



Red Hat Network Satellite 5.5

スタートガイド

Red Hat Network Satellite を使用したプロビジョニングとデプロイメント
エディション 2

Red Hat Network Satellite 5.5 スタートガイド

Red Hat Network Satellite を使用したプロビジョニングとデプロイメント
エディション 2

Red Hat Documentation Team

法律上の通知

Copyright © 2011 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

本書では、Red Hat Network Satellite に実装されているキックスタートプロビジョニング機能に関する情報と使用方法を説明します。Satellite の基礎についての更に詳しい情報は、Satellite ユーザーガイド をご参照ください。

目次

第1章 はじめに	3
第2章 キックスタート	5
2.1. 必要なパッケージ	5
2.2. キックスタートツリー	5
2.3. キックスタートのプロファイル	7
2.4. テンプレーティング	11
2.5. マシンのキックスタート	14
第3章 複数の SATELLITE	19
3.1. INTER-SATELLITE SYNCHRONIZATION	19
3.2. 組織別の同期	20
3.3. ISS ユースケース	21
第4章 高度な API メソッドおよびコマンド	24
4.1. XML-RPC API	24
4.2. COBBLER	25
4.3. KOAN	29
第5章 トラブルシューティング	31
5.1. Web インターフェース	31
5.2. Anaconda	31
5.3. トレースバック	33
5.4. 登録	33
5.5. キックスタートとスニペット	34
付録A 改訂履歴	36

第1章 はじめに

プロビジョニングとは、物理もしくは仮想マシンを定義済みの既知の状態に設定するプロセスのことです。**Red Hat Network (RHN) Satellite** は、キックスタートプロセスを使ってシステムを準備します。プロビジョニングの機能を使うには、1 台もしくは複数台の ターゲット マシンが必要になります。ターゲットマシンは、物理マシンかベアメタルシステム、仮想マシンのいずれかになります。**RHN Satellite** の仮想マシンプロビジョニング機能を使用するには、**Xen** または **KVM** を使って仮想マシンを作成します。

定義

本書全体にわたって使用されている用語

キックスタート

ユーザーの介入をほとんどあるいは全く必要としない自動化された方法で **Red Hat** のシステムをインストールするプロセス。厳密に言えば、キックスタートとは、コンテンツの簡潔な記述とマシンの設定をインストーラーに提供できるようにし、それに基づいて動作を実行する、**Anaconda** インストールプログラムのメカニズムを指します。この簡潔なシステムの定義は、キックスタートプロファイル (*Kickstart profile*) と呼ばれています。

キックスタートプロファイル

キックスタートファイルは、マシンのキックスタートに必要なすべてのオプションを記載した、テキスト形式のファイルです。これには、パーティショニング情報、ネットワーク設定、インストールするパッケージが含まれます。**Satellite** の実装は、**Cobbler** のキックスタートへの機能強化をベースに構築されているため、**RHN Satellite** ではキックスタートプロファイルは従来の **Anaconda** キックスタートの定義のスーパーセットとなります。キックスタートプロファイルはキックスタートツリーが存在することを前提としています。

キックスタートツリー

マシンをキックスタートするのに必要なソフトウェアとサポートファイル。これは、「インストールツリー」とも呼ばれることもあります。通常、これは、特定のリリースで出荷されたインストールメディアから取り出したディレクトリ構造とファイルです。**Cobbler** の用語では、キックスタートツリーはディストリビューションの一部となっています。

PXE (Preboot eXecution Environment)

ターゲットマシン自体の事前設定なしで、電源投入時にベアメタルマシン (通常は物理マシンまたは実機) のキックスタートを可能にする低レベルのプロトコル。**PXE** は **DHCP** サーバーに依存して、ブートストラップサーバーに関する情報 (本書においては、**Satellite 5.5** 以降のバージョンのインストール) をクライアントに提供します。**PXE** を使用するには、ターゲットマシンのファームウェアでサポートされている必要があります。**PXE** は新しい物理マシンのブートや、**Satellite** に未登録のマシンの再インストールに非常に役立ちますが、**PXE** なしで仮想化を使用したり、**Satellite** の機能を再インストールすることも可能です。

プロビジョニングシナリオ

RHN Satellite がサポートするプロビジョニングシナリオのタイプ

新規インストール

オペレーティングシステムが未インストールのシステム (別名: ベアメタルインストール) のプロビジョニング

仮想インストール

Satellite は、KVM、Xen 完全仮想化ゲスト、および Xen 準仮想化ゲストをサポートしています。

再プロビジョニング

物理およびゲストシステムは、いずれも、再プロビジョニングが可能です。ただし、同一の **Satellite** インスタンスに登録されていることを条件とします。「[再プロビジョニング](#)」をご参照ください。

第2章 キックスター

2.1. 必要なパッケージ

システムがカスタムディストリビューションを使用している場合には、以下のパッケージが必要となります。これらは、**rhn-tools** Red Hat Network チャンネルから入手できます。

- **koan**
- **spacewalk-koan**

ご使用のカスタムチャンネルからこれらのパッケージにアクセスするには、既存の **rhn-tools** チャンネルのクローンを作成することをお勧めします。

RHN Satellite は、**kernel** と **initrd** のファイルがキックスターツリー内の特定の場所にあることを想定しています。ただし、これらの場所は、アーキテクチャによって異なります。以下の表は、それらの異なるロケーションについてまとめたものです。

表2.1 アーキテクチャ別の必要なディストリビューションファイル

アーキテクチャ	カーネル	初期 RAM ディスクイメージ
IBM System z	<i>TREE_PATH/images/kernel.img</i>	<i>TREE_PATH/images/initrd.img</i>
PowerPC	<i>TREE_PATH/ppc/ppc64/vmlinuz</i>	<i>TREE_PATH/images/pxeboot/vmlinux</i>
その他すべてのアーキテクチャ	<i>TREE_PATH/images/pxeboot/vmlinuz</i>	<i>TREE_PATH/images/pxeboot/initrd.img</i>

2.2. キックスターツリー

キックスタープロビジョニングを使用するには、ご使用の **Satellite** に最低でも 1 つのキックスターツリーがインストールされている必要があります。キックスターツリーのインストールは、自動または手動で行うことができます。

手順2.1 キックスターツリーの自動インストール

RHN にベースチャンネルがあるすべてのディストリビューションで、キックスターツリーを自動でインストールできます。これは、**satellite-sync** 介した通常のチャンネル同期の一環として行われます。

1. キックスターのベースとするディストリビューションを選択して、そのディストリビューションのベースチャンネルとそれに対応する RHN ツールのチャンネルを探します。

例えば、**x86** アーキテクチャを採用した **Red Hat Enterprise Linux 5** を使用したい場合、**rhel-i386-server-5** チャンネルとそれに対応する RHN Tools チャンネルである **rhn-tools-rhel-i386-server-5** が必要になります。

2. 接続された **Satellite** を使用している場合には、**satellite-sync** を使って **Satellite** サーバーを **Red Hat** サーバーと直接同期します。**Satellite** サーバーが接続されていない場合には、**Red Hat** サーバーから切断されたチャンネルダンプを取得して、それらと同期する必要があります。

3. チャンネルを同期すると、そのディストリビューション用の対応するキックスターツリーが自動的に作成されます。

手順2.2 キックスターツリーの手動インストール

カスタムディストリビューションは通常 Red Hat がサポートしていませんが、このようなディストリビューションや Red Hat Enterprise Linux のベータバージョンをキックスターツリーには、対応するキックスターツリーを手動で作成する必要があります。キックスターツリー用のディストリビューション用のインストール ISO が必要となります。

1. インストール ISO をご使用の Satellite Server にコピーして、`/mnt/iso` にマウントします。
2. ISO のコンテンツをカスタムのロケーションにコピーします。すべてのカスタムディストリビューションで、`/var/satellite` 内にディレクトリを作成することが推奨されます。例えば、RHEL のベータディストリビューションのコンテンツを `/var/satellite/custom-distro/rhel-i386-server-5.3-beta/` にコピーします。
3. RHN Satellite の Web インターフェースを使用して、カスタムソフトウェアチャンネルを作成します。チャンネル → ソフトウェアチャンネルの管理 → 新しいチャンネルの作成 で、適切な名前とラベルを付けて親チャンネルを作成します。上記で使用した例では、**rhel-5.3-beta** のラベルを使用します。
4. `rhnpush` コマンドを使用して、ソフトウェアパッケージをツリーのロケーションから新規作成されたソフトウェアチャンネルにプッシュします。

```
rhnpush --server=http://localhost/APP -c 'rhel-5.3-beta' \ -d  
/var/satellite/custom-distro/rhel-i386-server-5.3-beta/Server/
```

ご使用のディストリビューションによって、ツリー内のサブディレクトリが異なります。

5. ソフトウェアパッケージがプッシュされたら、`rm` コマンドを使用して、ツリーパス内で削除することができます。パッケージは、依然としてチャンネル内の Satellite サーバー上に格納され、ツリー内には必要なくなります。

```
rm /var/satellite/custom-distro/rhel-i386-server-5.3-beta/Server/*.rpm
```



注記

ソフトウェアパッケージをキックスターツリー内に残すように選択することもできます。これにより、後日、`yum` コマンドを使用して、随時インストールできるようにします。

6. RHN Satellite Web インターフェースを使用してディストリビューションを作成します。システム → キックスターツリー → ディストリビューション → 新規のディストリビューションを作成 に進み、適切なラベルとフルツリーパス (例: `/var/satellite/custom-distro/rhel-i386-server-5.3-beta/`) を使用して、ディストリビューションを作成します。あらかじめ作成したベースチャンネルと正しいインストーラ生成 (例: **Red Hat Enterprise Linux 5**) を選択します。作成を完了するには、**キックスターツリーディストリビューションの作成** を選択します。
7. 複数の環境とシステムにわたって同一ソフトウェアを維持するには、既存の Red Hat Enterprise Linux ベースチャンネルからの RHN Tools 子チャンネルを新たに作成したベースチャンネルの子チャンネルとしてクローンできます。子チャンネルのクローンが以下の手順で

行います。

1. RHN Satellite の Web インターフェースで、チャンネル → ソフトウェアチャンネルの管理 → チャンネルのクローン をクリックします。
2. **Clone From: (クローンする対象)** ドロップダウンボックスからクローンする子チャンネルとその状態を選択します。
3. **チャンネルの作成** をクリックします。
4. 必要な情報を記入し、クローンされた子チャンネルの上位になる親チャンネルを選びます。
5. **チャンネルの作成** をクリックします。

Create Kickstart Distribution

The following details are needed to define a kickstartable distribution. The tree path field should be a valid path to a installation tree located on this RHN Satellite server. The Kickstart RPM should be the name of the rpm containing the kickstart deploy scripts. These RPMs are provided by RHN Satellite and located in the RHN Satellite Tools child channels.

The Distribution Label field should contain only letters, numbers, hyphens, periods, and underscores. It must also be at least 4 characters long.

The Tree Path, Kickstart RPM, Base Channel, and Installer Generation should always match. This generally means that the versions for each field should be from the same version of Red Hat Enterprise Linux.

The Tree Path must be a local disk path on your RHN Satellite server containing the entire kickstart tree for a distribution include kernel, initrd, and repo information, but excluding any rpms. This directory should be readable by the apache and tomcat users. From within the specified tree path, a kernel should be available at `./images/pxeboot/vmlinuz` and an initrd image should be available at `./images/pxeboot/initrd.img`. For instance, if you have media located on the RHN Satellite server at: `/var/distro-trees/rhel-5-server/` you would specify that path as your Tree Path value which would check for a kernel and initrd here: `/var/distro-trees/rhel-5-server/images/pxeboot/`

Create Kickstart Distribution

Distribution Label*:

Tree Path*::

Base Channel*:

Installer Generation*:

Kernel Options:

Post Kernel Options:

図2.1 キックスタートディストリビューションの作成

2.3. キックスタートのプロファイル

キックスタートのプロファイルは、インストールで使用する設定オプションを指定します。

キックスタートプロファイルは、ウィザードインターフェースを使用して作成することができます。この方法では、一連の質問に対して答えた回答に基づいてプロファイルが生成されます。キックスタートプロファイルは、*raw* メソッドを使用して作成することもできます。この方法だと、プロファイルの内容を完全にコントロールすることができるようになります。

手順2.3 ウィザードを使用したキックスタートプロファイルの作成

1. システム → キックスタート → **新規のキックスタートプロファイルを作成** を選択します。
2. 適切な **ラベル** を提供し、希望する **ベースチャンネル** と **キックスタートツリー** を選択します。
3. 希望する **仮想化タイプ** を選択します。仮想化タイプについての詳しい情報は、[仮想化タイプ](#) を参照してください。 **次へ** をクリックして、続行します。
4. キックスタートプロファイルのダウンロードロケーションを選択します。カスタムディストリビューションを使用している場合には、そのツリーのロケーションを **URL (HTTP と FTP の両方をサポート)** として入力します。それ以外の場合は、デフォルトのオプションを使用します。 **次へ** をクリックして、続行します。
5. **root** のパスワードを入力して、**完了** をクリックし、プロファイルの作成を完了します。
6. 完全なキックスタートプロファイルが作成されます。このプロファイルは、**キックスタートファイル** をクリックすると表示することができます。

手順2.4 raw メソッドを使用したキックスタートプロファイルの作成

1. システム → キックスタート → **キックスタートファイルをアップロード** を選択します。
2. 適切な **ラベル** を提供し、希望する **ディストリビューション** を選択します。
3. 希望する **仮想化タイプ** を選択します。仮想化タイプについての詳しい情報は、[仮想化タイプ](#) を参照してください。
4. 既存のキックスタートファイルがある場合には、ファイルをアップロードします。そうでない場合には、**ファイルの内容** テキストボックスにキックスタートプロファイルを書き込みます。

スターティングポイントとして使用できる raw キックスタートの例は以下の通りです。

```
install
text
network --bootproto dhcp
url --url http://$http_server/ks/dist/org/1/ks-rhel-i386-server-5
lang en_US
keyboard us
zerombr
clearpart --all
part / --fstype=ext3 --size=200 --grow
part /boot --fstype=ext3 --size=200
part swap --size=1000 --maxsize=2000
bootloader --location mbr
timezone America/New_York
auth --enablemd5 --enablesshadow
rootpw --iscrypted $1$X/CrCfCE$x0veQ088TCm2VprcMkH.d0
selinux --permissive
reboot
firewall --disabled
skipx
key --skip

%packages
@ Base
```

```
%post
$SNIPPET('redhat_register')
```

5. RHN Satellite Server は、指定されたディストリビューションをキックスタート内の **url** として処理しないため、**url --url** オプションをプロファイルに記載する必要があります。以下は、その例です。

```
url --url http://satellite.example.com/ks/dist/org/1/my_distro
```

my_distro をディストリビューションラベルに、**1** をご使用の組織 ID に置き換えます。

6. raw キックスタートプロファイルは、Satellite のホスト名の代わりに、**\$http_server** を使用します。これは、キックスタートテンプレートがレンダリングされる際に自動的に記入されます。

7. **redhat_register** スニペットを使用して登録処理が行われます。

The screenshot shows the 'Create Kickstart Profile' page in the Satellite web interface. The top navigation bar includes links for Overview, Systems, Errata, Channels, Configuration, Schedule, Users, Admin, and Help. A red banner indicates '1 SYSTEM SELECTED' with 'MANAGE' and 'CLEAR' buttons. The left sidebar lists various system management tools. The main content area is titled 'Create Kickstart Profile' and contains the following sections:

- Kickstart Details:** A note stating that each kickstart file needs a label for later reference.
- Label*:** A text input field.
- Kickstartable Tree*:** A dropdown menu currently showing 'ks-rhel-i386-as-4'.
- Virtualization Type:** A dropdown menu currently showing 'None'. A note below explains that changing this type may require changes to the bootloader and partition options.
- File Contents:** A text editor area with a single line of text. Below the editor is a 'Toggle editor' checkbox.
- File to Upload:** A section with a 'Browse...' button and a 'Create' button.

At the bottom of the interface, there is a status bar showing '32.06 KB/s', '0.999', and 'Done'.

図2.2 raw キックスタート

仮想化タイプ

すべてのキックスタートプロファイルには、仮想化タイプが関連付けされます。以下の表に、様々なオプションを簡単にまとめました。

表2.2 仮想化タイプ

タイプ	説明	用途
なし	仮想化なし	このタイプは、Xen もしくは KVM 以外 (例 : VMware、Virtage など) の通常のプロビジョニング、ベアメタルインストール、仮想化インストールに使用します。
KVM 仮想化ゲスト	KVM ゲスト	このタイプは、KVM ゲストのプロビジョニングに使用します。
Xen 完全仮想化ゲスト	Xen ゲスト	<p>このタイプは、Xen ゲストのプロビジョニングに使用します。</p> <div>  <div> <p>注記</p> <p>このオプションには、ホスト上でのハードウェアサポートが必要ですが、ゲスト上では修正されたオペレーティングシステムは必要ありません。</p> </div> </div>
Xen 準仮想化ゲスト	Xen ゲスト	Xen 準仮想化を使用する仮想ゲストのプロビジョニングに使用します。準仮想化は、最速の仮想化モードです。これには、システム CPU 上の PAE フラグと修正されたオペレーティングシステムが必要です。Red Hat Enterprise Linux 5 は、準仮想化でのゲストをサポートしています。
Xen 仮想化ホスト	Xen ホスト	このタイプは、Xen 準仮想化を使用する仮想ホストのプロビジョニングに使用します。ハードウェアに互換性がある場合には、Xen 準仮想化のゲストとホストがサポートされます。

Xen ホストとして使用するために作成されたキックスタートプロファイルには、**%packages** セクションに **kernel-xen** パッケージが含まれている必要があります

KVM ホストとして使用するために作成されたキックスタートプロファイルには、**%packages** セクションに **qemu** パッケージが含まれている必要があります。

完全仮想化システムには、コンピューターの BIOS メニューで仮想化サポートがオンになっている必要がある場合があります。



注記

キックスタートに関する詳しい情報は、『Red Hat Enterprise Linux インストールガイド』の『キックスタートのインストール』の章をご参照ください。

2.4. テンプレーティング

キックスタートのテンプレーティングにより、ご使用のキックスタートファイル内に変数、スニペット、ならびに **for** ループや **if** ステートメントなどのフロー制御ステートメントを追加することができます。これは、**cheetah** ツールを使用して行うことができます。

テンプレーティングは、以下のような様々な理由で有用となります。

- 複数のキックスタート間のディスクのパーティショニングセクションなどの、キックスタートの特定のセクションを再利用することができます。
- 複数のキックスタート全体にわたって、一貫して **%post** の動作を実行することができます。
- DNS サーバー、Proxy サーバー、Web サーバーといった、複数の種類のサーバーのロール全体にわたってスニペットを定義することができます。例えば、Web サーバーには、以下のようなスニペットが定義されます。

```
httpd
mod_ssl
mod_python
```

Web サーバーのプロファイルを作成したい場合は、キックスタートファイルの **%package** セクションに Web サーバースニペットを追加します。プロファイルを Web サーバーと Proxy サーバーの両方にしたい場合は、パッケージセクションに両方のスニペットを記載します。Web サーバースニペットにもう一つのパッケージを追加したい場合 (例えば、**mod_perl** の場合) には、スニペットを更新すると、そのスニペットを使用しているすべてのプロファイルが動的に更新されます。

変数

テンプレーティングにより、キックスタートファイル全体で変数の定義を使用することができます。変数は、一つのレベルで設定し、それ以下のレベルでは上書きされる設定が可能な継承の対象となります。このため、変数がシステムレベルで定義されている場合には、この変数がプロファイルもしくはキックスタートツリーのレベルで定義された同一の変数に優先します。同様に、変数がプロファイルレベルで定義されている場合は、この変数がキックスタートツリーレベルで定義されている同一の変数に優先します。



注記

Satellite の同期時に作成されるキックスタートツリーなどの自動生成されるキックスタートツリーに対しては、キックスタートツリーの変数定義はできない点に注意してください。

スニペット

スニペットは、複数のキックスタートテンプレート間でコードの断片を再利用します。これらは、多くの行にまたがる可能性があり、その中に変数が含まれる場合もあります。スニペットは、**\$SNIPPET('snippet_name')** のテキストを使用することにより、キックスタートプロファイルに組み入れることができます。特定のパッケージ一覧や、特定の **%post** スクリプト、もしくはキックスタートファイルに通常含まれる任意のテキスト用にスニペットを作成することもできます。

スニペットを管理するには、システム → キックスタート → キックスタートスニペットに進みます。

キックスタートスニペット のページには、編集はできなくても、どの組織でも使用することができるデフォルトのスニペットがいくつか表示されます。デフォルトのスニペットは、RHN Satellite Server 上に書き込まれた、もしくは RHN Satellite Server にアップロードされたキックスタートで使用することができます。デフォルトのスニペットは、RHN Satellite サーバーのファイルシステムの `/var/lib/cobbler/snippets/` に格納されます。`/var/lib/rhn/kickstarts/wizard/` には、ウィザードスタイルのキックスタートからのテンプレートがあり、異なるデフォルトのスニペットとそれらの使用方法について説明しています。

redhat_register スニペットは、マシンをキックスタートの一部として Satellite Server に登録するために使用されるデフォルトのスニペットです。**redhat_management_key** と呼ばれる変数を使用して、マシンを登録します。このスニペットを使用するには、**redhat_management_key** の変数をシステム、プロファイルもしくはディストリビューションのいずれかのレベルで設定してから、キックスタートの `%post` セクションに `$SNIPPET('redhat_register')` を追加します。RHN Satellite Server が生成したウィザードスタイルのキックスタートにはいずれも、`%post` のセクションにこのスニペットが既に入っています。

カスタムスニペット タブでは、組織で使用するよう作成されたスニペットを表示・編集することができます。**新しいスニペットの作成** をクリックすると、新しいスニペットを作成することができます。カスタムスニペットは、`/var/lib/rhn/kickstarts/snippets/` ディレクトリに格納されます。RHN Satellite は、スニペットを組織別に異なるディレクトリに格納するため、カスタムスニペットは以下の例のようなファイル名で格納されます。`1` は組織 ID です。

```
$SNIPPET('spacewalk/1/snippet_name')
```

キックスタートにカスタムスニペットを挿入するために使用すべきテキストを判断するには、スニペット一覧もしくはスニペットの詳細ページで、**スニペットマクロ** の列を探してください。



注記

スニペットは、グローバルレベルで存在し、同一の継承構造を変数として共有しません。ただし、スニペット内の変数を使用して、異なるシステムがキックスタートを要求する時の動作のしかたを変更することができます。

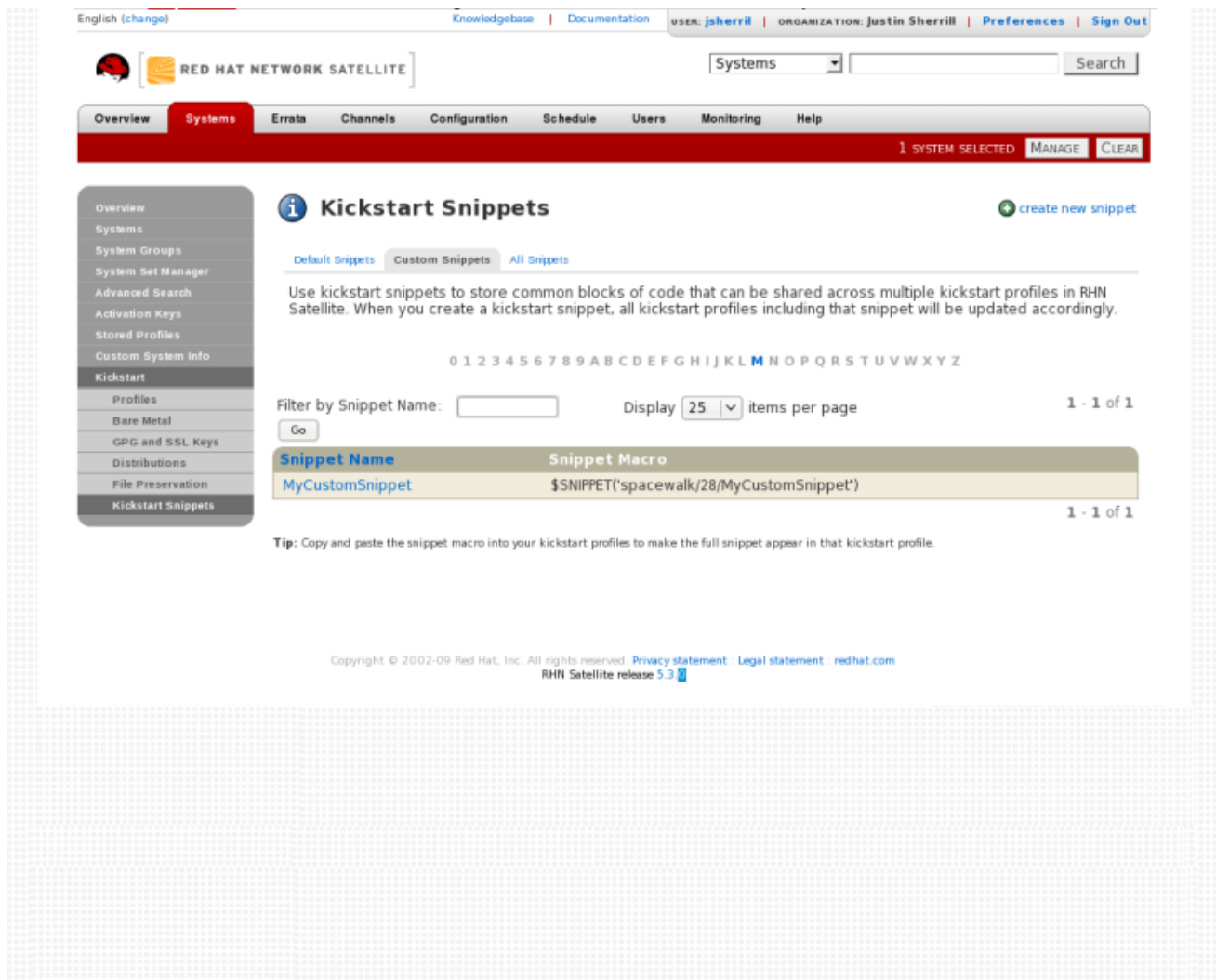


図2.3 キックスタートスニペット

特殊文字のエスケープ

\$と#の文字は、テンプレートの実行中に変数や制御フローを指定するために使用されます。スクリプト内で他の目的でこれらの文字が必要な場合には、これらを実際文字として認識されないようにする必要があります。これには、いくつかの方法があります。

- テンプレートの実行中に無視したい\$または#の各インスタンスの前にバックスラッシュ文字(\)を配置します。
- スクリプト全体を#raw ... #end raw内にラップします。

ウィザードスタイルのキックスタートを使用して作成された%preおよび%postのスクリプトはすべて、デフォルトでは#raw...#end rawでラップされます。これは、%postまたは%preスクリプトを編集する際に利用可能なテンプレートチェックボックスを使用して切り替えることができます。

- スニペットの最初の行に#errorCatcher Echoを追加します。

例2.1 テンプレート内の特殊文字のエスケープ

この例では、キックスタートテンプレート内での特殊文字のエスケープ方法について説明します。

以下のbashスクリプトは、%post セクション内に挿入する必要があります。

```
%post
echo $foo > /tmp/foo.txt
```

\$ をエスケープしないと、テンプレートエンジンは **\$foo** という名前の変数を探そうとしますが、**foo** は変数としては存在しないため、失敗してしまいます。

\$ をエスケープするもっとも簡単な方法は、バックスラッシュ文字 (****) を使用する方法です。

```
%post
echo \$foo > /tmp/foo.txt
```

これにより、**\\$foo** が **\$foo** としてレンダリングされます。

第 2 の方法は、以下のように、**bash** スクリプト全体を **#raw ... #end raw** 内にラップする方法です。

```
%post
#raw
echo $foo > /tmp/foo.txt
#end raw
```

最後の方法は、キックスターテンプレートの 1 行目に **#errorCatcher Echo** を追加する方法です。これは、テンプレートエンジンに対して、存在しない変数はいずれも無視して、テキストを現状通りに出力するように指示します。このオプションは、ウィザードスタイルのキックスターに既に含まれていますが、手動で作成する **raw** キックスターにも組み入れが可能です。

2.5. マシンのキックスター

2.5.1. ベアメタルからのキックスター

オペレーティングシステムがインストールされていないマシンや間違ったオペレーティングシステムがインストールされているマシンは、ベアメタルマシンと呼ばれます。ベアメタルからマシンをプロビジョニングする方法は 3 つあります。

- 標準のオペレーティングシステムのインストールメディア
- PXE ブート

手順 2.5 インストールメディアからのブート

1. インストールメディアをマシンに挿入します。このメディアは使用するキックスターと適合している必要があります。例えば、キックスターが **ks-rhel-i386-server-5-u2** キックスターツリーを使用するように構成されている場合には、Red Hat Enterprise Linux 5.2 i386 インストールメディアを使用します。
2. ブートプロンプトが表示されたら、以下のコマンドでキックスターをアクティブにします。

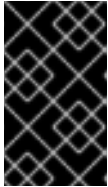
```
linux ks=http://satellite.example.com/path/to/kickstart
```

3. システムが起動したら、キックスターファイルをダウンロードして、自動的にインストールします。

手順2.6 PXE ブート

PXE ブートを実行するには、各システムが BIOS レベルで PXE ブートをサポートしている必要があります。最新のハードウェアの大部分がこれをサポートしています。また、インストール後にシステムが静的に設定される場合でも、DHCP サーバーを必要とします。

1.



重要

ネットワーク上の別のシステムに DHCP サーバーがデプロイされている場合、DHCP の設定ファイル編集には、DHCP サーバーへの管理アクセスが必要となります。

マシンが複数のネットワーク上にある場合は、すべてのマシンが DHCP サーバーに接続できることを確認してください。これは、DHCP サーバーのマルチホーミングを行い (リアルまたはトランク VLAN を使用)、すべてのルータまたはスイッチがネットワーク境界を越えて DHCP プロトコルをわたすように設定することで可能になります。

RHN Satellite が管理するシステムの **next-server** アドレスを設定することで、DHCP サーバーが PXE サーバーに向くように構成します。

インストール中にホスト名を使用するには、以下の行を追加して DHCP サーバーがドメインと IP アドレスに向くように設定します。

```
option domain-name DOMAIN_NAME;
option domain-name-servers IP_ADDRESS1, IP_ADDRESS2;
```

2. DHCP サーバー上で、root ユーザーに切り替え、**/etc/dhcpd.conf** ファイルを開きます。PXE ブートインストールを実行するオプションを伴う新たなクラスを追加します。

```
allow booting;
allow bootp;
class "PXE" {
    match if substring(option vendor-class-identifier, 0, 9) =
"PXEclient";
    next-server 192.168.2.1;
    filename "pxelinux.0";
}
```

このクラスは、以下のような動作を実行します。

1. **bootp** プロトコルによるネットワークブートを有効化します。
 2. **PXE** と呼ばれるクラスを作成します。ブートの優先順位で **PXE** が第1位に設定されているシステムの場合は、**PXEclient** として自動的に認識します。
 3. DHCP サーバーは、**192.168.2.1** の IP アドレスの **Cobbler** サーバーにシステムをダイレクトします。
 4. DHCP サーバーは、**/var/lib/tftpboot/pxelinux.0** にあるブートイメージファイルを参照します。
3. **Xinetd** を設定します。**Xinetd** は、サービスのスイートを管理するデーモンです。これには、ブートイメージを PXE クライアントに転送するための FTP サーバーである、**TFTP** が含まれます。

chkconfig コマンドを使用して **Xinetd** を有効にします。

```
chkconfig xinetd on
```

もう一つの方法としては、**root** ユーザーに切り替えて、**/etc/xinetd.d/tftp** ファイルを開き、**disable = yes** の行を **disable = no** に変更する方法があります。

4. **Xinetd** サービスを起動し、**TFTP** が **pxelinux.0** ブートイメージに対してサービスを提供開始できるようにします

```
chkconfig --level 345 xinetd on
/sbin/service xinetd start
```

chkconfig コマンドは、すべてのユーザーランレベルに対して **xinetd** サービスを有効化する一方、**/sbin/service** コマンドは、**xinetd** を即時に有効化します。

2.5.2. 再プロビジョニング

既存のシステムを再インストールする作業は、**再プロビジョニング**と呼ばれています。再プロビジョニングは **RHN Satellite** の **Web** インターフェースを使用して行うことができます。システムは、再プロビジョニングが行われる前と同じシステムプロファイルを使用します。これによって、システムの情報やセッティングの多くが維持されることになります。

再プロビジョニングは、システムを表示して、**Provisioning** のタブでスケジュールすることができます。追加のオプションを設定するには、**高度な設定** のページに進みます。このページ上で、カーネルのオプションやネットワーク情報、パッケージプロファイルの同期などの詳細設定を行うことができます。**カーネルオプション** のセクションは、キックスタート時に使用されるカーネルオプションへのアクセスを提供します。**カーネルの後のオプション** は、キックスタートの完了後のシステムの初回起動時に使用されるカーネルオプションです。

例2.2 カーネルオプションとポスターカーネルオプションの設定

この例では、再プロビジョニングプロセスにおけるカーネルオプションとカーネルの後のオプションの相異点を示しています。

キックスタートをリモートで監視するために **VNC** 接続を確立したい場合には、**Kernel Options** の行に **vnc vncpassword=PASSWORD** を追加します。

対象となるシステムのカーネルを **noapic** のカーネルオプションで起動させたい場合は、**Post Kernel Options** の行に **noapic** を追加します。

手順2.7 ファイル保持

ファイル保持の機能を使用して、再プロビジョニング中にファイルが紛失するのを防ぐことができます。この機能は、キックスタート中にファイルを一時的に保管して、再プロビジョニングが完了した後に復元します。



注記

ファイル保持一覧を使用できるのは、ウィザードスタイルのキックスタートに限定されており、再プロビジョニング中のみとなっています。

1. システム → キックスタート → **ファイル保持** → **新規のファイル保持一覧を作成** へと進み、保持するファイルの一覧を作成します。
2. システム → キックスタート → **プロフィール** に進み、希望のプロフィールを選択して、ファイル保持リストをキックスタートに関連付けします。
3. システムの詳細 → **ファイル保持** に進み、ファイル保持リストを選択します。

2.5.3. 仮想化ゲストのプロビジョニング

RHN Satellite 5.5 では、以下の仮想化技術を使用して仮想ゲストのプロビジョニングをサポートしています。

- KVM 仮想化ゲスト
- Xen 完全仮想化ゲスト
- Xen 準仮想化ゲスト

手順2.8 仮想化されたゲストのプロビジョニング

1. ホストシステムに **仮想化** または **仮想化プラットフォーム** のシステムエンタイトルメントがあることを確認します。
2. システム のページで、適切な仮想ホストを選択してから、**仮想化** → **Provisioning** を選択します。適切なキックスタートプロファイルを選択して、ゲスト名を入力します。
3. ゲストのメモリーや CPU の使用量などのパラメータを追加で設定したい場合には、**高度な設定** のボタンをクリックします。以下の項目を設定することができます。
 - ネットワーク: 静的または DHCP
 - カーネルオプション
 - パッケージプロファイルの同期: キックスタートの終了時にシステムが、そのパッケージプロファイルを別のシステムや保管されたプロファイルと同期。
 - メモリの割り当て: RAM (デフォルト値は 512MB)
 - 仮想ディスクのサイズ
 - 仮想 CPU (デフォルト値は 1)
 - 仮想ブリッジ: インストールに使用されるネットワーキングブリッジ。Xen プロビジョニングには **xenbr0**、KVM には **virbr0** がデフォルト値となっています。



注記

virbr0 ネットワーキングブリッジは、外部ネットワーキングを許可しません。外部ネットワーキングが必要な場合には、代わりに、ホストが実際のブリッジを作成するように設定してください。ただし、**xenbr0** は実際のブリッジであり、可能な場合にはこれを使用することが推奨されます。

- 仮想ストレージパス: ゲストのディスク情報を保管するファイル、LVM 論理ボリューム、ディレクトリ、ブロックデバイスへのパス。これに

は、`/dev/sdb`、`/dev/LogVol00/mydisk`、`VolGroup00`、`/var/lib/xen/images/myDisk`などが含まれます。

4. キックスタートをスケジュールしてから終了する をクリックします。

2.5.4. RHN Proxy を介したプロビジョニング

プロビジョニングは、既にインストールされ、RHN Satellite に登録済みの RHN Proxy を使用して行うこともできます。

1. 仮想ゲストのプロビジョニングを行う時、もしくはシステムの再プロビジョニングを行う場合には、**RHN Satellite Proxy を選択** のドロップダウンメニューから希望の Proxy を選択してください。
2. ベアメタルインストールの場合には、RHN Satellite の完全修飾ドメイン名 (FQDN) を Proxy の FQDN に置き換えます。例えば、キックスタートファイルへの URL が以下のような場合、

```
http://satellite.example.com/ks/cfg/org/1/label/myprofile
```

Proxy を介したキックスタートを行うには、以下の URL を使用します。

```
http://proxy.example.com/ks/cfg/org/1/label/myprofile
```

第3章 複数の SATELLITE

Inter-satellite synchronization (ISS) は、複数の *Satellite* 間でのコンテンツ連携を可能にします。この機能は、組織のニーズに応じていくつか異なる方法で使用できます。本章では、ユースケースと組織に適した ISS の最善の設定方法について説明します。

ISS の要件

ISS 使用の必要条件は以下のとおりです。

- 2 台以上の RHN *Satellite* サーバー
- 1 つ以上のチャンネルが設定された 1 つ以上の RHN *Satellite*
- セキュアな接続のために、各スレーブ RHN *Satellite* にマスター RHN *Satellite* SSL 証明書が必要となります。

3.1. INTER-SATELLITE SYNCHRONIZATION

手順3.1 マスターサーバーの設定

別の *Satellite* に同期されるファイルを決定するには、マスターサーバーが使用されます。

1. *Inter-Satellite Synchronization (ISS)* 機能を有効化し、`/etc/rhn/rhn.conf` ファイルを開いて、以下の行を追加するか、同じとなるよう修正します。

```
disable_iss=0
```

2. `/etc/rhn/rhn.conf` ファイル内で、`allowed_iss_slaves=` の行を見つけます。デフォルトでは、同期にスレーブ *Satellite* は指定されていません。各スレーブ *Satellite* サーバーのホスト名をコンマ区切りで入力してください。

```
allowed_iss_slaves=slave1.satellite.example.org,slave2.satellite.example.org
```

3. 設定ファイルを保存して、`httpd` サービスを再起動します。

```
service httpd restart
```

手順3.2 スレーブサーバーの設定

スレーブ *Satellite* サーバーとは、マスターサーバーにコンテンツが同期されるマシンです。

1. コンテンツをスレーブサーバーに安全に転送するには、マスターサーバーからの **ORG-SSL** 証明書が必要となります。この証明書は HTTP を介して任意の *Satellite* の `/pub/` ディレクトリからダウンロードできます。ファイル名は **RHN-ORG-TRUSTED-SSL-CERT** ですが、変更することができ、スレーブのローカルファイルシステム内の任意の場所 (例: `/usr/share/rhn/` ディレクトリなど) に置くことができます。
2. マスターサーバーからの同期が可能なチャンネル一覧を表示するには、以下のコマンドを実行します。これにより、正式な Red Hat チャンネルと、利用可能なカスタムチャンネルが表示されます。

-

```
satellite-sync --iss-parent=master.satellite.example.com --ca-
cert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT --list-channels
```

master.satellite.example.com の箇所は、マスターサーバーのホスト名に置き換えてください。

手順3.3 Inter-Satellite Synchronizationの実行

マスターサーバーとスレーブサーバーの設定が完了したら、それらの間での同期が可能になります。

1. スレーブサーバー上で `/etc/rhn/rhn.conf` ファイルをテキストエディタで開き、マスターサーバーのホスト名と SSL 証明書ファイルのパス詳細を追加します。

```
iss_parent      = master.satellite.example.com
iss_ca_chain    = /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
```

2. **satellite-sync** のコマンドを以下のように実行して、同期を開始します。

```
satellite-sync -c your-channel
```



注記

satellite-sync コマンドで手動で提供されるコマンドラインオプションは、`/etc/rhn/rhn.conf` ファイル内のカスタム設定を上書きします。

3.2. 組織別の同期

ISS は、特定の組織にコンテンツをインポートするためにも使用できます。これはローカルで、もしくはリモート同期を使用して行うことができます。この機能は、チャンネルダンプを介して、もしくは接続された **Satellite** からエクスポートして切断モードの **Satellite** にインポートする方法でコンテンツを取得している、複数の組織を伴う切断モードの **Satellite** に便利です。組織別の同期は、カスタムチャンネルを接続モードの **Satellite** からエクスポートするのに使用できます。また、組織の同期を使用すると、複数の組織間でのコンテンツの移行も効果的に行うことができます。

組織別の同期は、ソース組織の整合性を維持するために、一連の明確なルールに従っています。

- ソースコンテンツが **NULL** 組織に属する場合は (すなわち **Red Hat** のコンテンツ)、同期先の組織が指定されている場合であってもデフォルトで **NULL** 組織に設定されます。これにより、指定されたコンテンツは、常に権限がある **NULL** 組織にあることになります。
- コマンドラインで組織を指定した場合は、コンテンツがその組織からインポートされます。
- 組織が指定されていない場合は、デフォルトで組織 1 に指定されます。

以下に 組織 ID (**orgid**) を使用した **Satellite** 間同期の 3 つのシナリオ例を示します。

例3.1 マスター **Satellite** からスレーブ **Satellite** へのコンテンツのインポート

この例では、マスター **Satellite** からスレーブ **Satellite** にコンテンツをインポートします。

```
satellite-sync --parent-sat=master.satellite.example.com -c channel-name
--orgid=2
```


例3.2 特定の組織のエクスポートされたダンプからのコンテンツのインポート

この例では、特定の組織のエクスポートされたダンプからコンテンツをインポートします。

```
$ satellite-sync -m /dump -c channel-name --orgid=2
```

例3.3 Red Hat Network Hosted からのコンテンツのインポート

この例では、Red Hat Network Hosted からコンテンツをインポートします (システムが登録および有効化されている前提)。

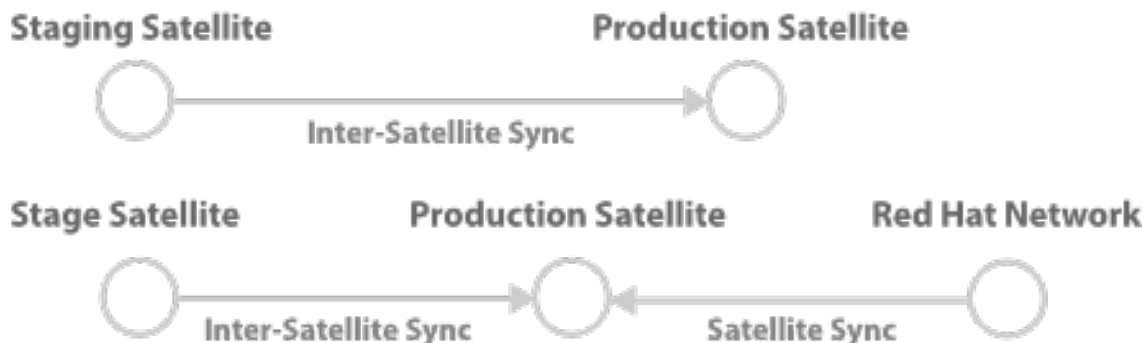
```
$ satellite-sync -c channel-name
```

3.3. ISS ユースケース

ISS は、組織のニーズに応じていくつか異なる方法で使用することができます。このセクションでは、ISS の使用方法と、これらのケースを設定/操作する方法を説明します。

例3.4 ステージング Satellite

この例では、**Satellite** を **ステージング Satellite** として使用して、コンテンツの準備とパッケージの品質保証作業を行い、実稼働での使用に適切であることを確認します。コンテンツの実稼働での使用が承認されると、実稼働 **Satellite** はステージング **Satellite** からコンテンツを同期します。



1. **satellite-sync** コマンドを実行して、**rhn_parent** (通常は Red Hat Network Hosted) とデータを同期します。

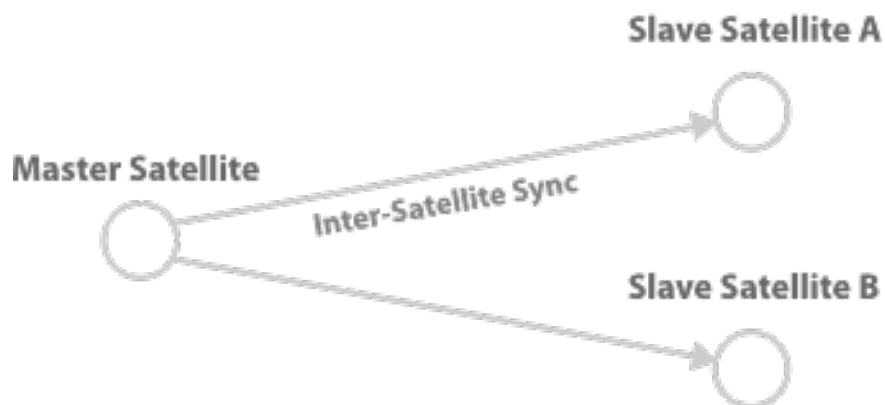
```
satellite-sync -c your-channel
```

2. 以下のコマンドを実行して、ステージングサーバーからデータを同期します。

```
satellite-sync --iss-parent=staging-satellite.example.com -c custom-channel
```

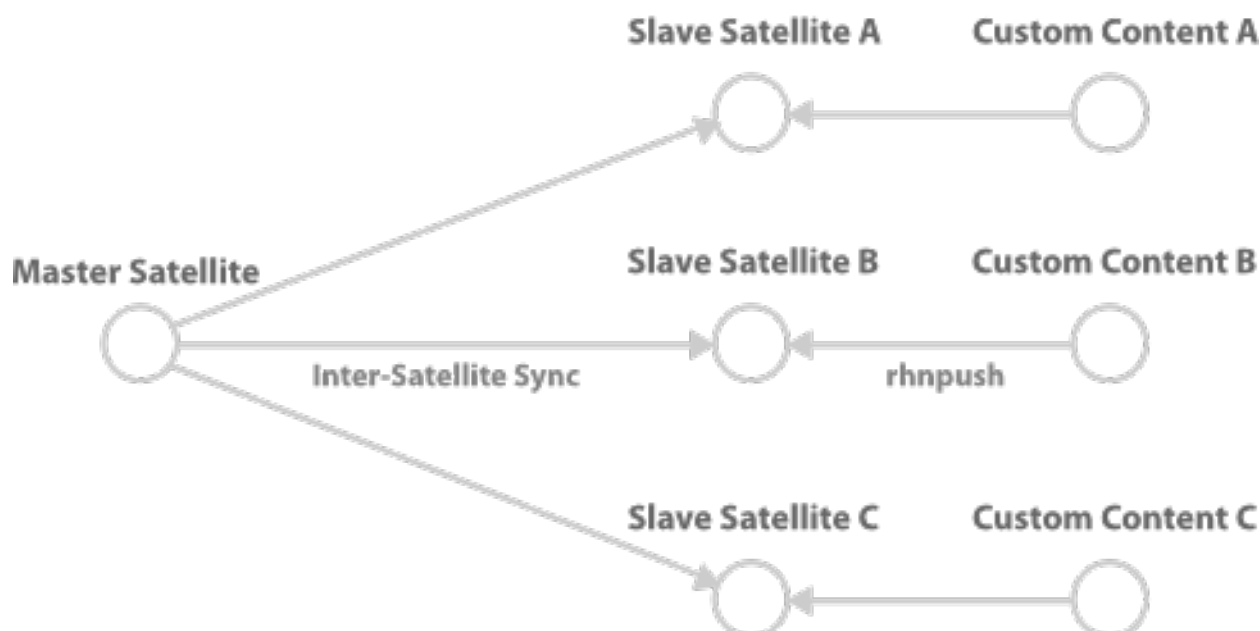
例3.5 同期されたスレーブ

この例では、マスター **Satellite** がスレーブに直接データを提供し、変更は定期的に同期されます。



例3.6 スレーブのカスタムコンテンツ

この例では、マスター **Satellite** を開発チャンネルとして使用します。コンテンツはこのチャンネルからすべての実稼働スレーブ **Satellite** に配布されます。一部のスレーブ **Satellite** には、マスター **Satellite** チャンネルには存在しない追加のコンテンツが含まれています。これらのパッケージは維持されますが、マスター **Satellite** からのすべての変更はスレーブに同期されます。



例3.7 双方向同期

この環境では、2 台の **RHN Satellite** サーバーがお互いのマスターとして機能し、両サーバーの間でコンテンツを同期することができます。



1. 両方の **Satellite** が SSL 証明書を共有できることを確認します。
2. 第1 **Satellite** で、`/etc/rhn/rhn.conf` ファイルを開き、第2 **Satellite** のホストネームをポイントするように `iss_parent` オプションを設定します。

3. 第2 Satellite で、`/etc/rhn/rhn.conf` ファイルを開き、第1 Satellite のホストネームをポイントするように `iss_parent` オプションを設定します。

第4章 高度な API メソッドおよびコマンド

4.1. XML-RPC API

RHN Satellite 5.5 は、XML-RPC API を使用してプロビジョニングをサポートします。

以下のような API メソッドを使用して、キックスタープロファイルとツリーの維持管理を行います。

表4.1 XML-RPC メソッド

XML-RPC 名前空間	用途
kickstart	キックスタープロファイルの作成/インポート/削除。利用可能なキックスターツリーとプロファイルの一覧表示にも使用します。
kickstart.tree	キックスターツリーの作成/名前変更/更新/削除
kickstart.filepreservation	キックスタープロファイルに関連付けすることが可能なファイル保持一覧の一覧表示/作成/削除。 ファイル保持一覧を作成した後 に、 kickstart.profile.system.add_file_preservations API メソッドを呼び出すことによって、キックスタープロファイルと関連付けすることができます。
kickstart.keys	異なるキックスタープロファイルに関連付けすることが可能な暗号化キー (GPG/SSL) の一覧表示/作成/削除。 <div data-bbox="812 1317 922 1590" data-label="Image"> </div> <div data-bbox="992 1319 1067 1357" data-label="Section-Header"> <h4>注記</h4> </div> <div data-bbox="992 1395 1431 1588" data-label="Text"> <p>暗号化キーを作成したら、kickstart.profile.system.add_keys API メソッドを呼び出すことによって、キックスタープロファイルと関連付けすることができます。</p> </div>
kickstart.profile	IP 範囲の操作、キックスターツリーおよび子チャンネルの変更、プロファイルに関連付けされたキックスターファイルのダウンロード、高度なオプションやカスタムオプションの操作、キックスタープロファイルへのプリスクリプト/ポストスクリプトの追加。
kickstart.profile.keys	キックスタープロファイルに関連付けされているアクティベーションキーの一覧表示/追加 (関連付け)/削除 (関連付け解除)。

XML-RPC 名前空間	用途
<code>kickstart.profile.software</code>	キックスタープロファイルと関連付けされたパッケージ一覧の操作。
<code>kickstart.profile.system</code>	ファイル保持の管理、暗号化キーの管理、設定管理とリモートコマンドの有効化/無効化、パーティショニングスキームのセットアップ、特定のキックスタープロファイルに関連付けされたロケール情報のセットアップ。

ホストの再プロビジョニングとゲストインストールのスケジューリングに以下の API メソッドの呼び出しを使用することが可能です。

- `system.provision_system`
- `system.provision_virtual_guest`

API 呼び出しに関する詳しい情報は、<https://satellite.example.com/rpc/api> で入手可能な API ドキュメントをご参照ください。

4.2. COBBLER

RHN Satellite はプロビジョニングに **Cobbler** を使用します。キックスターのプロファイル、ツリー (ディストリビューション)、プロビジョニング対象システムが RHN Satellite 内で更新された際には、RHN Satellite ホスト上の **Cobbler** インスタンスと同期されます。これは、**Cobbler** を直接使用してプロビジョニングの管理を行うことができることを意味します。

以下の表は、**Cobbler** のコマンドについてまとめたものです。

表4.2 Cobbler のコマンド

コマンド	用途
<code>cobbler profile list</code>	RHN Satellite ホスト上でコマンドを実行して、プロファイルのリストを表示します。
<code>cobbler distro list</code>	キックスターツリー、カーネル、RAM ディスク、その他のオプションの一覧を表示します。
<code>cobbler system list</code>	キックスターのスケジュール時に作成されたシステム記録の一覧を表示します。
<code>cobbler profile report --name=profile-name</code> または <code>cobbler system report --name=system-name</code>	特定のオブジェクトに関するより詳細な出力を表示します。
<code>cobbler profile edit --name=profile-name --virt-ram=1024</code>	様々なパラメータを編集します。この例では、特定のプロファイルの仮想化インストールにそれぞれ 1 GB の RAM が割り当てられます。

コマンド	用途
<code>cobbler system edit --name=system-name --netboot-enabled=1</code>	次の再起動時に、システムを再インストールするように強制します。
<code>cobbler system edit --name=system-name --profile=new-profile-name --netboot-enabled=1</code>	再インストール用の新たなプロファイルにシステムを割り当てます。
<code>cobbler system find --profile=profile-name</code>	特定のプロファイルに割り当てられたシステムを一覧表示します。
<code>cobbler system find --profile="abc" xargs -n1 --replace cobbler system edit \ --name={} --profile="def" --netboot-enabled=1</code>	現在 <i>abc</i> プロファイルに設定されている全システムを <i>def</i> プロファイルに割り当て、次の再起動時に再インストールします。
<code>cobbler profile edit --name=profilename --kopts="variablename=3" --in-place</code>	その他の変数を一切変更せずに、プロファイル上の追加のテンプレーティング変数を設定します。
<code>cobbler system edit --name=systemname --kopts="selinux=disabled asdf=jkl"</code>	以前に設定されている可能性がある古い変数を無視して、システム記録に様々な変数を割り当てます。
<code>cobbler profile find --name="*webserver*" xargs -n1 --replace cobbler profile edit --name={} --profile="RHEL5-i386"</code>	<i>webserver</i> が文字列として含まれるプロファイルのすべての新規インストールを設定し、 <i>RHEL5-i386</i> という名前のプロファイルを使用します。

Cobbler のその他のセッティング

`/etc/cobbler/settings` ディレクトリ内で直接変更すべき Cobbler 設定はわずかしきありません。そのうちの 하나가、`pxe_just_once` オプションです ([手順4.3「PXE を使用するための Cobbler の設定」](#) で解説しています)。`server` オプションも、RHN Satellite Server のアドレスまたはホスト名を反映するように変更することができます。

`/etc/cobbler/settings` を変更した後は、以下のコマンドを実行して、変更をピックアップします。

```
/sbin/service cobblerd restart
cobbler sync
```



重要

`/etc/cobbler/settings` 内のその他の設定は、調整しないでください。RHN Satellite は、RHN Satellite のインストーラが決定した一定の設定が維持されていることを必要とします。また同様に、認証ソースを制御する `/etc/cobbler/modules.conf` ファイルも、RHN Satellite インストーラが作成した状態を維持する必要があります。特に、認証モジュールは `authn_spacewalk` のままにする必要があります、変更はできません。

手順4.1 Cobbler と併用するための SELinux の設定

Red Hat Enterprise Linux では、SELinux サポートとセキュアなファイアウォールがデフォルトでインストールされます。Cobbler を使用するように Red Hat Enterprise Linux サーバーを適切に設定するには、Cobbler サーバーとの接続を可能にするように SELinux を設定する必要があります。

1. SELinux の Cobbler サポートを有効化するには、以下のコマンドを使用して、SELinux のブーリアンが HTTPD Web サービスコンポーネントを許可するように設定する必要があります。

```
setsebool -P httpd_can_network_connect true
```

-P のスイッチは不可欠です。これによって、HTTPD 接続は再起動しても永続的に有効となります。

2. Cobbler サーバー上で以下のコマンドを使用して、TFTP 用の SELinux ファイルコンテキストルールがブートイメージファイルにサービスを提供するように設定します。

```
semanage fcontext -a -t public_content_t "var/lib/tftpboot/.*"

```

3. IPTables は、Cobbler サーバー上でのネットワークトラフィックの発着信を可能にするように設定する必要があります。

IPTables を使用する既存のファイアウォールルールセットがある場合、以下のようにルールを追加して、Cobbler 関連のポートを開きます。

TFTP の場合:

```
/sbin/iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 69 -j ACCEPT
/sbin/iptables -A INPUT -m state --state NEW -m udp -p udp --dport 69 -j ACCEPT
```

HTTPD の場合:

```
/sbin/iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
/sbin/iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
```

Cobbler の場合:

```
/sbin/iptables -A INPUT -m state --state NEW -m udp -p udp --dport 25150 -j ACCEPT
```

Koan の場合:

```
/sbin/iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 25151 -j ACCEPT
```

4. ファイアウォールの設定を保存します。

```
/sbin/iptables-save
```

5. 以下のコマンドを実行して、設定ファイルがすべて確実に同期されるようにします。

```
cobbler sync
```

6. Satellite Server を起動します。

```
/usr/sbin/rhn-satellite start
```



警告

Satellite サービスに依存しない **cobblerd** サービスを起動/停止しないでください。エラーやその他の問題が発生する原因となる可能性があります。

RHN Satellite の起動/停止には、必ず **/usr/sbin/rhn-satellite** を使用してください。

手順4.2 Cobbler システム記録の設定

Cobbler システム記録は、システムおよび関連するキックスタープロファイルを記録する Cobbler 内のオブジェクトです。PXE キックスターを実行するには、Satellite キックスタープロファイルを、キックスターするマシンの Cobbler システム記録に関連付けする必要があります。

1. 各システムのシステムの詳細 → **Provisioning** に進み、関連付けするキックスターを選択します。
2. **Cobbler システム記録の作成** ボタンをクリックし、関連付けします。
3. この関連付けは、任意のマシンに対して Cobbler 内で **pxe_just_once** を **true** に設定していない限りは、永久的に残ります。**true** に設定した場合は、キックスターが成功した後に関連付けは無効となります。この設定に関する詳しい情報は、[手順4.3「PXE を使用するための Cobbler の設定」](#)をご参照してください。

手順4.3 PXE を使用するための Cobbler の設定

Cobbler はデフォルトで PXE 設定を生成するように構成されていますが、BIOS 内で可能な限り最適な PXE ワークフローを取得するには、**pxe_just_once** 設定オプションの調整が可能です。

1. BIOS の順序は、多くの場合、PXE ブートが最初に実行されるように設定されます。これは、PXE サーバーがリモートで指示しない限りはローカルディスクからはブートしないということになります。この設定により、システムが継続的に再インストールを行う ブートループが発生する場合があります。

ブートループを防止するには、**/etc/cobbler/settings** ファイルを開いて、以下の行を追加します。

```
pxe_just_once: 1
```


この設定により、**\$kickstart_done** マクロがキックスターテンプレートに追加されます。これは、インストールが完了した後にシステムがネットワークからではなく、ローカルでブートするように指示します。

2. **pxe_just_once**: 1 セットアップを追加して、後でシステムを再インストールしたい場合には、システム上の **netboot-enabled** フラグを切り替える必要があります。これは、RHN Satellite の Web インターフェースを使用するか、直接 Cobbler 内で行うことができます。システムが次回再起動する時には、PXE インストールが実行され、その後には、フラグがリセットされるまでローカルディスクからのブートに戻ります。

ローカルのハードドライブから最初に起動するように BIOS が設定されている場合は、**pxe_just_once** を有効にする必要はありません。ただし、PXE を使用してシステムを再プロビジョニングするには、MBR (マスターブートレコード) をゼロ設定する必要があります。

命名規則

RHN Satellite と Cobbler 間のデータ同期を維持するために、RHN Satellite はディストリビューションとプロファイル用の命名規則を使用します。これらの命名規則は、コマンドラインインターフェースを使用して Cobbler とインタラクションを行う場合に重要となります。

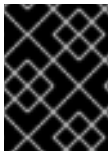
ディストリビューション

\$tree_name:\$org_id:\$org_name (手動で作成する場合)

\$tree_name (RHN Satellite によって同期される場合)

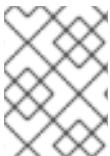
プロファイル

\$profile_name:\$org_id:\$org_name



重要

RHN Satellite が自動生成した名前は変更しないでください。名前が変更されると、RHN Satellite はこれらのアイテムを維持できなくなります。



注記

トラブルシューティングのために、Cobbler は RHN Satellite ログと `/var/log/cobbler/` ファイル内のデータのログ記録を取ります。

4.3. KOAN

koan (ネットワーク上でのキックスター) ユーティリティにより、プロビジョニング済みのホストから RHN Satellite へのリモートアクセスが可能になります。Koan により、キックスターのプロビジョニングを実行して、仮想ゲストを (仮想ホスト上に) 作成し、RHN Satellite ホストから利用可能なキックスターを一覧表示することが可能となります。これは、**koan** パッケージで利用可能です。

表4.3 Koan コマンド

コマンド	用途
<code>man koan</code>	koan の man ページをご参照ください。

コマンド	用途
<code>koan --replace-self --server=satellite.example.org --profile=profile-name</code> または <code>koan --replace-self --server=satellite.example.org --system=system-name</code>	既存システムの再プロビジョニングを行います。このコマンドを実行した後に再起動し、新たな OS をインストールします。これは、キックスタートのアップグレードにも使用できます (例: 多数のマシンを Red Hat Enterprise Linux のあるバージョンから次のバージョンにアップグレード)。
<code>koan --virt --server=satellite.example.org --profile=profile-name</code> または <code>koan --virt --server=satellite.example.org --system=system-name</code>	仮想ゲストのプロビジョニングを行います。
<code>koan --list=profiles --server=satellite.example.org</code> または <code>koan --list=systems --server=satellite.example.org</code>	Cobbler にクエリを行い、リモートインストールに利用可能なプロファイルまたはシステムの一覧を表示します。



注記

トラブルシューティングのために、Koan は `/var/log/koan` 内のデータのログ記録を取ります。

第5章 トラブルシューティング

5.1. Web インターフェース

問: **RHN Satellite** のユーザーインターフェースに不具合が生じました。どのログファイルをチェックしたらよいでしょうか。

答: **RHN Satellite** のユーザーインストールのキックスタートで表示、スケジューリング、操作のエラーが生じた場合には、**/var/log/tomcat5/catalina.out** ログファイルをチェックしてください。

その他すべてのインターフェースエラーの場合は、**/var/log/httpd/error_log** ログファイルをチェックします。

5.2. Anaconda

問: **Error downloading kickstart file** (キックスタートファイルのダウンロードでエラーが発生しました) というエラーが表示されます。何が問題なのでしょう。どのようにしたら修復できますか。

答: このエラーは、通常ネットワークの問題が原因です。問題を究明するには、**cobbler check** のコマンドを実行して出力を確認します。以下のような出力が表示されるはずです。

```
# 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
```

cobbler check で問題が究明できない場合には、以下の点を確認してください。

httpd が実行されているかを確認します。**service httpd status**

cobblerd が実行されているかを確認します。**service cobblerd status**

wget を使用して、異なるホストからキックスタートファイルを取得できることを確認します。

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

問: **The file chkconfig-1.3.30.1-2.i386.rpm cannot be opened.** (**chkconfig-1.3.30.1-2.i386.rpm** のファイルは開くことができません) というパッケージインストールエラーが表示されます。何が問題なのでしょう。どのようにしたら修復できますか。

答：クライアントは、キックスタート内の **--url** パラメータに基づいて **RHN Satellite** からコンテンツを取得します。例：

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

Anaconda から、イメージまたはパッケージが見つからないというエラーを受信した場合には、キックスタート内の URL が **200 OK** の応答を生成することを確認します。これは、**wget** でその URL にあるファイルの取得を試みることによって行うことができます。

```
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]
```

200 OK 以外の応答が返された場合は、エラーログを確認して問題を究明します。また、**access_log** ファイルを検索することによって、**Anaconda** がダウンロードを試みた実際のファイルをチェックすることもできます。

```
# 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"
```

これらの要求が **access_log** ファイル内に記載されていない場合は、システムのネットワーク設定に問題がある可能性があります。要求が記載されていてもエラーが生成される場合には、エラーログを確認してください。

手動でのファイルダウンロードを試みて、パッケージが入手可能かどうかを確認することもできます。

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

5.3. トレースバック

問: "WEB TRACEBACK" という表題の電子メールが送信されてきます。どう対処したらよいでしょうか。

答: 標準的なトレースバック電子メールは、以下のような内容となっています。

```
Subject: WEB TRACEBACK from satellite.example.com
Date: Wed, 19 Aug 2011 20:28:01 -0400
From: RHN 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
```

これは、**Cobbler** と **taskomatic** サービスとの通信において問題が生じたことを示しています。以下の点を確認してください。

httpd が実行されているかを確認します。 **service httpd status**

cobblerd が実行されているかを確認します。 **service cobblerd status**

localhost への接続を妨げるファイアウォールルールがないことを確認します。

5.4. 登録

問: `rhncp` コマンドを実行すると、**ERROR: unable to read system id** (エラー: システム ID を読み取ることができません) というエラーメッセージが表示されて、失敗してしまいます。何が問題なのでしょう。

答: キックスタートファイルの末尾には、`%post` のセクションがあり、これがマシンを RHN Satellite に登録します。

```
# 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/*
rhncp --serverUrl=https://satellite.example.com/XMLRPC --
sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT --activationkey=1-
c8d01e2f23c6bbaedd0f6507e9ac079d
# end Red Hat management server registration
```

追加された順番でこれを解釈すると、以下が実行されます。

RHN Satellite が使用するカスタム SSL 証明書を格納するディレクトリを作成します。

登録中に使用する SSL 証明書を取得します。

rhncp 設定ファイルからの SSL 証明書のストリングの検索/置換が行われ、SSL 証明書とアクティベーションキーを使用して RHN Satellite に登録されます。各キックスタートプロファイルにはアクティベーションキーが含まれ、これによってシステムに正しいベースチャンネルと子チャンネルが確実に割り当てられ、正しいシステムエンタイトルメントが取得されます。既存システムの再プロビジョニングの場合には、アクティベーションキーが以前のシステムプロファイルとの関連付けを確実にします。

`rhncp` コマンドが失敗した場合には、**ks-post.log** ログファイル内に以下のようなエラーが表示される場合があります。

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

このようなエラーは、**rhncp** を実行した際にシステムが RHN Satellite に登録されていない場合にも発生します。

この問題に対処する最善のトラブルシューティングは、キックスタートが完了した後に、キックスタートファイルを確認して、上記の 4 つのステップをコピーし、コマンドプロンプトに直接貼り付ける方法です。これによって、問題の究明に役立つ、より詳細なエラーメッセージが生成されます。

5.5. キックスタートとスニペット

問: キックスタートのディレクトリ構造はどのようになっていますか。

答: キックスタートファイルが保管されるベースパスは、`/var/lib/rhn/kickstarts/` です。このディレクトリ内において、**raw** キックスタートは **upload** サブディレクトリ内に格納され、ウィザードによって生成されたキックスタートは **wizard** サブディレクトリ内に格納されます。

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

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

問： **Cobbler** スニペットのディレクトリ構造はどのようになっていますか。

答： **Cobbler** スニペットは **/var/lib/rhn/kickstarts/snippets** に格納されます。 **Cobbler** は、 **/var/lib/cobbler/snippets/spacewalk** のシンボリックリンクを使用して、スニペットにアクセスします。

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



重要

RHN Satellite RPM は、**Cobbler** のキックスタートとスニペットのディレクトリがデフォルトの場所にあることを想定しているので、これらの場所を変更しないでください。

付録A 改訂履歴

改訂 4-2.2.400 Rebuild with publican 4.0.0	2013-10-31	Rüdiger Landmann
改訂 4-2.2 翻訳完成	Tue Apr 23 2013	Credit Translator's
改訂 4-2.1 翻訳ファイルをXMLソース 4-2と同期	Thu Feb 21 2013	Noriko Mizumoto
改訂 4-2 5.5 向け最終パッケージ	Wed Sept 19 2012	Dan Macpherson
改訂 4-1 リリースをステージング	Thu Aug 9 2012	Athene Chan
改訂 4-0 RHN Satellite 5.5 向けに第 1 章と第 2 章を更新 RHN Satellite 5.5 向けに第 3 章から第 5 章までを編集 BZ#787348 Cobbler iptable 行に誤り BZ#702529 新たなキックスタート情報を追加 BZ#797688 Cobbler 起動 ISO がサポートされていません。	Mon June 25 2012	Athene Chan
改訂 3-0 BZ#826803 - 「マシンのキックスタート」セクションの句読点を訂正。 「キックスタート」セクションの文法を若干変更	Thu May 31 2012	Athene Chan
改訂 2-0 BZ#783339 - 「プロビジョニングトラブルシューティングTaskomatic」セクションの文章を再構成 BZ#783340 - 「s390x」を「IBM System z」に更新	Thu May 24 2012	Athene Chan
改訂 1-3 z-stream リリースの変更を y-stream に適用	Mon Aug 15 2011	Lana Brindley
改訂 1-2 翻訳の準備	Wed Jun 15 2011	Lana Brindley
改訂 1-1 翻訳者からのフィードバックに基づいた更新	Fri May 27 2011	Lana Brindley
改訂 1-0 最終 QE レビューの編集 翻訳の準備	Fri May 6, 2011	Lana Brindley
改訂 0-8 BZ#701822 - QE レビュー	Thu May 5, 2011	Lana Brindley
改訂 0-7 テクニカルレビューのフィードバック	Thu April 14, 2011	Lana Brindley
改訂 0-6 テクニカルレビューの準備	Wed March 23, 2011	Lana Brindley
改訂 0-5	Tue March 22, 2011	Lana Brindley

BZ#666435
BZ#666846
BZ#678080
BZ#682364

改訂 0-4 トラブルシューティング	Tue March 22, 2011	Lana Brindley
改訂 0-3 複数の Satellite	Mon March 21, 2011	Lana Brindley
改訂 0-2 はじめに キックスタート 上級コマンド 章を再編成	Thu March 17, 2011	Lana Brindley
改訂 0-1 新たな章構成を完了	Wed Jan 5, 2011	Lana Brindley
改訂 0-0 オリジナルの RHN Satellite デプロイメントガイドから新規ドキュメントを作成	Tue Dec 21, 2010	Lana Brindley