that is not registered to itself is registered to Red Hat Network Classic Hosted servers then activated as a Satellite Server. The Satellite can then use the **`satellite-sync`** command get new packages and content from the Red Hat Network Classic Hosted servers.

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 Network Classic Hosted servers. This process allows control of the Red Hat Enterprise Linux packages on the Satellite in the same manner as clients registered to the Satellite.



IMPORTANT

Self-subscribed Satellites have several limitations. These are:

- A self-subscribed Satellite cannot be used as means to monitor itself. Installing the client side **`rhnmd`** package will break the monitoring of the Satellite. Red Hat Network Classic Hosted provides custom monitoring probes that can be configured to monitor a self-subscribed Satellite.
- 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.

10.2.1. Installing and Configuring a Self-Subscribed Satellite

Procedure 10.3. 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*.
2. Install Red Hat Satellite 5 following the instructions provided in *Scenario 1: Installing Satellite with Embedded Database* in the *Red Hat Satellite 5 Installation Guide*. Allow the Satellite to register and activate the Satellite subscription to Red Hat Network Classic Hosted.
3. Use the **`satellite-sync`** command to download and import the base channel that matches the version of Red Hat Enterprise Linux installed on the Satellite server in Step 1. The **`satellite-sync`** command can import the necessary files either from the Red Hat Network Classic Hosted servers or 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. Use the following command to rename the **systemid** file. This file allows communication between the Satellite and the Red Hat Network Classic Hosted servers.

```
# mv /etc/sysconfig/rhn/systemid /etc/sysconfig/rhn/systemid.sat
```

6. Install the client side Satellite SSL certificate onto the Satellite using the following command.

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

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

   The **/etc/sysconfig/rhn/up2date** options will be set

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

NOTE

The HTTP proxy information that allows the Satellite access to the Red Hat Network Classic Hosted server must be removed from the configuration settings. This will permit the Update Agent to communicate with the Satellite.

Depending on network settings it may be necessary to use the main IP address associated with the default network card to communicate with the Satellite rather than the hostname or localhost.

8. Register the Satellite server using one of the following commands:

   - Red Hat Enterprise Linux 5, 6, and 7:

     rhnreg_ks --username *satellite_username* --password *satellite_password* command.

   - Red Hat Enterprise Linux 3 and 4:

     up2date --register

9. Once registration is complete rename the **systemid** file to **systemid.up2date** and rename the **systemid.sat** file back to **systemid** using the following commands.

```
# mv /etc/sysconfig/rhn/systemid /etc/sysconfig/rhn/systemid.up2date
# mv /etc/sysconfig/rhn/systemid.sat /etc/sysconfig/rhn/systemid
```

10. Change the **systemIdPath** option of **up2date** to the path of the **systemid.up2date** file.

```
systemIdPath=/etc/sysconfig/rhn/systemid.up2date
```

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

10.2.2. Testing Self-Subscribed Satellite Functionality

Procedure 10.4. Testing Self-Subscribed Satellite Functionality

1. Test the **satellite-sync** command by running the following command.

```
# satellite-sync -l
```

   Running the **satellite-sync** command should return information indicating the Satellite connects to satellite.rhn.redhat.com. The output should resemble the following:

```
16:50:22 Red Hat Network Satellite - live synchronization
16:50:22    url: https://satellite.rhn.redhat.com
16:50:22    debug/output level: 1
```

2. Test using one of the following commands:

   - Red Hat Enterprise Linux 5, 6, and 7:

     yum check-update

   - Red Hat Enterprise Linux 3 and 4:

     up2date -l

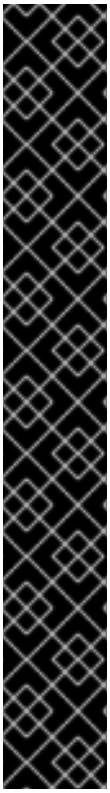
   This should display information indicating that packages are downloaded from the Satellite rather than from the Red Hat Network Classic Hosted server.

10.2.3. 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.

WARNING

Do not force the installation of the **rhnmd** client-side monitoring package onto a self-subscribed Satellite as this will break Monitoring.



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 with no Provisioning or Monitoring entitlements. 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 **up2date**, **rhn_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.

- *Applet*

  Both the **rhn-applet-tui** package and the **rhn-applet-gui** package will function normally. Installation and configuration of the **rhn-applet-tui** and **rhn-applet-gui** packages will complete normally. These packages allow client systems to communicate with the Satellite.

  NOTE

  The **rhn-applet-gui** requires packages that are not installed by default.

- *Configuration Client Tool*

  The **rhncfg-client** package will function normally after a change to the configuration file. Edit the **/etc/sysconfig/rhn/rhncfg-client.conf** file and change the **systemIdPath** option to match the path to **systemid.up2date** created in Procedure 10.3, “Installing and Configuring a Self-Subscribed Satellite”.

- *Configuration Management Tool*

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

- *Custom Info*

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

- *Client Monitoring*

  The **rhnmd** package will not install. The **rhnmd** package conflicts with the server-side monitoring packages. Do not attempt to force the installation of **rhnmd** as this will break Monitoring on the Satellite.

CHAPTER 11. 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.

11.1. 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
```

11.2. 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.

11.3. 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
```

11.4. 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.

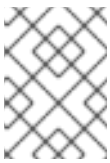
If also using one or more Red Hat Satellite Proxy Servers, ensure each Proxy's SSL certificates are prepared correctly. The Proxy should have both its own server SSL key-pair and CA SSL public (client) certificate installed, since it will serve in both capacities. See the SSL Certificates chapter of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

11.5. 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 11.1, “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 11.1. Log Files

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

Component/Task	Log File Location
Red Hat Satellite Synchronization Tool	**`/var/log/rhn/rhn_server_satellite.log`**
Monitoring infrastructure	**`/var/log/nocpulse/`** directory
Monitoring notifications	**`/var/log/notification/`** directory
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 11.2. `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

Report	Invoked as	Description
Relationship Mapping	`host-guests`	Provides host-guest mapping details
Inactive Systems	`inactive-systems`	
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

Report	Invoked as	Description
Users in System Groups	`system-groups-users`	Lists all system groups and their affiliated users
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**.

11.6. 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 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/rpclib.py", line 603, in
__call__
    return self._send(self._name, args)
  File "/usr/lib/python2.4/site-packages/rhn/rpclib.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/transports.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 the Satellite in connected mode, using the **rhn-satellite-activate** command as the root user, including the path and filename of the satellite certificate:

   ```
   # rhn-satellite-activate --rhn-cert=/path/to/file.cert
   ```

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 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 fix it?

A: If **satellite-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 certificate 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 certificate again on your Satellite using this command as the root user:

```
# rhn-satellite-activate -vvv --rhn-cert=/path/to/certificate
```

11.7. 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.

11.8. 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
```

11.9. 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

11.10. 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.

11.11. 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.

11.12. Monitoring

Q: Are there any diagnostic tools that help determine the cause of monitoring errors?

A: Though all monitoring-related activities are conducted through the Satellite interface, Red Hat provides access to some command line diagnostic tools that may help you determine the cause of errors. To use these tools, you must be able to become the **nocpulse** user on the Satellite conducting the monitoring.

First log into the Satellite as root. Then switch to the **nocpulse** user with the following command:

```
su - nocpulse
```

To thoroughly troubleshoot a probe, you must first obtain its probe ID. You may obtain this information by running **rhn-catalog** on the Red Hat Satellite Server as the **nocpulse** user. The output will resemble:

```
2 ServiceProbe on example1.redhat.com (199.168.36.245): test 2
3 ServiceProbe on example2.redhat.com (199.168.36.173): rhel2.1 test
4 ServiceProbe on example3.redhat.com (199.168.36.174): SSH
5 ServiceProbe on example4.redhat.com (199.168.36.175): HTTP
```

The probe ID is the first number, while the probe name (as entered in the Satellite interface) is the final entry on the line. In the above example, the 5 probe ID corresponds to the probe named HTTP.

Further, you may pass the **--commandline** (**-c**) and **--dump** (**-d**) options along with a probe ID to **rhn-catalog** to obtain additional details about the probe, like so:

```
rhn-catalog --commandline --dump 5
```

The **--commandline** option yields the command parameters set for the probe, while **--dump** retrieves everything else, including alert thresholds and notification intervals and methods.

The command above will result in output similar to:

```
5 ServiceProbe on example4.redhat.com (199.168.36.175  ):
linux:cpu usage
      Run as: Unix::CPU.pm --critical=90 --sshhost=199.168.36.175
--warn=70 --timeout=15 --sshuser=nocpulse
--shell=SSHRemoteCommandShell --sshport=4545
```

Now that you have the ID, use it with **rhn-runprobe** to examine the probe's output.

Q: How do I interpret the output of rhn-runprobe?

A: Now that you have obtained the probe ID with **rhn-catalog**, use it in conjunction with **rhn-runprobe** to examine the complete output of the probe. Note that by default, **rhn-runprobe** works in test mode, meaning no results are entered in the database. Here are its options:

Table 11.3. rhn-runprobe Options

Option	Description
--help	List the available options and exit.
--probe=*PROBE_ID*	Run the probe with this ID.
--prob_arg=*PARAMETER*	Override any probe parameters from the database.
--module=*PERL_MODULE*	Package name of alternate code to run.
--log=all=*LEVEL*	Set log level for a package or package prefix.
--debug=*LEVEL*	Set numeric debugging level.
--live	Execute the probe, queue data and send out notifications (if needed).

At a minimum, include the **--probe** option, the **--log** option, and values for each. The **--probe** option takes the probeID as its value and the **--log** option takes the value "all" (for all run levels) and a numeric verbosity level as its values. Here is an example:

```
rhn-runprobe --probe=5 --log=all=4
```

The above command requests the probe output for probeID 5, for all run levels, with a high level of verbosity.

More specifically, you may provide the command parameters derived from **rhn-catalog**, like so:

```
rhn-runprobe 5 --log=all=4 --sshuser=nocpulse --sshport=4545
```

This yields verbose output depicting the probe's attempted execution. Errors are clearly identified.

11.13. 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.

11.14. 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:

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

is installed on the Red Hat Satellite Proxy 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 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:

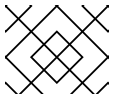
1. Stop the Apache Web server: **service httpd stop**
2. Stop the Squid server: **service squid stop**
3. Delete the contents of that directory: **rm -fv /var/spool/squid/***
4. Restart both services:

```
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.

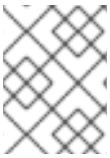
It may take several minutes for the system to generate and archive the report to a compressed file. Once finished, email the new file from the **/tmp/** directory to your Red Hat representative for immediate diagnosis.

APPENDIX A. PROBES

Monitoring-entitled systems can have probes applied to them that constantly confirm their health and full operability. This section lists the available probes broken down by command group, such as Apache.

Many probes that monitor internal system aspects (such as the Linux::Disk Usage probe) rather than external aspects (such as the Network Services::SSH probe) require the installation of the Red Hat Network monitoring daemon (**rhnmd**). This requirement is noted within the individual probe reference.

Each probe has its own reference in this section that identifies required fields (marked with *), default values, and the thresholds that may be set to trigger alerts. Similarly, the beginning of each command group's section contains information applicable to all probes in that group. Section A.1, “Probe Guidelines” covers general guidelines; the remaining sections examine individual probes.



NOTE

Nearly all of the probes use *Transmission Control Protocol* (TCP) as their transport protocol. Exceptions to this are noted within the individual probe references.

A.1. PROBE GUIDELINES

The following general guidelines outline the meaning of each probe state, and provide guidance in setting thresholds for your probes.

The following list provides a brief description of the meaning of each probe state:

Unknown

The probes that cannot collect the metrics needed to determine probe state. Most (though not all) probes enter this state when exceeding their timeout period. Probes in this state may be configured incorrectly, as well.

Pending

The probes whose data has not been received by the Red Hat Satellite. It is normal for new probes to be in this state. However, if all probes move into this state, the monitoring infrastructure may be failing.

OK

The probes that have run successfully without error. This is the desired state for all probes.

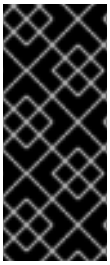
Warning

The probes that have crossed their WARNING thresholds.

Critical

The probes that have crossed their CRITICAL thresholds or reached a critical status by some other means. (Some probes become critical when exceeding their timeout period.)

While adding probes, select meaningful thresholds that, when crossed, notify you and your administrators of problems within your infrastructure. Timeout periods are entered in seconds unless otherwise indicated. Exceptions to these rules are noted within the individual probe references.



IMPORTANT

Some probes have thresholds based on time. In order for such CRITICAL and WARNING thresholds to work as intended, their values cannot exceed the amount of time allotted to the timeout period. Otherwise, an UNKNOWN status is returned in all instances of extended latency, thereby nullifying the thresholds. For this reason, Red Hat strongly recommends ensuring that timeout periods exceed all timed thresholds.

Run your probes without notifications for a time to establish baseline performance for each of your systems. Although the default values provided for probes may suit your needs, every organization has a different environment that may require altering thresholds.

A.2. APACHE 1.3.X AND 2.0.X

The probes in this section may be applied to instances of the Apache web server. Although the default values presume you will apply these probes using standard HTTP, you may also use them over secure connections by changing the application protocol to **`https`** and the port to **`443`**.

A.2.1. Apache::Processes

The Apache::Processes probe monitors the processes executed on an Apache web server and collects the following metrics:

- Data Transferred Per Child - Records data transfer information about individual children. A child process is one that is created from the parent process or another process.
- Data Transferred Per Slot - The cumulative amount of data transferred by a child process that restarts. The number of slots is configured in the **`httpd.conf`** file using the **`MaxRequestsPerChild`** setting.

The **`ExtendedStatus`** directive in the **`httpd.conf`** file of the Web server must be set to **`On`** for this probe to function properly.

Table A.1. Apache::Processes settings

Field	Value
Application Protocol*	http
Port*	80
Pathname*	/server-status
UserAgent*	NOCpulse-ApacheUptime/1.0
Username	
Password	
Timeout*	15

Field	Value
Critical Maximum Megabytes Transferred Per Child	
Warning Maximum Megabytes Transferred Per Child	
Critical Maximum Megabytes Transferred Per Slot	
Warning Maximum Megabytes Transferred Per Slot	

A.2.2. Apache::Traffic

The Apache::Traffic probe monitors the requests on an Apache web server and collects the following metrics:

- Current Requests - The number of requests being processed by the server at probe runtime.
- Request Rate - The accesses to the server per second since the probe last ran.
- Traffic - The kilobytes per second of traffic the server has processed since the probe last ran.

The **`ExtendedStatus`** directive in the **`httpd.conf`** file of the Web server must be set to **`On`** for this probe to function properly.

Table A.2. Apache::Traffic settings

Field	Value
Application Protocol*	http
Port*	80
Pathname*	/server-status
UserAgent*	NOCpulse-ApacheUptime/1.0
Username	
Password	
Timeout*	15
Critical Maximum Current Requests (number)	
Warning Maximum Current Requests (number)	
Critical Maximum Request Rate (events per second)	
Warning Maximum Request Rate (events per second)	

Field	Value
Critical Maximum Traffic (kilobytes per second)	
Warning Maximum Traffic (kilobytes per second)	

A.2.3. Apache::Uptime

The Apache::Uptime probe stores the cumulative time since the Web server was last started. No metrics are collected by this probe, which is designed to help track service level agreements (SLAs).

Table A.3. Apache::Uptime settings

Field	Value
Application Protocol*	http
Port*	80
Pathname*	/server-status
UserAgent*	NOCpulse-ApacheUptime/1.0
Username	
Password	
Timeout*	15

A.3. BEA WEBLOGIC 6.X AND HIGHER

The probes in this section (with the exception of JDBC Connection Pool) can be configured to monitor the properties of any BEA WebLogic 6.x and higher server (Administration or Managed) running on a given host, even in a clustered environment. Monitoring of a cluster is achieved by sending all SNMP queries to the Administration Server of the domain and then querying its Managed Servers for individual data.

In order to obtain this higher level of granularity, the **`BEA Domain Admin Server`** parameter must be used to differentiate between the Administration Server receiving SNMP queries and the Managed Server undergoing the specified probe. If the host to be probed is the Administration Server, then the **`BEA Domain Admin Server`** parameter can be left blank, and both the SNMP queries and the probe will be sent to it only.

If the host to be probed is a Managed Server, then the IP address of the Administration Server should be provided in the **`BEA Domain Admin Server`** parameter, and the Managed Server name should be included in the **`BEA Server Name`** parameter and appended to the end of the **`SNMP Community`**

String field. This causes the SNMP queries to be sent to the Administration Server host, as is required, but redirects the specific probe to the Managed Server host.

It should also be noted that the community string needed for probes run against Managed Server hosts should be in the form of **community_prefix@managed_server_name** in order for the SNMP query to return results for the desired Managed Server. Finally, SNMP must be enabled on each monitored system. SNMP support can be enabled and configured through the WebLogic Console.

See the documentation that came with your BEA server or information about the BEA website for more details about BEA's community string naming conventions.

A.3.1. BEA WebLogic::Execute Queue

The BEA WebLogic::Execute Queue probe monitors the WebLogic execute queue and provides the following metrics:

- Idle Execute Threads - The number of execution threads in an idle state.
- Queue Length - The number of requests in the queue.
- Request Rate - The number of requests per second.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.4. BEA WebLogic::Execute Queue settings

Field	Value
SNMP Community String*	public
SNMP Port*	161
SNMP Version*	1
BEA Domain Admin Server	
BEA Server Name*	myserver
Queue Name*	default
Critical Maximum Idle Execute Threads	
Warning Maximum Idle Execute Threads	
Critical Maximum Queue Length	
Warning Maximum Queue Length	
Critical Maximum Request Rate	
Warning Maximum Request Rate	

A.3.2. BEA WebLogic::Heap Free

The BEA WebLogic::Heap Free probe collects the following metric:

- Heap Free - The percentage of free heap space.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.5. BEA WebLogic::Heap Free settings

Field	Value
SNMP Community String*	public
SNMP Port*	161
SNMP Version*	1
BEA Domain Admin Server	
BEA Server Name*	myserver
Critical Maximum Heap Free	
Warning Maximum Heap Free	
Warning Minimum Heap Free	
Critical Minimum Heap Free	

A.3.3. BEA WebLogic::JDBC Connection Pool

The BEA WebLogic::JDBC Connection Pool probe monitors the Java Database Connection (JDBC) pool on a domain Admin Server only (no Managed Servers) and collects the following metrics:

- Connections - The number of connections to the JDBC.
- Connections Rate - The speed at which connections are made to the JDBC, measured in connections per second.
- Waiters - The number of sessions waiting to connect to the JDBC.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.6. BEA WebLogic::JDBC Connection Pool settings

Field	Value
SNMP Community String*	public
SNMP Port*	161

Field	Value
SNMP Version*	1
BEA Domain Admin Server	
BEA Server Name*	myserver
JDBC Pool Name*	MyJDBC Connection Pool
Critical Maximum Connections	
Warning Maximum Connections	
Critical Maximum Connection Rate	
Warning Maximum Connection Rate	
Critical Maximum Waiters	
Warning Maximum Waiters	

A.3.4. BEA WebLogic::Server State

The BEA WebLogic::Server State probe monitors the current state of a BEA Weblogic Web server. If the probe is unable to make a connection to the server, a CRITICAL status results.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.7. BEA WebLogic::Server State settings

Field	Value
SNMP Community String*	public
SNMP Port*	161
SNMP Version*	1
BEA Domain Admin Server	
BEA Server Name*	

A.3.5. BEA WebLogic::Servlet

The BEA WebLogic::Servlet probe monitors the performance of a particular servlet deployed on a WebLogic server and collects the following metrics:

- High Execution Time - The highest amount of time in milliseconds that the servlet takes to execute since the system was started.
- Low Execution Time - The lowest amount of time in milliseconds that the servlet takes to execute since the system was started.
- Execution Time Moving Average - A moving average of the execution time.
- Execution Time Average - A standard average of the execution time.
- Reload Rate - The number of times the specified servlet is reloaded per minute.
- Invocation Rate - The number of times the specified servlet is invoked per minute.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.8. BEA WebLogic::Servlet settings

Field	Value
SNMP Community String*	public
SNMP Port*	161
SNMP Version*	1
BEA Domain Admin Server	
BEA Server Name*	myserver
Servlet Name*	
Critical Maximum High Execution Time	
Warning Maximum High Execution Time	
Critical Maximum Execution Time Moving Average	
Warning Maximum Execution Time Moving Average	

A.4. GENERAL

The probes in this section are designed to monitor basic aspects of your systems. When applying them, ensure their timed thresholds do not exceed the amount of time allotted to the timeout period. Otherwise, the probe returns an UNKNOWN status in all instances of extended latency, thereby nullifying the thresholds.

A.4.1. General::Remote Program

The General::Remote Program probe allows you to run any command or script on your system and obtain a status string. Note that the resulting message will be limited to 1024 bytes.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe.

Table A.9. General::Remote Program settings

Field	Value
Command*	
OK Exit Status*	0
Warning Exit Status*	1
Critical Exit Status*	2
Timeout	15

A.4.2. General::Remote Program with Data

The General::Remote Program with Data probe allows you to run any command or script on your system and obtain a value, as well as a status string. To use this probe, you must include XML code in the body of your script. This probe supports the following XML tags:

- <perldata> </perldata>
- <hash> </hash>
- <item key =" "> </item>

The remote program will need to output some iteration of the following code to **STDOUT**:

```
<perldata> <hash> <item
key="data">10</item> <item
key="status_message">status message here</item>
</hash> </perldata>
```

The required value for **data** is the data point to be inserted in the database for time-series trending. The **status_message** is optional and can be whatever text string is desired with a maximum length of 1024 bytes. Remote programs that do not include a **status_message** still report the value and status returned.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe. XML is case-sensitive. The **data** item key name cannot be changed and it must collect a number as its value.

Table A.10. General::Remote Program with Data settings

Field	Value
Command*	

Field	Value
OK Exit Status*	0
Warning Exit Status*	1
Critical Exit Status*	2
Timeout	15

A.4.3. General::SNMP Check

The General::SNMP Check probe tests your SNMP server by specifying a single object identifier (OID) in dotted notation (such as **`1.3.6.1.2.1.1.1.0`**) and a threshold associated with the return value. It collects the following metric:

- Remote Service Latency - The time it takes in seconds for the SNMP server to answer a connection request.

Requirements - SNMP must be running on the monitored system to perform this probe. Only integers can be used for the threshold values.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.11. General::SNMP Check settings

Field	Value
SNMP OID*	
SNMP Community String*	public
SNMP Port*	161
SNMP Version*	2
Timeout*	15
Critical Maximum Value	
Warning Maximum Value	
Warning Minimum Value	
Critical Minimum Value	

A.4.4. General::TCP Check

The General::TCP Check probe tests your TCP server by verifying that it can connect to a system via the specified port number. It collects the following metric:

- Remote Service Latency - The time it takes in seconds for the TCP server to answer a connection request.

The probe passes the string specified in the **Send** field upon making a connection. The probe anticipates a response from the system, which should include the substring specified in the **`Expect`** field. If the expected string is not found, the probe returns a CRITICAL status.

Table A.12. General::TCP Check settings

Field	Value
Send	
Expect	
Port*	1
Timeout*	10
Critical Maximum Latency	
Warning Maximum Latency	

A.4.5. General::UDP Check

The General::UDP Check probe tests your UDP server by verifying that it can connect to a system via the specified port number. It collects the following metric:

- Remote Service Latency - The time it takes in seconds for the UDP server to answer a connection request.

The probe passes the string specified in the **Send** field upon making a connection. The probe anticipates a response from the system, which should include the substring specified in the **`Expect`** field. If the expected string is not found, the probe returns a CRITICAL status.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.13. General::UDP Check settings

Field	Value
Port*	1
Send	
Expect	
Timeout*	10

Field	Value
Critical Maximum Latency	
Warning Maximum Latency	

A.4.6. General::Uptime (SNMP)

The General::Uptime (SNMP) probe records the time since the device was last started. It uses the SNMP object identifier (OID) to obtain this value. The only error status it will return is UNKNOWN.

Requirements - SNMP must be running on the monitored system and access to the OID must be enabled to perform this probe.

This probe's transport protocol is User Datagram Protocol (UDP).

Table A.14. General::Uptime (SNMP) settings

Field	Value
SNMP Community String*	public
SNMP Port*	161
SNMP Version*	2
Timeout*	15

A.5. LINUX

The probes in this section monitor essential aspects of your Linux systems, from CPU usage to virtual memory. Apply them to mission-critical systems to obtain warnings prior to failure.

Unlike other probe groups, which may or may not require the Red Hat Network monitoring daemon, every Linux probe requires that the **rhnmd** daemon be running on the monitored system.

A.5.1. Linux::CPU Usage

The Linux::CPU Usage probe monitors the CPU utilization on a system and collects the following metric:

- CPU Percent Used - The five-second average of the percent of CPU usage at probe execution.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to run this probe.

Table A.15. Linux::CPU Usage settings

Field	Value
Timeout*	15

Field	Value
Critical Maximum CPU Percent Used	
Warning Maximum CPU Percent Used	

A.5.2. Linux::Disk IO Throughput

The Linux::Disk IO Throughput probe monitors a given disk and collects the following metric:

- Read Rate - The amount of data that is read in kilobytes per second.
- Write Rate - The amount of data that is written in kilobytes per second.

To obtain the value for the required **`Disk number or disk name`** field, run **`iostat`** on the system to be monitored and see what name has been assigned to the disk you desire. The default value of **`0`** usually provides statistics from the first hard drive connected directly to the system.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe. Also, the **`Disk number or disk name`** parameter must match the format visible when the **`iostat`** command is run. If the format is not identical, the configured probe enters an UNKNOWN state.

Table A.16. Linux::Disk IO Throughput settings

Field	Value
Disk number or disk name*	0
Timeout*	15
Critical Maximum KB read/second	
Warning Maximum KB read/second	
Warning Minimum KB read/second	
Critical Minimum KB read/second	
Critical Maximum KB written/second	
Warning Maximum KB written/second	
Warning Minimum KB written/second	
Critical Minimum KB written/second	

A.5.3. Linux::Disk Usage

The Linux::Disk Usage probe monitors the disk space on a specific file system and collects the following metrics:

- File System Used - The percentage of the file system currently in use.
- Space Used - The amount of the file system in megabytes currently in use.
- Space Available - The amount of the file system in megabytes currently available.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe.

Table A.17. Linux::Disk Usage settings

Field	Value
File system*	/dev/hda1
Timeout*	15
Critical Maximum File System Percent Used	
Warning Maximum File System Percent Used	
Critical Maximum Space Used	
Warning Maximum Space Used	
Warning Minimum Space Available	
Critical Minimum Space Available	

A.5.4. Linux::Inodes

The Linux::Inodes probe monitors the specified file system and collects the following metric:

- Inodes - The percentage of inodes currently in use.

An inode is a data structure that holds information about files in a Linux file system. There is an inode for each file, and a file is uniquely identified by the file system on which it resides and its inode number on that system.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe.

Table A.18. Linux::Inodes settings

Field	Value
File system*	/

Field	Value
Timeout*	15
Critical Maximum Inodes Percent Used	
Warning Maximum Inodes Percent Used	

A.5.5. Linux::Interface Traffic

The Linux::Interface Traffic probe measures the amount of traffic into and out of the specified interface (such as eth0) and collects the following metrics:

- Input Rate - The traffic in bytes per second going into the specified interface.
- Output Rate - The traffic in bytes per second going out of the specified interface.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe.

Table A.19. Linux::Interface Traffic settings

Field	Value
Interface*	
Timeout*	30
Critical Maximum Input Rate	
Warning Maximum Input Rate	
Warning Minimum Input Rate	
Critical Minimum Input Rate	
Critical Maximum Output Rate	
Warning Maximum Output Rate	
Warning Minimum Output Rate	
Critical Minimum Output Rate	

A.5.6. Linux::Load

The Linux::Load probe monitors the CPU of a system and collects the following metric:

- Load - The average load on the system CPU over various periods.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe.

Table A.20. Linux::Load settings

Field	Value
Timeout*	15
Critical CPU Load 1-minute average	
Warning CPU Load 1-minute average	
Critical CPU Load 5-minute average	
Warning CPU Load 5-minute average	
Critical CPU Load 15-minute average	
Warning CPU Load 15-minute average	

A.5.7. Linux::Memory Usage

The Linux::Memory Usage probe monitors the memory on a system and collects the following metric:

- RAM Free - The amount of free random access memory (RAM) in megabytes on a system.

You can also include the reclaimable memory in this metric by entering **yes** or **no** in the **Include reclaimable memory** field.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe.

Table A.21. Linux::Memory Usage settings

Field	Value
Include reclaimable memory	no
Timeout*	15
Warning Maximum RAM Free	
Critical Maximum RAM Free	

A.5.8. Linux::Process Counts by State

The Linux::Process Counts by State probe identifies the number of processes in the following states:

- Blocked - A process that has been switched to the waiting queue and whose state has been switched to **`waiting`**.
- Defunct - A process that has terminated (either because it has been killed by a signal or because it has called **`exit()`**) and whose parent process has not yet received notification of its termination by executing some form of the **`wait()`** system call.
- Stopped - A process that has been stopped before its execution could be completed.
- Sleeping - A process that is in the **`Interruptible`** sleep state and that can later be reintroduced into memory, resuming execution where it left off.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe.

Table A.22. Linux::Process Counts by State settings

Field	Value
Timeout*	15
Critical Maximum Blocked Processes	
Warning Maximum Blocked Processes	
Critical Maximum Defunct Processes	
Warning Maximum Defunct Processes	
Critical Maximum Stopped Processes	
Warning Maximum Stopped Processes	
Critical Maximum Sleeping Processes	
Warning Maximum Sleeping Processes	
Critical Maximum Child Processes	
Warning Maximum Child Processes	

A.5.9. Linux::Process Count Total

The Linux::Process Count Total probe monitors a system and collects the following metric:

- Process Count - The total number of processes currently running on the system.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe.

Table A.23. Linux::Process Count Total settings

Field	Value
Timeout*	15
Critical Maximum Process Count	
Warning Maximum Process Count	

A.5.10. Linux::Process Health

The Linux::Process Health probe monitors user-specified processes and collects the following metrics:

- CPU Usage - The CPU usage rate for a given process in milliseconds per second. This metric reports the time column of **ps** output, which is the cumulative CPU time used by the process. This makes the metric independent of probe interval, allows sane thresholds to be set, and generates usable graphs (i.e. a sudden spike in CPU usage shows up as a spike in the graph).
- Child Process Groups - The number of child processes spawned from the specified parent process. A child process inherits most of its attributes, such as open files, from its parent.
- Threads - The number of running threads for a given process. A thread is the basic unit of CPU utilization, and consists of a program counter, a register set, and a stack space. A thread is also called a lightweight process.
- Physical Memory Used - The amount of physical memory (or RAM) in kilobytes used by the specified process.
- Virtual Memory Used - The amount of virtual memory in kilobytes used by the specified process, or the size of the process in real memory plus swap.

Specify the process by its command name or process ID. (PID). Entering a PID overrides the entry of a command name. If no command name or PID is entered, the error Command not found is displayed and the probe will be set to a CRITICAL state.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe.

Table A.24. Linux::Process Health settings

Field	Value
Command Name	
Process ID (PID) file	
Timeout*	15
Critical Maximum CPU Usage	
Warning Maximum CPU Usage	

Field	Value
Critical Maximum Child Process Groups	
Warning Maximum Child Process Groups	
Critical Maximum Threads	
Warning Maximum Threads	
Critical Maximum Physical Memory Used	
Warning Maximum Physical Memory Used	
Critical Maximum Virtual Memory Used	
Warning Maximum Virtual Memory Used	

A.5.11. Linux::Process Running

The Linux::Process Running probe verifies that the specified process is functioning properly. It counts either processes or process groups, depending on whether the **`Count process groups`** checkbox is selected.

By default, the checkbox is selected, thereby indicating that the probe should count the number of process group leaders independent of the number of children. This allows you, for example, to verify that two instances of the Apache web server are running regardless of the (dynamic) number of child processes. If it is not selected, the probe conducts a straightforward count of the number of processes (children and leaders) matching the specified process.

Specify the process by its command name or process ID. (PID). Entering a PID overrides the entry of a command name. If no command name or PID is entered, the error Command not found is displayed and the probe enters a CRITICAL state.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe.

Table A.25. Linux::Process Running settings

Field	Value
Command name	
PID file	
Count process groups	(checked)
Timeout*	15

Field	Value
Critical Maximum Number Running	
Critical Minimum Number Running	

A.5.12. Linux::Swap Usage

The Linux::Swap Usage probe monitors the swap partitions running on a system and reports the following metric:

- Swap Free - The percent of swap memory currently free.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe.

Table A.26. Linux::Swap Usage settings

Field	Value
Timeout*	15
Warning Minimum Swap Free	
Critical Minimum Swap Free	

A.5.13. Linux::TCP Connections by State

The Linux::TCP Connections by State probe identifies the total number of TCP connections, as well as the quantity of each in the following states:

- TIME_WAIT - The socket is waiting after close for remote shutdown transmission so it may handle packets still in the network.
- CLOSE_WAIT - The remote side has been shut down and is now waiting for the socket to close.
- FIN_WAIT - The socket is closed, and the connection is now shutting down.
- ESTABLISHED - The socket has a connection established.
- SYN_RCVD - The connection request has been received from the network.

This probe can be helpful in finding and isolating network traffic to specific IP addresses or examining network connections into the monitored system.

The filter parameters for the probe let you narrow the probe's scope. This probe uses the **netstat -ant** command to retrieve data. The **Local IP address** and **Local port** parameters use values in the **Local Address** column of the output; the **Remote IP address** and **Remote port** parameters use values in the **Foreign Address** column of the output for reporting.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe.

Table A.27. Linux::TCP Connections by State settings

Field	Value
Local IP address filter pattern list	
Local port number filter	
Remote IP address filter pattern list	
Remote port number filter	
Timeout*	15
Critical Maximum Total Connections	
Warning Maximum Total Connections	
Critical Maximum TIME_WAIT Connections	
Warning Maximum TIME_WAIT Connections	
Critical Maximum CLOSE_WAIT Connections	
Warning Maximum CLOSE_WAIT Connections	
Critical Maximum FIN_WAIT Connections	
Warning Maximum FIN_WAIT Connections	
Critical Maximum ESTABLISHED Connections	
Warning Maximum ESTABLISHED Connections	
Critical Maximum SYN_RCVD Connections	
Warning Maximum SYN_RCVD Connections	

A.5.14. Linux::Users

The Linux::Users probe monitors the users of a system and reports the following metric:

- Users - The number of users currently logged in.

Requirements - The Red Hat Network monitoring daemon (**`rhnmd`**) must be running on the monitored system to execute this probe.

Table A.28. Linux::Users settings

Field	Value
Timeout*	15
Critical Maximum Users	
Warning Maximum Users	

A.5.15. Linux::Virtual Memory

The Linux::Virtual Memory probe monitors the total system memory and collects the following metric:

- Virtual Memory - The percent of total system memory - random access memory (RAM) plus swap - that is free.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe.

Table A.29. Linux::Virtual Memory settings

Field	Value
Timeout*	15
Warning Minimum Virtual Memory Free	
Critical Minimum Virtual Memory Free	

A.6. LOGAGENT

The probes in this section monitor the log files on your systems. You can use them to query logs for certain expressions and track the sizes of files. For LogAgent probes to run, the **nocpulse** user must be granted read access to your log files.

Note that data from the first run of these probes is not measured against the thresholds to prevent spurious notifications caused by incomplete metric data. Measurements will begin on the second run.

A.6.1. LogAgent::Log Pattern Match

The LogAgent::Log Pattern Match probe uses regular expressions to match text located within the monitored log file and collects the following metrics:

- Regular Expression Matches - The number of matches that have occurred since the probe last ran.
- Regular Expression Match Rate - The number of matches per minute since the probe last ran.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe. For this probe to run, the **nocpulse** user must be granted read access to your log files.

In addition to the name and location of the log file to be monitored, you must provide a regular expression to be matched against. The expression must be formatted for **egrep**, which is equivalent to **grep -E** and supports extended regular expressions. This is the regular expression set for **egrep**:

```
^ beginning of line
$ end of line
. match one char
* match zero or more chars
[] match one character set, e.g. '[Ff]oo'
[^] match not in set '[^A-F]oo'
+ match one or more of preceding chars
? match zero or one of preceding chars
| or, e.g. a|b
() groups chars, e.g., (foo|bar) or (foo)+
```

WARNING

Do not include single quotation marks (') within the expression. Doing so causes **egrep** to fail silently and the probe to time out.

Table A.30. LogAgent::Log Pattern Match settings

Field	Value
Log file*	/var/log/messages
Basic regular expression*	
Timeout*	45
Critical Maximum Matches	
Warning Maximum Matches	
Warning Minimum Matches	
Critical Minimum Matches	
Critical Maximum Match Rate	
Warning Maximum Match Rate	
Warning Minimum Match Rate	
Critical Maximum Match Rate	

A.6.2. LogAgent::Log Size

The LogAgent::Log Size probe monitors log file growth and collects the following metrics:

- Size - The size the log file has grown in bytes since the probe last ran.
- Output Rate - The number of bytes per minute the log file has grown since the probe last ran.
- Lines - The number of lines written to the log file since the probe last ran.
- Line Rate - The number of lines written per minute to the log file since the probe last ran.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe. For this probe to run, the **nocpulse** user must be granted read access to your log files.

Table A.31. LogAgent::Log Size settings

Field	Value
Log file*	/var/log/messages
Timeout*	20
Critical Maximum Size	
Warning Maximum Size	
Warning Minimum Size	
Critical Minimum Size	
Critical Maximum Output Rate	
Warning Maximum Output Rate	
Warning Minimum Output Rate	
Critical Minimum Output Rate	
Critical Maximum Lines	
Warning Maximum Lines	
Warning Minimum Lines	
Critical Minimum Lines	
Critical Maximum Line Rate	

Field	Value
Warning Maximum Line Rate	
Warning Minimum Line Rate	
Critical Minimum Line Rate	

A.7. MYSQL 3.23 - 3.33

The probes in this section monitor aspects of the MySQL database using the **`mysqladmin`** binary. No specific user privileges are needed for these probes.

Note that the **`mysql-server`** package must be installed on the system conducting the monitoring for these probes to complete. See the MySQL Installation section of the *Red Hat Satellite Installation Guide* for instructions.

A.7.1. MySQL::Database Accessibility

The MySQL::Database Accessibility probe tests connectivity through a database account that has no database privileges. If no connection is made, a CRITICAL status results.

Table A.32. MySQL::Database Accessibility settings

Field	Value
Username*	
Password	
MySQL Port	3306
Database*	mysql
Timeout	15

A.7.2. MySQL::Opened Tables

The MySQL::Opened Tables probe monitors the MySQL server and collects the following metric:

- Opened Tables - The tables that have been opened since the server was started.

Table A.33. MySQL::Opened Tables settings

Field	Value
Username	

Field	Value
Password	
MySQL Port*	3306
Timeout	15
Critical Maximum Opened Objects	
Warning Maximum Opened Objects	
Warning Minimum Opened Objects	
Critical Minimum Opened Objects	

A.7.3. MySQL::Open Tables

The MySQL::Open Tables probe monitors the MySQL server and collects the following metric:

- Open Tables - The number of tables open when the probe runs.

Table A.34. MySQL::Open Tables settings

Field	Value
Username	
Password	
MySQL Port*	3306
Timeout	15
Critical Maximum Open Objects	
Warning Maximum Open Objects	
Warning Minimum Open Objects	
Critical Minimum Open Objects	

A.7.4. MySQL::Query Rate

The MySQL::Query Rate probe monitors the MySQL server and collects the following metric:

- Query Rate - The average number of queries per second per database server.

Table A.35. MySQL::Query Rate settings

Field	Value
Username	
Password	
MySQL Port*	3306
Timeout	15
Critical Maximum Query Rate	
Warning Maximum Query Rate	
Warning Minimum Query Rate	
Critical Minimum Query Rate	

A.7.5. MySQL::Threads Running

The MySQL::Threads Running probe monitors the MySQL server and collects the following metric:

- Threads Running - The total number of running threads within the database.

Table A.36. MySQL::Threads Running settings

Field	Value
Username	
Password	
MySQL Port*	3306
Timeout	15
Critical Maximum Threads Running	
Warning Maximum Threads Running	
Warning Minimum Threads Running	
Critical Minimum Threads Running	

A.8. NETWORK SERVICES

The probes in this section monitor various services integral to a functioning network. When applying them, ensure that their timed thresholds do not exceed the amount of time allotted to the timeout period. Otherwise, an UNKNOWN status is returned in all instances of extended latency, thereby nullifying the thresholds.

A.8.1. Network Services::DNS Lookup

The Network Services::DNS Lookup probe uses the **dig** command to see if it can resolve the system or domain name specified in the **Host or Address to look up** field. It collects the following metric:

- Query Time - The time in milliseconds required to execute the **dig** request.

This is useful in monitoring the status of your DNS servers. To monitor one of your DNS servers, supply a well-known host/domain name, such as a large search engine or corporate Web site.

Table A.37. Network Services::DNS Lookup settings

Field	Value
Host or Address to look up	
Timeout*	10
Critical Maximum Query Time	
Warning Maximum Query Time	

A.8.2. Network Services::FTP

The Network Services::FTP probe uses network sockets to test FTP port availability. It collects the following metric:

- Remote Service Latency - The time it takes in seconds for the FTP server to answer a connection request.

This probe supports authentication. Provide a username and password in the appropriate fields to use this feature.The optional **Expect** value is the string to be matched against after a successful connection is made to the FTP server. If the expected string is not found, the probe returns a CRITICAL state.

Table A.38. Network Services::FTP settings

Field	Value
Expect	FTP
Username	
Password	
FTP Port*	21

Field	Value
Timeout*	10
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.8.3. Network Services::IMAP Mail

The Network Services::IMAP Mail probe determines if it can connect to the IMAP 4 service on the system. Specifying an optional port will override the default port 143. It collects the following metric:

- Remote Service Latency - The time it takes in seconds for the IMAP server to answer a connection request.

The required **`Expect`** value is the string to be matched against after a successful connection is made to the IMAP server. If the expected string is not found, the probe returns a CRITICAL state.

Table A.39. Network Services::IMAP Mail settings

Field	Value
IMAP Port*	143
Expect*	OK
Timeout*	5
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.8.4. Network Services::Mail Transfer (SMTP)

The Network Services::Mail Transfer (SMTP) probe determines if it can connect to the SMTP port on the system. Specifying an optional port number overrides the default port 25. It collects the following metric:

- Remote Service Latency - The time it takes in seconds for the SMTP server to answer a connection request.

Table A.40. Network Services::Mail Transfer (SMTP) settings

Field	Value
SMTP Port*	25
Timeout*	10

Field	Value
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.8.5. Network Services::Ping

The Network Services::Ping probe determines if the Red Hat Satellite Server can **`ping`** the monitored system or a specified IP address. It also checks the packet loss and compares the round trip average against the Warning and Critical threshold levels. The required **`Packets to send`** value allows you to control how many ICMP ECHO packets are sent to the system. This probe collects the following metrics:

- Round-Trip Average - The time it takes in milliseconds for the ICMP ECHO packet to travel to and from the monitored system.
- Packet Loss - The percent of data lost in transit.

Although optional, the **`IP Address`** field can be instrumental in collecting metrics for systems that have multiple IP addresses. For instance, if the system is configured with multiple virtual IP addresses or uses Network Address Translation (NAT) to support internal and external IP addresses, this option may be used to check a secondary IP address rather than the primary address associated with the hostname.

Note that this probe conducts the **`ping`** from a Red Hat Satellite Server and not the monitored system. Populating the IP Address field does not test connectivity between the system and the specified IP address but between the Red Hat Satellite Server and the IP address. Therefore, entering the same IP address for Ping probes on different systems accomplishes precisely the same task. To conduct a **`ping`** from a monitored system to an individual IP address, use the Remote Ping probe instead. See Section A.8.7, “Network Services::Remote Ping”.

Table A.41. Network Services::Ping settings

Field	Value
IP Address (defaults to system IP)	
Packets to send*	20
Timeout*	10
Critical Maximum Round-Trip Average	
Warning Maximum Round-Trip Average	
Critical Maximum Packet Loss	
Warning Maximum Packet Loss	

A.8.6. Network Services::POP Mail

The Network Services::POP Mail probe determines if it can connect to the POP3 port on the system. A port number must be specified; specifying another port number overrides the default port 110. This probe collects the following metric:

- Remote Service Latency - The time it takes in seconds for the POP server to answer a connection request.

The required **Expect** value is the string to be matched against after a successful connection is made to the POP server. The probe looks for the string in the first line of the response from the system. The default is **+OK**. If the expected string is not found, the probe returns a CRITICAL state.

Table A.42. Network Services::POP Mail settings

Field	Value
Port*	110
Expect*	+OK
Timeout*	10
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.8.7. Network Services::Remote Ping

The Network Services::Remote Ping probe determines if the monitored system can **ping** a specified IP address. It also monitors the packet loss and compares the round trip average against the Warning and Critical threshold levels. The required **Packets to send** value allows you to control how many ICMP ECHO packets are sent to the address. This probe collects the following metrics:

- Round-Trip Average - The time it takes in milliseconds for the ICMP ECHO packet to travel to and from the IP address.
- Packet Loss - The percent of data lost in transit.

The **IP Address** field identifies the precise address to be pinged. Unlike the similar, optional field in the standard Ping probe, this field is required. The monitored system directs the ping to a third address, rather than to the Red Hat Satellite Server. Since the Remote Ping probe tests connectivity from the monitored system, another IP address must be specified. To conduct pings from the Red Hat Satellite Server to a system or IP address, use the standard Ping probe instead. See Section A.8.5, “Network Services::Ping”.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe.

Table A.43. Network Services::Remote Ping settings

Field	Value
IP Address*	

Field	Value
Packets to send*	20
Timeout*	10
Critical Maximum Round-Trip Average	
Warning Maximum Round-Trip Average	
Critical Maximum Packet Loss	
Warning Maximum Packet Loss	

A.8.8. Network Services::RPCService

The Network Services::RPCService probe tests the availability of remote procedure call (RPC) programs on a given IP address. It collects the following metric:

- Remote Service Latency - The time it takes in seconds for the RPC server to answer a connection request.

RPC server programs, which provide function calls via that RPC network, register themselves in the RPC network by declaring a program ID and a program name. NFS is an example of a service that works via the RPC mechanism.

Client programs that wish to use the resources of RPC server programs do so by asking the machine on which the server program resides to provide access to RPC functions within the RPC program number or program name. These conversations can occur over either TCP or UDP (but are almost always UDP).

This probe allows you to test simple program availability. You must specify the program name or number, the protocol over which the conversation occurs, and the usual timeout period.

Table A.44. Network Services::RPCService settings

Field	Value
Protocol (TCP/UDP)	udp
Service Name*	nfs
Timeout*	10
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.8.9. Network Services::Secure Web Server (HTTPS)

The Network Services::Secure Web Server (HTTPS) probe determines the availability of the secure Web server and collects the following metric:

- Remote Service Latency - The time it takes in seconds for the HTTPS server to answer a connection request.

This probe confirms that it can connect to the HTTPS port on the specified host and retrieve the specified URL. If no URL is specified, the probe fetches the root document. The probe looks for a HTTP/1. message from the system unless you alter that value. Specifying another port number overrides the default port of 443.

This probe supports authentication. Provide a username and password in the appropriate fields to use this feature. Unlike most other probes, this probe returns a CRITICAL status if it cannot contact the system within the timeout period.

Table A.45. Network Services::Secure Web Server (HTTPS) settings

Field	Value
URL Path	/
Expect Header	HTTP/1
Expect Content	
UserAgent*	NOCpulse-check_http/1.0
Username	
Password	
Timeout*	10
HTTPS Port*	443
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.8.10. Network Services::SSH

The Network Services::SSH probe determines the availability of SSH on the specified port and collects the following metric:

- Remote Service Latency - The time it takes in seconds for the SSH server to answer a connection request.

Upon successfully contacting the SSH server and receiving a valid response, the probe displays the protocol and server version information. If the probe receives an invalid response, it displays the message returned from the server and generates a WARNING state.

Table A.46. Network Services::SSH settings

Field	Value
SSH Port*	22
Timeout*	5
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.8.11. Network Services::Web Server (HTTP)

The Network Services::Web Server (HTTP) probe determines the availability of the Web server and collects the following metric:

- Remote Service Latency - The time it takes in seconds for the HTTP server to answer a connection request.

This probe confirms it can connect to the HTTP port on the specified host and retrieve the specified URL. If no URL is specified, the probe will fetch the root document. The probe looks for a HTTP/1. message from the system, unless you alter that value. Specifying another port number will override the default port of 80. Unlike most other probes, this probe will return a CRITICAL status if it cannot contact the system within the timeout period.

This probe supports authentication. Provide a username and password in the appropriate fields to use this feature. Also, the optional Virtual Host field can be used to monitor a separate documentation set located on the same physical machine presented as a standalone server. If your Web server is not configured to use virtual hosts (which is typically the case), you should leave this field blank. If you do have virtual hosts configured, enter the domain name of the first host here. Add as many probes as necessary to monitor all virtual hosts on the machine.

Table A.47. Network Services::Web Server (HTTP) settings

Field	Value
URL Path	/
Virtual Host	
Expect Header	HTTP/1
Expect Content	
UserAgent*	NOCpulse-check_http/1.0
Username	
Password	
Timeout*	10

Field	Value
HTTP Port*	80
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.9. ORACLE 8I, 9I, 10G, AND 11G

The probes in this section may be applied to instances of the Oracle database matching the versions supported. Oracle probes require the configuration of the database and associations made by running the following command:

```
$ORACLE_HOME/rdbms/admin/catalog.sql
```

In addition, for these probes to function properly, the Oracle user configured in the probe must have minimum privileges of CONNECT and SELECT_CATALOG_ROLE.

Some Oracle probes are specifically aimed at tuning devices for long-term performance gains, rather than avoiding outages. Therefore, Red Hat recommends scheduling them to occur less frequently, between every hour and every two days. This provides a better statistical representation, de-emphasizing anomalies that can occur at shorter time intervals. This applies to following probes: Buffer Cache, Data Dictionary Cache, Disk Sort Ratio, Library Cache, and Redo Log.

For CRITICAL and WARNING thresholds based upon time to work as intended, their values cannot exceed the amount of time allotted to the timeout period. Otherwise, an UNKNOWN status is returned in all cases of extended latency, thereby nullifying the thresholds. For this reason, Red Hat strongly recommends ensuring that timeout periods exceed all timed thresholds. In this section, this refers specifically to the probe TNS Ping.

Finally, customers using these Oracle probes against a database using Oracle's Multi-Threaded Server (MTS) must contact Red Hat support to have entries added to the Red Hat Network Server's /etc/hosts file to ensure that the DNS name is resolved correctly.

A.9.1. Oracle::Active Sessions

The Oracle::Active Sessions probe monitors an Oracle instance and collects the following metrics:

- Active Sessions - The number of active sessions based on the value of **V$PARAMETER.PROCESSES**.
- Available Sessions - The percentage of active sessions that are available based on the value of **V$PARAMETER.PROCESSES**.

Table A.48. Oracle::Active Sessions settings

Field	Value
Oracle SID*	

Field	Value
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Timeout*	30
Critical Maximum Active Sessions	
Warning Maximum Active Sessions	
Critical Maximum Available Sessions Used	
Warning Maximum Available Sessions Used	

A.9.2. Oracle::Availability

The Oracle::Availability probe determines the availability of the database from the Red Hat Satellite.

Table A.49. Oracle::Availability settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Timeout*	30

A.9.3. Oracle::Blocking Sessions

The Oracle::Blocking Sessions probe monitors an Oracle instance and collects the following metric:

- Blocking Sessions - The number of sessions preventing other sessions from committing changes to the Oracle database, as determined by the required *Time Blocking* value you provide. Only those sessions that have been blocking for this duration, which is measured in seconds, are counted as blocking sessions.

Table A.50. Oracle::Blocking Sessions settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Time Blocking (seconds)*	20
Timeout*	30
Critical Maximum Blocking Sessions	
Warning Maximum Blocking Sessions	

A.9.4. Oracle::Buffer Cache

The Oracle::Buffer Cache probe computes the Buffer Cache Hit Ratio so as to optimize the system global area (SGA) Database Buffer Cache size. It collects the following metrics:

- Db Block Gets - The number of blocks accessed via single block gets (not through the consistent get mechanism).
- Consistent Gets - The number of accesses made to the block buffer to retrieve data in a consistent mode.
- Physical Reads - The cumulative number of blocks read from disk.
- Buffer Cache Hit Ratio - The rate at which the database goes to the buffer instead of the hard disk to retrieve data. A low ratio suggests more RAM should be added to the system.

Table A.51. Oracle::Buffer Cache settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port	1521
Timeout*	30
Warning Minimum Buffer Cache Hit Ratio	

Field	Value
Critical Minimum Buffer Cache Hit Ratio	

A.9.5. Oracle::Client Connectivity

The Oracle::Client Connectivity probe determines if the database is up and capable of receiving connections from the monitored system. This probe opens an **rhnmd** connection to the system and issues a **sqlplus connect** command on the monitored system.

The **Expected DB name** parameter is the expected value of **V$DATABASE.NAME**. This value is case-insensitive. A CRITICAL status is returned if this value is not found.

Requirements - The Red Hat Network monitoring daemon (**rhnmd**) must be running on the monitored system to execute this probe. For this probe to run, the **nocpulse** user must be granted read access to your log files.

Table A.52. Oracle::Client Connectivity settings

Field	Value
Oracle Hostname or IP address*	
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
ORACLE_HOME*	/opt/oracle
Expected DB Name*	
Timeout*	30

A.9.6. Oracle::Data Dictionary Cache

The Oracle::Data Dictionary Cache probe computes the Data Dictionary Cache Hit Ratio so as to optimize the SHARED_POOL_SIZE in **init.ora**. It collects the following metrics:

- Data Dictionary Hit Ratio - The ratio of cache hits to cache lookup attempts in the data dictionary cache. In other words, the rate at which the database goes to the dictionary instead of the hard disk to retrieve data. A low ratio suggests more RAM should be added to the system.
- Gets - The number of blocks accessed via single block gets (not through the consistent get mechanism).

- Cache Misses - The number of accesses made to the block buffer to retrieve data in a consistent mode.

Table A.53. Oracle::Data Dictionary Cache settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Timeout*	30
Warning Minimum Data Dictionary Hit Ratio	
Critical Minimum Data Dictionary Hit Ratio	

A.9.7. Oracle::Disk Sort Ratio

The Oracle::Disk Sort Ratio probe monitors an Oracle database instance and collects the following metric:

- Disk Sort Ratio - The rate of Oracle sorts that were too large to be completed in memory and were instead sorted using a temporary segment.

Table A.54. Oracle::Disk Sort Ratio settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Timeout*	30
Critical Maximum Disk Sort Ratio	
Warning Maximum Disk Sort Ratio	

A.9.8. Oracle::Idle Sessions

The Oracle::Idle Sessions probe monitors an Oracle instance and collects the following metric:

- Idle Sessions - The number of Oracle sessions that are idle, as determined by the required *Time Idle* value you provide. Only those sessions that have been idle for this duration, which is measured in seconds, are counted as idle sessions.

Table A.55. Oracle::Idle Sessions settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Time Idle (seconds)*	20
Timeout*	30
Critical Maximum Idle Sessions	
Warning Maximum Idle Sessions	

A.9.9. Oracle::Index Extents

The Oracle::Index Extents probe monitors an Oracle instance and collects the following metric:

- Allocated Extents - The number of allocated extents for any index.
- Available Extents - The percentage of available extents for any index.

The required **`Index Name`** field contains a default value of `%` that matches any index name.

Table A.56. Oracle::Index Extents settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Index Owner*	%

Field	Value
Index Name*	%
Timeout*	30
Critical Maximum of Allocated Extents	
Warning Maximum of Allocated Extents	
Critical Maximum of Available Extents	
Warning Maximum of Available Extents	

A.9.10. Oracle::Library Cache

The Oracle::Library Cache probe computes the Library Cache Miss Ratio so as to optimize the SHARED_POOL_SIZE in **`init.ora`**. It collects the following metrics:

- Library Cache Miss Ratio - The rate at which a library cache pin miss occurs. This happens when a session executes a statement that it has already parsed but finds that the statement is no longer in the shared pool.
- Executions - The number of times a pin was requested for objects of this namespace.
- Cache Misses - The number of pins that must now retrieve the object of the disk. These pins are made up of objects with previous pins from the time the object handle was created.

Table A.57. Oracle::Library Cache settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Timeout*	30
Critical Maximum Library Cache Miss Ratio	
Warning Maximum Library Cache Miss Ratio	

A.9.11. Oracle::Locks

The Oracle::Locks probe monitors an Oracle database instance and collects the following metric:

- Active Locks - The current number of active locks as determined by the value in the v$locks table. Database administrators should be aware of high numbers of locks present in a database instance.

Locks are used so that multiple users or processes updating the same data in the database do not conflict. This probe is useful for alerting database administrators when a high number of locks are present in a given instance.

Table A.58. Oracle::Locks settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Timeout*	30
Critical Maximum Active Locks	
Warning Maximum Active Locks	

A.9.12. Oracle::Redo Log

The Oracle::Redo Log probe monitors an Oracle database instance and collects the following metrics:

- Redo Log Space Request Rate - The average number of redo log space requests per minute since the server has been started.
- Redo Buffer Allocation Retry Rate - The average number of buffer allocation retries per minute since the server was started.

The metrics returned and the thresholds they are measured against are numbers representing the rate of change in events per minute. The rate of change for these metrics should be monitored because fast growth can indicate problems requiring investigation.

Table A.59. Oracle::Redo Log settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521

Field	Value
Timeout*	30
Critical Maximum Redo Log Space Request Rate	
Warning Maximum Redo Log Space Request Rate	
Critical Maximum Redo Buffer Allocation Retry Rate	
Warning Maximum Redo Buffer Allocation Retry Rate	

A.9.13. Oracle::Table Extents

The Oracle::Table Extents probe monitors an Oracle database instance and collects the following metrics:

- Allocated Extents-Any Table - The total number of extents for any table.
- Available Extents-Any Table - The percentage of available extents for any table.

In Oracle, table extents allow a table to grow. When a table is full, it is *extended* by an amount of space configured when the table is created. Extents are configured on a per-table basis, with an extent size and a maximum number of extents.

For example, a table that starts with 10 MB of space and that is configured with an extent size of 1 MB and max extents of 10 can grow to a maximum of 20 MB (by being extended by 1 MB ten times). This probe can be configured to alert by (1) the number of allocated extents (e.g. "go critical when the table has been extended 5 or more times"), or (2) the table is extended past a certain percentage of its max extents (e.g. "go critical when the table has exhausted 80% or more of its max extents").

The required **`Table Owner`** and **`Table Name`** fields contain a default value of `%` that matches any table owner or name.

Table A.60. Oracle::Table Extents settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Table Owner*	%
Table Name*	%

Field	Value
Timeout*	30
Critical Maximum Allocated Extents	
Warning Maximum Allocated Extents	
Critical Maximum Available Extents	
Warning Maximum Available Extents	

A.9.14. Oracle::Tablespace Usage

The Oracle::Tablespace Usage probe monitors an Oracle database instance and collects the following metric:

- Available Space Used - The percentage of available space in each tablespace that has been used.

Tablespace is the shared pool of space in which a set of tables live. This probe alerts the user when the total amount of available space falls below the threshold. Tablespace is measured in bytes, so extents do not factor into it directly (though each extension removes available space from the shared pool).

The required **`Tablespace Name`** field is case insensitive and contains a default value of`%` that matches any table name.

Table A.61. Oracle::Tablespace Usage settings

Field	Value
Oracle SID*	
Oracle Username*	
Oracle Password*	
Oracle Port*	1521
Tablespace Name*	%
Timeout*	30
Critical Maximum Available Space Used	
Warning Maximum Available Space Used	

A.9.15. Oracle::TNS Ping

The Oracle::TNS Ping probe determines if an Oracle listener is alive and collects the following metric:

- Remote Service Latency - The time it takes in seconds for the Oracle server to answer a connection request.

Table A.62. Oracle::TNS Ping settings

Field	Value
TNS Listener Port*	1521
Timeout*	15
Critical Maximum Remote Service Latency	
Warning Maximum Remote Service Latency	

A.10. RED HAT SATELLITE

The probes in this section may be applied to the Red Hat Satellite itself to monitor its health and performance. Since these probes run locally, no specific application or transport protocols are required.

A.10.1. Red Hat Satellite::Disk Space

The Red Hat Satellite::Disk Space probe monitors the free disk space on a Satellite and collects the following metrics:

- File System Used - The percent of the current file system now in use.
- Space Used - The file size used by the current file system.
- Space Available - The file size available to the current file system.

Table A.63. Red Hat Satellite::Disk Space settings

Field	Value
Device Pathname*	/dev/hda1
Critical Maximum File System Used	
Warning Maximum File System Used	
Critical Maximum Space Used	
Warning Maximum Space Used	
Critical Maximum Space Available	
Warning Maximum Space Available	

A.10.2. Red Hat Satellite::Execution Time

The Red Hat Satellite::Execution Time probe monitors the execution time for probes run from a Satellite and collects the following metric:

- Probe Execution Time Average - The seconds required to fully execute a probe.

Table A.64. Red Hat Satellite::Execution Time settings

Field	Value
Critical Maximum Probe Execution Time Average	
Warning Maximum Probe Execution Time Average	

A.10.3. Red Hat Satellite::Interface Traffic

The Red Hat Satellite::Interface Traffic probe monitors the interface traffic on a Satellite and collects the following metrics:

- Input Rate - The amount of traffic in bytes per second the device receives.
- Output Rate - The amount of traffic in bytes per second the device sends.

Table A.65. Red Hat Satellite::Interface Traffic settings

Field	Value
Interface*	eth0
Timeout (seconds)*	30
Critical Maximum Input Rate	
Critical Maximum Output Rate	

A.10.4. Red Hat Satellite::Latency

The Red Hat Satellite::Latency probe monitors the latency of probes on a Satellite and collects the following metric:

- Probe Latency Average - The lag in seconds between the time a probe becomes ready to run and the time it is actually run. Under normal conditions, this is generally less than a second. When a Satellite is overloaded (because it has too many probes with respect to their average execution time), the number goes up.

Table A.66. Red Hat Satellite::Latency settings

Field	Value
Critical Maximum Probe Latency Average	

Field	Value
Warning Maximum Probe Latency Average	

A.10.5. Red Hat Satellite::Load

The Red Hat Satellite::Load probe monitors the CPU load on a Satellite and collects the following metric:

- Load - The load average on the CPU for a 1-, 5-, and 15-minute period.

Table A.67. Red Hat Satellite::Load settings

Field	Value
Critical Maximum 1-minute Average	
Warning Maximum 1-minute Average	
Critical Maximum 5-minute Average	
Warning Maximum 5-minute Average	
Critical Maximum 15-minute Average	
Warning Maximum 15-minute Average	

A.10.6. Red Hat Satellite::Probe Count

The Red Hat Satellite::Probe Count probe monitors the number of probes on a Satellite and collects the following metric:

- Probes - The number of individual probes running on a Satellite.

Table A.68. Red Hat Satellite::Probe Count settings

Field	Value
Critical Maximum Probe Count	
Warning Maximum Probe Count	

A.10.7. Red Hat Satellite::Process Counts

The Red Hat Satellite::Process Counts probe monitors the number of processes on a Satellite and collects the following metrics:

- Blocked - The number of processes that have been switched to the waiting queue and waiting state.

- Child - The number of processes spawned by another process already running on the machine.
- Defunct - The number of processes that have terminated (either because they have been killed by a signal or have called **`exit()`**) and whose parent processes have not yet received notification of their termination by executing some form of the **`wait()`** system call.
- Stopped - The number of processes that have stopped before their executions could be completed.
- Sleeping - A process that is in the **`Interruptible`** sleep state and that can later be reintroduced into memory, resuming execution where it left off.

Table A.69. Red Hat Satellite::Process Counts settings

Field	Value
Critical Maximum Blocked Processes	
Warning Maximum Blocked Processes	
Critical Maximum Child Processes	
Warning Maximum Child Processes	
Critical Maximum Defunct Processes	
Warning Maximum Defunct Processes	
Critical Maximum Stopped Processes	
Warning Maximum Stopped Processes	
Critical Maximum Sleeping Processes	
Warning Maximum Sleeping Processes	

A.10.8. Red Hat Satellite::Processes

The Red Hat Satellite::Processes probe monitors the number of processes on a Satellite and collects the following metric:

- Processes - The number of processes running simultaneously on the machine.

Table A.70. Red Hat Satellite::Processes settings

Field	Value
Critical Maximum Processes	
Warning Maximum Processes	

A.10.9. Red Hat Satellite::Process Health

The Red Hat Satellite::Process Health probe monitors customer-specified processes and collects the following metrics:

- CPU Usage - The CPU usage percent for a given process.
- Child Process Groups - The number of child processes spawned from the specified parent process. A child process inherits most of its attributes, such as open files, from its parent.
- Threads - The number of running threads for a given process. A thread is the basic unit of CPU utilization, and consists of a program counter, a register set, and a stack space. A thread is also called a lightweight process.
- Physical Memory Used - The amount of physical memory in kilobytes being used by the specified process.
- Virtual Memory Used - The amount of virtual memory in kilobytes being used by the specified process, or the size of the process in real memory plus swap.

Specify the process by its command name or process ID. (PID). Entering a PID overrides the entry of a command name. If no command name or PID is entered, the error Command not found is displayed and the probe is set to a CRITICAL state.

Table A.71. Red Hat Satellite::Process Health settings

Field	Value
Command Name	
Process ID (PID) file	
Timeout*	15
Critical Maximum CPU Usage	
Warning Maximum CPU Usage	
Critical Maximum Child Process Groups	
Warning Maximum Child Process Groups	
Critical Maximum Threads	
Warning Maximum Threads	
Critical Maximum Physical Memory Used	
Warning Maximum Physical Memory Used	
Critical Maximum Virtual Memory Used	

Field	Value
Warning Maximum Virtual Memory Used	

A.10.10. Red Hat Satellite::Process Running

The Red Hat Satellite::Process Running probe verifies that the specified process is running. Specify the process by its command name or process ID. (PID). Entering a PID overrides the entry of a command name. A Critical status results if the probe cannot verify the command or PID.

Table A.72. Red Hat Satellite::Process Running settings

Field	Value
Command Name	
Process ID (PID) file	
Critical Number Running Maximum	
Critical Number Running Minimum	

A.10.11. Red Hat Satellite::Swap

The Red Hat Satellite::Swap probe monitors the percent of free swap space available on a Satellite. A CRITICAL status results if the value falls below the Critical threshold. A WARNING status results if the value falls below the Warning threshold.

Table A.73. Red Hat Satellite::Swap settings

Field	Value
Critical Minimum Swap Percent Free	
Warning Minimum Swap Percent Free	

A.10.12. Red Hat Satellite::Users

The Red Hat Satellite::Users probe monitors the number of users currently logged into a Satellite. A CRITICAL status results if the value exceeds the Critical threshold. A WARNING status results if the value exceeds the Warning threshold.

Table A.74. Red Hat Satellite::Users settings

Field	Value
Critical Maximum Users	

Field	Value
Warning Maximum Users	

APPENDIX B. REVISION HISTORY

Revision 3-35 **Wed Jul 26 2017** **Satellite Documentation Team**
BZ#1195533 - Added information on spacecmd.

Revision 3-34 **Thu Aug 20 2015** **Dan Macpherson**
Mass publication of all Satellite 5.7 books

Revision 3-33 **Tue Aug 11 2015** **Dan Macpherson**
Fixing database service name for backup

Revision 3-32 **Wed May 27 2015** **Dan Macpherson**
Minor Revisions

Revision 3-31 **Fri Apr 10 2015** **Dan Macpherson**
Adding information on spacewalk-final-archive

Revision 3-30 **Mon Mar 2 2015** **Dan Macpherson**
Added spacecmd section

Revision 3-29 **Tue Feb 17 2015** **Dan Macpherson**
Updating rhnsd and osad commands to include RHEL 7 (BZ#1190249)
Fixed bytes/kilobytes specification for certain parameters. Also included rhn_java.conf (BZ#1189283)

Revision 3-28 **Tue Feb 3 2015** **Dan Macpherson**
Pushing maintenance update for Satellite 5.7

Revision 3-27 **Wed Jan 7 2015** **Dan Macpherson**
Packaging snapshot versions

Revision 3-26 **Thu Jan 1 2015** **Dan Macpherson**
Release Candidate for Satellite 5.7

Revision 3-25 **Mon Dec 8 2014** **Dan Macpherson**
Preparing books for technical review

Revision 3-24 **Thu Nov 20 2014** **Megan Lewis**
BZ#1142723 Added details about the default value of max file edit size.
BZ#1024490 Added note clarifying when users will be logged.
BZ#1153335 Added section on Setting Up a Self Subscribed Red Hat Satellite.

Revision 3-23 **Wed Nov 19 2014** **Megan Lewis**
BZ#1124939 Corrected location of configuration files.

Revision 3-22 **Sun Nov 16 2014** **Megan Lewis**
BZ#1043837 Added procedure for changing parts of web interface.
Updated author group.
Updated headings to match standards.
Implemented brand changes.

Revision 3-21 **Wed Oct 8 2014** **Megan Lewis**
Moved in several chapters from the Reference Guide.

Revision 3-20 **Fri Sep 27 2013** **Dan Macpherson**
Final version of documentation suite

Revision 3-19 **Wed Sep 11 2013** **Dan Macpherson**

Minor change

Revision 3-18 **Wed Sep 11 2013** **Dan Macpherson**
Modified tablespaces section to align with Satellite 5.6 requirements

Revision 3-17 **Wed Sep 11 2013** **Dan Macpherson**
Removing old screenshots

Revision 3-16 **Tue Sep 10 2013** **Dan Macpherson**
Revised Subtitle, Abstract and Preface for all Guides

Revision 3-15 **Thu Aug 29 2013** **Dan Macpherson**
First implementation of QE Review feedback

Revision 3-14 **Tues Aug 20 2013** **Megan Lewis**
Corrections for BZ#990387, BZ#990393, BZ#990398, BZ#990400, and BZ#990383

Revision 3-13 **Mon Jul 29 2013** **Dan Macpherson**
Adding Software Failure chapter based upon tech review feedback

Revision 3-12 **Sun Jul 28 2013** **Dan Macpherson**
Second implementation of tech review feedback

Revision 3-11 **Wed Jul 24 2013** **Dan Macpherson**
Corrections for BZ#987245

Revision 3-10 **Tue Jul 23 2013** **Dan Macpherson**
First implementation of tech review feedback

Revision 3-9 **Fri Jul 12 2013** **Dan Macpherson**
Final beta updates

Revision 3-8 **Fri Jul 12 2013** **Dan Macpherson**
Updates to beta packages

Revision 3-6 **Thu Jul 11 2013** **David O'Brien**
Move chapter on creating RPM files to Reference Guide.
Update section on failover. Include information about online backup and restore.
Review section on OpenSCAP.
Add section on scheduling administrative tasks.
Add section on disaster recovery.
Update section on cloning software channels.

Revision 3-5 **Wed Sep 19 2012** **Dan Macpherson**
Final packaging for 5.5

Revision 3-4 **Fri Aug 31 2012** **Athene Chan**
BZ#839798 Minor edit

Revision 3-3 **Fri Aug 24 2012** **Athene Chan**
BZ#839798 Changed 4.3 example to a standard format

Revision 3-3 **Fri Aug 24 2012** **Athene Chan**
BZ#839798 Changed 4.3 example to a standard format

Revision 3-2 **Fri Aug 24 2012** **Athene Chan**

BZ#826501 QA-reviewed changes applied.
BZ#884313 QA-reviewed changes applied.

Revision 3-1 **Fri Aug 17 2012** **Athene Chan**
BZ#848313 OpenSCAP chapter "How to View SCAP Results" added

Revision 3-0 **Thu Aug 9 2012** **Athene Chan**
Staging for Review

Revision 2-5 **Wed Aug 1 2012** **Athene Chan**
BZ#839798 Added spacewalk-clone-by-date chapter
BZ#826501 New OpenSCAP information added

Revision 2-0 **Fri Jul 6 2012** **Athene Chan**
Prepared for RHN Satellite 5.5 release
BZ#826501 Added OpenSCAP Chapter
OpenSCAP Screenshots added

Revision 1-5 **Mon Aug 15 2011** **Lana Brindley**
Folded z-stream release into y-stream

Revision 1-4 **Mon Jun 20 2011** **Lana Brindley**
BZ#701900 - PAM Authentication

Revision 1-3 **Mon Jun 20 2011** **Lana Brindley**
BZ#714029 - Fixed color in image

Revision 1-2 **Wed Jun 15 2011** **Lana Brindley**
Prepared for publication

Revision 1-1 **Fri May 27 2011** **Lana Brindley**
Updates from translators

Revision 1-0 **Fri May 6 2011** **Lana Brindley**
Prepare for translation

Revision 0-15 **Thu May 5 2011** **Lana Brindley**
BZ#701818 - QE Review

Revision 0-14 **Mon May 2 2011** **Lana Brindley**
BZ#248465 - QE Review

Revision 0-13 **Fri Apr 29 2011** **Lana Brindley**
BZ#692295 - QE Review

Revision 0-12 **Mon Apr 18 2011** **Lana Brindley**
BZ#691985 - Updating image

Revision 0-11 **Mon Apr 18 2011** **Lana Brindley**
BZ#691990 - QE Review

Revision 0-10 **Mon Apr 18 2011** **Lana Brindley**
BZ#691985 - QE Review

Revision 0-9 **Thu Apr 14 2011** **Lana Brindley**
Technical review feedback

Revision 0-8 **Wed Apr 13 2011** **Lana Brindley**

BZ#692314 - QE Review
BZ#692294 - QE Review
BZ#692291 - QE Review
BZ#692290 - QE Review
BZ#691988 - QE Review
BZ#691986 - QE Review
BZ#691981 - QE Review

Revision 0-7 **Wed Mar 23 2011** **Lana Brindley**
Preparation for technical review

Revision 0-6 **Mon Feb 21 2011** **Lana Brindley**
RPMs
Boot Devices
Organizations

Revision 0-5 **Fri Feb 18 2011** **Lana Brindley**
Monitoring
PAM Authentication

Revision 0-4 **Mon Jan 10 2011** **Lana Brindley**
Backup and Restore

Revision 0-3 **Fri Jan 7 2011** **Lana Brindley**
User Administration
Preface
Automatic Synchronization

Revision 0-2 **Wed Jan 5 2011** **Lana Brindley**
User Administration

Revision 0-1 **Tue Jan 4 2011** **Lana Brindley**
Completed new chapter structure

Revision 0-0 **Tue Dec 21 2010** **Lana Brindley**
New document creation from original RHN Satellite Deployment Guide