



# Red Hat Satellite 5.8

## スタートガイド

Red Hat Satellite での基本セットアップと設定



# Red Hat Satellite 5.8 スタートガイド

---

Red Hat Satellite での基本セットアップと設定

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

## 法律上の通知

Copyright © 2017 Red Hat.

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## 概要

本書では、Red Hat Satellite を初めてお使いになるユーザー向けの情報と使用方法を説明します。本書では、基本的な概念や設定手順について説明します。Satellite の基本情報について更に詳しくは、Red Hat Satellite ユーザーガイドを参照してください。

## 目次

<b>第1章 チャンネル管理</b>	<b>3</b>
1.1. RED HAT NETWORK チャンネルの管理	3
1.2. カスタムチャンネルの作成と管理	7
1.3. 設定チャンネルの管理	25
<b>第2章 組織</b>	<b>30</b>
2.1. 組織の作成	30
2.2. 組織別のエンタイトルメントの管理	31
2.3. 複数の組織の管理	32
<b>第3章 システムのプロビジョニング</b>	<b>40</b>
3.1. RED HAT SATELLITE を使用したプロビジョニング	40
3.2. RED HAT SATELLITE PROXY を介したプロビジョニング	67
3.3. 仮想化されたゲストのプロビジョニング	67
3.4. クローンされたチャンネルまたはカスタムチャンネル経由のキックスタート	68
3.5. 再プロビジョニング	69
3.6. スナップショットでのロールバックのプロビジョニング	70
<b>第4章 システム管理</b>	<b>73</b>
4.1. SATELLITE へのシステムの登録	73
4.2. SATELLITE を使ったシステムの管理	89
4.3. 仮想化クライアントシステムの管理	100
<b>第5章 エラータ管理</b>	<b>113</b>
5.1. エラータの作成と編集	113
5.2. パッケージのエラータへの割り当て	114
5.3. エラータの発行	114
5.4. エラータのクローン作成	115
<b>付録A ブートデバイス</b>	<b>116</b>
<b>付録B カスタムパッケージの管理</b>	<b>121</b>
B.1. RED HAT NETWORK のパッケージの構築	121
B.2. RED HAT NETWORK パッケージ用のデジタル署名	123
<b>付録C 改訂履歴</b>	<b>127</b>



# 第1章 チャンネル管理

**Red Hat Satellite** チャンネルとは、複数のソフトウェアパッケージを1つにまとめた集合です。チャンネルは、目的に応じたルールでパッケージ群を分別する場合に役立ちます。例えば、特定の **Red Hat** ディストリビューションのパッケージ群をすべて1つのチャンネルに組み込むことができます。または、特定のアプリケーション用のパッケージ群や、特定ファミリーの傘下となる複数のアプリケーション用のパッケージ群を組み込むこともできます。ユーザー側の特定のニーズに合わせてチャンネルを定義することも可能です。例えば、企業ネットワークの特定のアーキテクチャー用のパッケージ群を含むチャンネルを作成することができます。

本章では、**Red Hat Satellite** で使用できる以下の2種類の基本的なチャンネルについて説明します。

1. **Red Hat チャンネル** - **Red Hat** のリリース済みパッケージ群を含む公式の **Red Hat** リポジトリです。
2. **カスタムチャンネル** - **Satellite** 管理者が組織やグループのニーズに基づいて作成するチャンネルです。これらは組織によって管理され、サードパーティーのパッケージとリポジトリが含まれる場合もあります。

## 1.1. RED HAT NETWORK チャンネルの管理

**Red Hat Satellite** チャンネルとは、複数のソフトウェアパッケージを1つにまとめた集合です。

### 1.1.1. ベースチャンネルと子チャンネルの区別

チャンネルには2種類のタイプがあります。ベースチャンネルと子チャンネルです。ベースチャンネルは、アーキテクチャーと **Red Hat Enterprise Linux** のリリースに応じたパッケージ群で構成されます。子チャンネルはベースチャンネルに付随するチャンネルであり、その他のパッケージがこれに含まれます。

システムがサブスクライブできるベースチャンネルは1つのみです。システムは、そのベースチャンネルに付随する複数の子チャンネルにサブスクライブさせることができます。**Satellite** チャンネルにサブスクライブさせたシステムは、そのチャンネルから利用できるパッケージのみのインストールと更新を行うことができます。

システムを **Satellite** に登録すると、そのシステムの **Red Hat Enterprise Linux** バージョンに応じたベースチャンネルが割り当てられます。このデフォルトのベースチャンネルをプライベートのベースチャンネルに変更する場合は、システムの登録が完了してから **Satellite web** インターフェースまたは **API** 経由でシステム単位で行なうことができます。または、カスタムのベースチャンネルに関連付けを行ったアクティベーションキーを使用することもできます。このようなアクティベーションキーで登録を行うと、システムに自動的にそのカスタムのベースチャンネルが関連付けられます。

**Satellite Web** インターフェース上のチャンネルページ(上部ナビゲーションバーにあるチャンネルタブの下)には、すべてのベースチャンネルとその子チャンネルの一覧が表示されます。チャンネル名をクリックすると、チャンネルの詳細ページが表示されます。このページには、そのチャンネル内の全パッケージ、エラータ、および関連付けられているシステムなどの一覧が表示されます。

### 1.1.2. Red Hat Satellite へのシステムのサブスクライブ

システムを次の手順でチャンネルにサブスクライブさせます。

- アクティベーションキーを使って登録する - アクティベーションキーを使用する方法が簡単で速いため、**Red Hat Satellite Proxy Server** や **Red Hat Satellite Server** いずれかのクライアントとしてシステムを登録する場合には適した方法になります。アクティベーションキーを使用し

て登録されたシステムはそのアクティベーションキーに関連付けられたすべてのチャンネルにサブスクライブされます。アクティベーションキーに関しては『Red Hat Satellite クライアント設定ガイド』または『Red Hat Satellite リファレンスガイド』をご覧ください。

- インストール時に登録する - **Red Hat Update Agent** または **Red Hat Network Registration Client** のいずれかで初めてシステムを登録すると、そのシステムの Red Hat Enterprise Linux バージョンに応じたベースチャンネルが自動的に割り当てられます。そのデフォルトのベースチャンネルをプライベートのベースチャンネルに変更する場合は、一旦システムの登録を完了させた後に、**Red Hat Satellite** を使用してシステム単位で行なうことができます。これらのアプリケーションの使い方については、エンタイトルメントレベル (**Management** または **Provisioning**) に応じて『Red Hat Satellite リファレンスガイド』の該当する章を参照してください。
- Web サイトでのサブスクライブ - システムのベースチャンネルに応じて、さまざまな子チャンネルのサブスクライブが可能です。子チャンネルへのサブスクライブは **Satellite Web** インターフェースから行なうことができます。カスタムのベースチャンネルを作成している場合、そのカスタムのベースチャンネルへの再割り当ても Web サイトから行なうことができます。オンラインでのチャンネルへのサブスクライブについてさらに詳しくは、『Red Hat Satellite リファレンスガイド』の **Satellite Web** サイトの章を参照してください。
- **spacewalk-channel** コマンドラインツール (CLI) の使用 - **spacewalk-channel** を使用すると、Red Hat Network Web サイトにログインしなくてもコマンドラインを使って特定のチャンネルへのサブスクライブを行なうことができます。

いくつか例を示します。2 種類のチャンネルにサブスクライブさせる場合:

```
# spacewalk-channel --add -c rhn-tools-rhel-x86_64-server-6 -c \
    rhel-x86_64-server-6 --user username --password password
```

チャンネルのサブスクライブを中止する場合:

```
# spacewalk-channel --remove -c rhn-tools-rhel-x86_64-server-6 -c \
    rhel-x86_64-server-6 --user username --password password
```

サブスクライブしているチャンネルを表示させる場合:

```
# spacewalk-channel --list
```

### 1.1.3. Red Hat Satellite からの Red Hat ベースチャンネルの削除

管理者が Red Hat ベースチャンネルを削除しなければならないいくつかの状況があります。以下がその一部の例になります。

- 特定のアーキテクチャーが組織によってサポートされておらず、利用できない。
- 特定のアーキテクチャーのサブスクリプションの有効期限が切れており、更新が利用できない。
- 製品チャンネルのサポートが終了している。
- Red Hat Satellite サーバーにディスクの空き容量が必要である。

前提条件:

**spacewalk-backend-tools** バージョン 0.5.28.49 以降が以下のコマンドの実行に必要です。



Red Hat ベースチャンネルを Red Hat Satellite から削除するには:

1. Red Hat Satellite サーバーに **root** としてログインします。
2. Satellite がサブスクライブされているすべてのサブスクライブしているチャンネルを表示するには:

```
# spacewalk-remove-channel --list
```

削除するチャンネルに注意してください。これは、**channel\_label** というチャンネルで、次のステップで使用されます。

3. チャンネルを Satellite から削除するには:

```
# /usr/bin/spacewalk-remove-channel -c channel_label --unsubscribe
```

上記の設定で、

- **--unsubscribe** は、削除するチャンネルにサブスクライブされているすべてのシステムのサブスクライブを自動的に中止します。



### 注記

この手順は、Red Hat Satellite 5.3 以降にのみ適用されます。Red Hat Satellite 5.2 以前の場合は、スクリプトを使用して、Red Hat が提供するチャンネルを Red Hat Satellite から削除することができます。ナレッジベースの記事の『[How do I delete a Red Hat Base channel from my Red Hat Satellite?](#)』をご覧ください。

## 1.1.4. リポジトリの管理

リポジトリはチャンネルとよく似ており、これにはパッケージのセットが含まれます。ただし、チャンネルは複数のリポジトリを格納できるものの、リポジトリにはパッケージのセットのみが含まれます。Red Hat Satellite では、ほとんどのリポジトリは **yum** ツールを使用して作成されます。これらのリポジトリは、通常はサードパーティーの特定のアプリケーション用のパッケージを提供できるように Red Hat Satellite のカスタムチャンネルに追加されます。

### 1.1.4.1. リポジトリの追加

リポジトリを追加して、異なるソースからの追加パッケージをカスタムチャンネルと同期することができます。リポジトリ URL は、yum リポジトリのルートディレクトリを示す必要があります。

1. チャンネル管理者または組織管理者としてログインします。
2. チャンネル → ソフトウェアチャンネルの管理 → リポジトリの管理 をクリックします。
3. ページの右上にある、**新規リポジトリの作成** をクリックします。
4. 以下のフィールドに入力します。
  - **リポジトリラベル** - リポジトリを識別する名前です
  - **リポジトリ URL** - リポジトリの場所への有効な URL。
  - **SSL CA 証明書** - HTTPS 上でリポジトリにアクセスするために必要な SSL 証明機関。

- **SSL クライアント証明書** - HTTPS 上でリポジトリにアクセスするために必要な SSL 証明書。
- **SSL クライアントキー** - HTTPS 上でリポジトリにアクセスするために必要な SSL キー。



### 注記

システム → キックスタート → **GPG および SSL キー** と移動して、新しい SSL 証明書の作成および既存の証明書の管理を行います。

- **Filters** - リポジトリに適用するパッケージフィルターを定義します。プラス (+) を使用してパッケージを追加し、マイナス (-) を使用してパッケージを除外します。以下の例では、**kernel** パッケージと、**zsh-html** 以外の **zsh** で始まるパッケージが除外されます。

```
-zsh*,kernel +zsh-html
```

## 5. リポジトリの作成 を作成します。

### 1.1.4.2. リポジトリのチャンネルへの追加

Red Hat Satellite 上でリポジトリを使用するために、リポジトリをチャンネルに追加する必要があります。リポジトリは、必要な分だけチャンネルに追加できます。

リポジトリをチャンネルに追加するには以下を実行します。

1. チャンネル → ソフトウェアチャンネルの管理 をクリックします。
2. リポジトリを組み込む特定のチャンネルを選択します。
3. リポジトリ サブタブを選択し、チャンネルに追加するリポジトリを選択します。
4. チャンネルに追加するリポジトリを選択します。
5. リポジトリの更新 をクリックします。

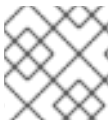
### 1.1.4.3. リポジトリの同期のスケジューリング

ソースリポジトリは、アップグレード、セキュリティおよびバグ修正に基づいて変更されたり、更新されたりする可能性があります。Red Hat Satellite のカスタムリポジトリは、インポートされるパッケージが最新の更新バージョンであることを保証できるよう、ソースリポジトリと同期する必要があります。ほかにも、新規のパッケージがソースリポジトリに追加されている場合は、これらのパッケージは、Red Hat Satellite カスタムリポジトリにもダウンロードされます。

リポジトリの同期をスケジュールするには以下を実行します。

1. チャンネル → ソフトウェアチャンネルの管理 をクリックします。
2. リポジトリがメンバーとなるチャンネルを選択します。
3. リポジトリ → 同期 をクリックします。
4. 同期をスケジュールする時間を選択します。同期を即時にスケジュールするには **今すぐ同期** をクリックするか、または以下のオプションを指定してスケジュールを選択します。

- **スケジュールを無効にする** - 現在実施されているスケジュールを無効にします。
  - **毎日** - 指定される時間に、ソースリポジトリとの日次の同期をスケジュールします。
  - **毎週** - 指定される日時に、ソースリポジトリとの週次の同期をスケジュールします。
  - **毎月** - 指定される月と時間に、月次の同期をスケジュールします。
  - **カスタム Quartz 形式** - 同期のカスタムスケジュールを定義します。
5. 変更を保存し、同期をスケジュールするには、**スケジュール** をクリックします。



### 注記

同期が完了した後にパッケージがチャンネルに割り当てられます。

#### 1.1.4.4. リポジトリの削除

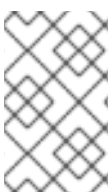
1. チャンネル管理者または組織管理者としてログインします。
2. チャンネル → ソフトウェアパッケージの管理 → リポジトリの管理 をクリックします。
3. 削除するリポジトリを選択します。
4. ページの右上にある、**リポジトリの削除** をクリックします。

## 1.2. カスタムチャンネルの作成と管理

Red Hat Satellite には多くのチャンネルがあります。すべてのユーザーが利用できるチャンネル、特定の組織に所属するユーザーが利用できるチャンネル、および特定のチャンネルのアクセス用にサブスクリプションを購入した場合にのみ使用できるチャンネルなどがあります。チャンネルは次のようなカテゴリーに分かれます。

- **有料サービスチャンネル** - 有料サービスチャンネルへのアクセスを直接購入した場合、または特定の Red Hat ソリューションと合わせてアクセスを購入した場合に使用できるチャンネルです。Red Hat Enterprise Linux などが有料サービスチャンネルの一例です。
- **カスタムチャンネル** - カスタムのパッケージを管理する目的で組織の管理者によって作成されるチャンネルです。プライベートのチャンネルとも呼ばれるこのチャンネルは、デフォルトでは作成側の企業または組織にしか表示されません。このため、それらの企業または組織以外からのアクセスは一切ありません。ただし、プライベートのチャンネルは、組織的な信頼を設定し、チャンネルを共有することで複数の組織間での共有が可能になります。

特定の組織によって構築され、保守が行われるパッケージの配備を管理者が Red Hat Satellite のインフラストラクチャーを利用して行えるようにするのがカスタムチャンネルです。チャンネルやパッケージの管理作業はすべて、Red Hat Satellite Web サイトの **チャンネル** タブで行います。



### 注記

実稼働環境向けのテストを行っていないパッケージを配備すると問題が発生する可能性があるため、ステージング用に選択したシステム群を対象とするベータチャンネルをまず作成することを推奨しています。

たとえば、複数の Web サーバーで構成され、カスタムパッケージのセットを受け取るシステムグルー

プが1つある場合、最初に一時的なチャンネルを作成し、重要性の低い代表システム上にパッケージをインストールします。このシステム群は **実際に稼働中のライブシステムではなく**、開発用またはステージング用のサーバーになります。一時的なチャンネルは「[ソフトウェアチャンネルの削除](#)」の手順に従い、後で削除してください。

### 1.2.1. ツール、リポジトリ、および実践例について

チャンネルを作成し、管理する前に、ユーザーが使用できる各種ツールやリポジトリの違いについて注意してください。特に **Red Hat Satellite Server** と **Red Hat Satellite Proxy Server** の両方を配備する場合は、使用できるユーティリティや保存場所も増えるため重要となります。また、**Proxy** と **Satellite** の組み合わせは、最適なパフォーマンスを得るためのベストプラクティスを実現します。

次のパッケージ管理ツールは、**Red Hat Satellite** で使用されます。

- **Red Hat Network Package Manager** - カスタムのパッケージを **Red Hat Satellite Proxy Server** 上のカスタムチャンネルにプッシュする際に使用します。
- **Red Hat Network Push** - カスタムのパッケージを **Red Hat Satellite Server** 上のカスタムチャンネルにプッシュする際に使用します。
- **Red Hat Satellite Synchronization Tool** - 指定の場所で、**Red Hat Network Classic** から **Red Hat Satellite** サーバーに標準パッケージをインポートしたり同期したりする際に使用します。インターネット経由または CD や DVD の ISO イメージを利用して行います。

各ツールにはそれぞれ対応するパッケージリポジトリがあります。**Red Hat Network Package Manager** および **Red Hat Network Push** は、いずれの場合も **Proxy** または **Satellite** にアップロードするカスタムパッケージを配置する一時的なステージングディレクトリを作成する必要があります。使用後はステージングディレクトリを削除してください。



#### 注記

カスタムのパッケージをアーカイブする際は、**Red Hat Satellite** とは別の場所にアーカイブすることを推奨しています。

**Red Hat Satellite Proxy Server** と **Red Hat Satellite** の両方を使用している場合は、**Red Hat Network Push** と **Red Hat Satellite Synchronization Tool** 以外のツールは使用しないでください。**Proxy** と **Satellite** を組み合わせる場合、カスタムのパッケージやチャンネルのアップロード先は **Satellite** のみにしてください。そこから、**Proxy** はパッケージを取得し、それらをクライアントのシステム群に配信します。

### 1.2.2. ソフトウェアチャンネルの作成

パッケージをサーバーにアップロードする前に、パッケージを格納するカスタムチャンネルを作成することができます。詳細は「[カスタムパッケージのアップロードと保守](#)」を参照してください。パッケージをアップロードしたら、「[パッケージのソフトウェアチャンネルへの割り当て](#)」に説明されているように **Web** サイトからパッケージを再び割り当てることができます。

チャンネルは次のように **Red Hat Satellite Web** サイト上で作成します。

1. チャンネル管理者として **Red Hat Satellite Web** サイトにログインします。
2. 上部のナビゲーションバーで **チャンネル** タブをクリックしてから左側のナビゲーションバーにある **ソフトウェアチャンネルの管理** ボタンをクリックします。
3. **ソフトウェアチャンネル管理** ページで右上にある **新しいソフトウェアチャンネルの作成** をク

リックします。Red Hat Satellite Server の管理者には **チャンネルのクローン作成** のオプションが表示されます。詳細は「[ソフトウェアチャンネルのクローン作成](#)」を参照してください。

4. **新規チャンネル** ページ上でページの指示に従ってチャンネルの詳細を定義します。ほとんどのチャンネル管理作業でチャンネルを識別する際は **チャンネルラベル** が使用されるため、ラベルにはわかりやすいラベル名を付けてください。既存のチャンネルの詳細を参照するとよいでしょう。

**GPG キーの URL** はサーバー上のキーの配置場所になります。この配置場所は、クライアント設定のプロセスで定義します。『Red Hat Satellite クライアント設定ガイド』を参照してください。GPG キー ID は「DB42A60E」などの固有の識別子となります。GPG キーのフィンガープリントは「CA20 8686 2BD6 9DFC 65F6 ECC4 2191 80CD DB42 A60E」などの文字列になります。キー ID はキーのフィンガープリント内の最後の 8 文字と同じである点に注意してください。

5. 終了したら、ページ下部の **チャンネルの作成** をクリックします。

### 1.2.3. パッケージのソフトウェアチャンネルへの割り当て

初めてパッケージをアップロードする場合、パッケージを1つのカスタムチャンネルまたは複数のカスタムチャンネルに割り当てることができます。また、チャンネルに割り当てなくても構いません。詳細は「[カスタムパッケージのアップロードと保守](#)」を参照してください。アップロードが終了したら、カスタムチャンネル間で再割り当てを行ったり、割り当てチャンネルなしのリポジトリに再度割り当てたりすることもできます。

以下の手順にしたがって、これらの機能を使用できるようにします。

1. 上部ナビゲーションバーにある **チャンネル** タブ、次に左側のナビゲーションバーの **ソフトウェアチャンネルの管理** をクリックします。
2. **ソフトウェアチャンネル管理** ページで、パッケージを受信するチャンネルのチャンネル名をクリックします。
3. **ベースチャンネルの詳細** ページで、**パッケージ** のタブ、次に **追加** のサブタブをクリックします。編集しているチャンネルにパッケージを関連付けるには、**表示** のドロップダウンメニューからそのパッケージを含むオプションを選択し、**表示** をクリックします。



#### 注記

編集しているチャンネルにすでに割り当てられているパッケージは表示されません。特定のチャンネルに割り当てられていないパッケージは **チャンネルに割り当てられていないパッケージ (Packages in no channels)** のメニューアイテムで確認できます。管理している全パッケージ (**All managed packages**) を選択すると使用できる全パッケージが表示されます。

4. 編集しているチャンネルに割り当てるパッケージのチェックボックスを選択して、ページ右下にある **パッケージの追加** をクリックします。選択したパッケージが記載された確認ページが表示されます。
5. **確認** をクリックするとパッケージがチャンネルに割り当てられます。**管理しているソフトウェアチャンネルの詳細** ページの **一覧表示/削除** のサブタブに新しいパッケージが表示されます。

パッケージがチャンネルに関連付けられると、エラータのキャッシュが更新されて変更が反映されま

す。ユーザーがチャンネルの編集を終了してからすべての変更が使用可能になるように更新のタイミングは若干遅れます。キャッシュへの変更を手動で開始する場合は **一覧表示/削除** サブタブの上部にあるテキスト内の **直ちに変更をコミット** をクリックします。

#### 1.2.4. チャンネル管理の特権の管理

チャンネル管理の作業を行うには **チャンネル管理者** として適切なパーミッションを取得している必要があります。パーミッションは **Red Hat Satellite Web** サイトで変更することができます。パーミッションは最上位レベルの管理者となる **組織の管理者** によってユーザーに割り当てられます。チャンネル管理者の特権は次のように割り当てます。

1. 組織の管理者として **Red Hat Satellite Web** サイトにログインします。
2. 上部ナビゲーションバーで **ユーザー** タブをクリックしてチャンネル管理の機能を使用するユーザー名をクリックします。
3. **ユーザーの詳細** ページで **ロール** セクションまでスクロールして **チャンネル管理者** のラベルが付いたチェックボックスを選択します。ページ下部にある **サブミット** をクリックします。組織の管理者にはチャンネル管理者の特権が自動的に与えられます。
4. このユーザーで **Red Hat Satellite Web** サイトにログインし、上部ナビゲーションバーの **チャンネル** タブをクリックして **ソフトウェアチャンネルの管理** ボタンが左側のナビゲーションバーに表示されることを確認します。

#### 1.2.5. カスタムチャンネルのパーミッションの変更

カスタムチャンネルは、特定の組織およびシステムに制限することができます。以下の場合にこの制限が役立ちます。

- 本番稼働前にソフトウェアの各種設定をテストするなどの評価目的で、チャンネルのコンテンツを特定の組織とシステムに制限する
- ライセンス化されているパッケージまたはサードパーティーのパッケージの制御された配信

##### 1.2.5.1. カスタムチャンネルのユーザーパーミッションの変更

###### 前提条件

パーミッションの変更を必要とする既存のチャンネルがあることを前提とします。

1. チャンネルまたは組織の管理者として **Satellite** サーバーにログインします。
2. **チャンネル** → **ソフトウェアチャンネルの管理** をクリックします。
3. パーミッションを変更する必要があるチャンネルをクリックします。
4. **チャンネルアクセス制御** → **ユーザー毎のサブスクリプション制限**、およびこの組織内の指定されたユーザーのみがこのチャンネルにサブスクライブ (**Only selected users within your organization may subscribe to this channel.**) までスクロールします。
5. **チャンネルの更新** をクリックして変更を保存します。
6. **サブスクライバーサブタブ** をクリックして、チャンネルにサブスクライブできるはずのユーザーを選択します。
7. **更新** をクリックします。

### 1.2.5.2. カスタムの企業/組織のパーミッションの変更

#### 前提条件

パーミッションの変更を必要とする既存のチャンネルがあることを前提とします。

1. チャンネルまたは組織の管理者として **Satellite** サーバーにログインします。
2. チャンネル→ソフトウェアチャンネルの管理をクリックします。
3. パーミッションを変更する必要があるチャンネルをクリックします。
4. チャンネルアクセス制御→組織共有までスクロールします。以下のいずれかを選択します。
  - このチャンネルはプライベートのため、他の組織はアクセスできません。
  - このチャンネルは保護されているため、特定の信頼された組織のみがアクセスできます。
  - このチャンネルは公開されているため、この組織が信頼するすべての信頼された組織がアクセスすることができます。
5. 「チャンネルの更新」をクリックします。
6. (オプション) 保護されたチャンネルを選択すると、**Satellite** サーバーは、チャンネル共有に対して行われた変更を確認するように指示します。チャンネルパーミッションの変更によって削除されるチャンネルにシステムがサブスクライブされている可能性があるためです。以下のいずれかを実行するように選択します。
  - **アクセスを拒否して確認** をクリックして、信頼された組織から以前にサブスクライブされたすべてのシステムのサブスクライブを中止します。
  - **アクセスを許可して確認** をクリックして、信頼された組織からサブスクライブされたシステムを保持します。
  - いずれかの操作を実行する前に、システムと信頼された組織を確認する場合は **キャンセル** をクリックします。

### 1.2.6. ソフトウェアチャンネルの管理

標準の **Red Hat Satellite** のユーザーが使用できるボタンやページのほかに、組織またはチャンネル管理者のロールを持つユーザーは **チャンネル** をクリックして、**Red Hat Satellite Server** と **Red Hat Satellite Proxy Server** のお客様の左側のナビゲーションバーにある **ソフトウェアチャンネルの管理** へのアクセスを許可することができます。このボタンをクリックすると、ソフトウェアチャンネル管理のページが開きます。カスタムのソフトウェアチャンネル管理の作業はすべてここで行います。



#### 警告

**Red Hat Satellite Proxy Server** と **Red Hat Satellite Server** の両方を使用している場合、**Proxy** サーバー群は **Satellite** から直接更新を受信するため、カスタムのチャンネルとパッケージの管理は **Satellite** 上でのみ行うようにしてください。これらの両方を使用する構成で任意の **Proxy** でパッケージやチャンネルの管理を手作業で行うとサーバー群が同期されない恐れがあります。



ソフトウェアチャンネル管理の一覧にある異なるサブタブを選択すると、ベースチャンネルの詳細ページの異なるタブに移動します。

### 1.2.7. ベースチャンネルの詳細

カスタムのチャンネル管理のほぼすべての作業は **ベースチャンネルの詳細** ページ内で行われます。左側のナビゲーションバーにある **ソフトウェアチャンネルの管理** をクリックして、変更するチャンネル名を選択するとこのページにアクセスすることができます。このページは次のようなタブで構成されています。

- **詳細** - 親チャンネル、チャンネル名、要約および説明などのチャンネルに関する基本的な情報が表示されます。この情報の一部は変更が可能です。また、組織の管理者とチャンネル管理者からは **ユーザー毎のサブスクリプション制限** のコンボボックスが見えるようになっています。すべてのチャンネルのデフォルトの動作により、すべてのユーザーがシステムをこのチャンネルにサブスクライブさせることができるようになっています。このボックスのチェックを外して **チャンネルの更新** をクリックすると **サブスクライバー** タブが表示されます。このタブを使用して特定のユーザーにこのチャンネルへのサブスクリプションパーミッションを与えます。
- **組織** - チャンネル内のコンテンツの表示および使用へのアクセスをチャンネルが付与した組織の一覧が表示されます。これらの組織は、組織の信頼が確立されているために表示されます。このチャンネルへの組織によるアクセスはここで修正できます。このチェックボックスを選択して **アクセスの変更** をクリックし、組織のアクセスを削除できます。組織の管理者とチャンネル管理者は全チャンネルへのサブスクリプションのアクセスが自動的に与えられていることに注意してください。
- **マネージャー** - カスタムチャンネルの管理パーミッションを有するユーザーを一覧表示します。組織の管理者およびチャンネル管理者に対してこのタブが表示されます。このチャンネルのすべての管理パーミッションを許可するユーザーのチェックボックスを選択して **更新** をクリックします。このステータスでは、ユーザーは新規のチャンネルを作成することはできません。組織の管理者とチャンネル管理者は全チャンネルへの管理アクセスが自動的に与えられていることに注意してください。
- **エラータ** - 各カスタムチャンネルに関連付けられたエラータを表示します。Red Hat Network が Red Hat Enterprise Linux ソフトウェアに対してエラータ更新を生成し、配信するのと同様に、最新コードによるサーバー更新の一環として、カスタムチャンネルにエラータ更新を配信します。このタブには、エラータの表示や追加、削除、およびクローン作成などができる **一覧表示/削除**、**追加**、および **クローン作成** などのサブタブが含まれています。エラータのクローン作成は Red Hat Satellite Server からしか行えませんが注意してください。
  - **一覧表示/削除** - カスタムチャンネルに現在関連付けられているすべてのエラータを表示し、その関連付けを取り消すことができます。チャンネルからエラータを削除するには、エラータのチェックボックスを選択してページ右下の **エラータの削除** をクリックします。削除するエラータが一覧表示されている確認のページが表示されます。**確認** をクリックして削除の作業を完了します。
  - **追加** - チャンネルにエラータを追加できます。チャンネルに適用できる可能性があるエラータがすべて表示されます。チャンネルにエラータを追加するには、該当のチェックボックスを選択して **エラータの追加** をクリックします。エラータの管理については、[5章 エラータ管理](#) を参照してください。
  - **クローン作成** - Satellite を利用している場合、これを使用するとクローン作成したチャンネル用にエラータと関連のパッケージを複製することができます。このサブタブは、「オリジナルの状態」または「エラータ選択」のいずれかのオプションを使ってクローン作成したチャンネル用にフィールドが入力された状態ですぐに表示されます。ターゲットチャ



ンネル(元となったチャンネル)に対してエラータが発行されると常に **クローン** タブもエラータを取得します。これは、現在の状態のオプションでチャンネルをクローン作成した場合に役立ちます。クローン作成オプションの詳細については「[ソフトウェアチャンネルのクローン作成](#)」を参照してください。

クローン作成したチャンネルにターゲットチャンネルからのエラータを組み込むには、各アドバイザリのドロップダウンメニューから **マージ** または **クローン作成** のいずれかを選択します。**マージ** オプションは、そのエラータのクローンが以前に作成されている場合にのみ表示されます。エラータをチャンネル全体に関連付ける際に重複エントリを避けるために使用します。以前のクローンからエラータに修正を加える場合など、新規のエントリを作成する場合は、**クローン作成** オプションを使用します。

デフォルトでは、クローン作成したエラータはオリジナルの **Red Hat** アドバイザリーのラベルを継承し、「RH?」のプレフィックス部分が「CL」になります。たとえば、**RHSA-2003:324** は **CLA-2003:324** になります。以降の同じアドバイザリーのクローンは「CM」や「CN」などのように 2 番目の文字がその順序を表します。ラベルは「エラータ管理」ページで変更することができます。詳細は [5章エラータ管理](#) を参照してください。

以前にクローンされたエラータには、**マージ** オプションのほかに、**所有しているエラータ** の列内に値が含まれます。エラータラベルは詳細ページにリンクしています。そのクローンが作成されたエラータは発行済みなのか、またはオリジナルのアドバイザリーから変更が加えられたのかは、括弧で囲まれた **pub** と **mod** の各フラグで識別します。フラグの前にプラスの印 + があれば肯定を示し、そのクローンのエラータは発行されています。フラグの前のマイナスの印 - は否定を意味します。たとえば、**(-mod)** はパッケージが削除されたという意味になる場合があります。カスタムのエラータの発行および編集については [5章エラータ管理](#) を参照してください。

クローン作成したチャンネルからエラータを排除する場合は、ドロップダウンメニューで **何も行わない** を選択します。変更を確認したら **エラータのクローン作成** をクリックします。確認のページで発生する変更を確認して **エラータの更新** をクリックします。

- **同期** - 最初にチャンネルのクローンを作成した時には含まれておらず、その後の更新で追加されたエラータパッケージを表示します。このページでは、必要なチェックボックスに印を付け **エラータの同期** をクリックすることで、クローン作成されたチャンネルを現在のエラータに同期させることができます。
- **パッケージ** - 各カスタムチャンネルに関連付けされたパッケージを表示します。このタブにはパッケージの表示、追加および削除を行うことができる **一覧表示/削除**、**追加**、**比較** などのサブタブが含まれています。
  - **一覧表示/削除** - 現在、カスタムチャンネルに関連付けられている全パッケージを表示します。また、その関連性を取り消すこともできます。チャンネルからパッケージを削除する場合は、そのパッケージのチェックボックスを選択してページ右下の **パッケージの削除** をクリックします。削除対象のパッケージを一覧表示した確認のページが表示されます。**確認** をクリックして削除作業を完了します。



## 重要

このページに表示される一覧は、**ソフトウェアチャンネルの詳細** ページで利用できる標準的なパッケージの一覧とは異なり、パッケージの最新バージョンだけでなくデータベースにある全バージョンを表示します。最新バージョンを削除するとパッケージの旧バージョンに戻ることができます。

- **追加** - チャンネルにパッケージを追加することができます。利用可能なパッケージを表示するには **表示** ドロップダウンメニューから 1 つのオプションを選択して **表示** をクリック

します。編集中のチャンネルにパッケージを追加する場合は、該当するチェックボックスを選択して **パッケージの追加** をクリックします。この手順に関する詳細は「[パッケージのソフトウェアチャンネルへの割り当て](#)」を参照してください。

- **比較** - 異なるチャンネル間でパッケージ一覧の比較ができます。違いを見るには、「**比較対照 (Compare to):**」ドロップダウンメニューからもう1つのチャンネルを選択して **比較** をクリックします。両方のチャンネルには含まれていないパッケージがすべて表示され、それぞれの既存チャンネルの場所が示されます。
- **リポジトリ - リポジトリの管理** を選択して、チャンネルに **yum** リポジトリを割り当てリポジトリの内容を同期します。
  - **追加/削除** - 設定したリポジトリの一覧を表示します。リポジトリ名の横にあるチェックボックスを選択してから **リポジトリの更新** をクリックするとリポジトリの追加や削除を行うことができます。
  - **同期** - 設定したリポジトリを一覧表示します。同期のスケジュールはドロップダウンボックスを使って設定できます。または、**今すぐ同期** をクリックするとすぐに同期を開始することもできます。

### 1.2.8. ソフトウェアチャンネルのクローン作成

Red Hat Satellite Server のチャンネル管理者の場合、パッケージの関連付けを容易にするためソフトウェアチャンネルのクローンを作成することもできます。クローンを作成することにより、チャンネルの完全な複製が可能になるため、パッケージとエラータのカスタムソフトウェアチャンネルへの関連付けが迅速に行なえるようになります。

次のようにして行ないます。

1. 上部ナビゲーションバーの **チャンネル** タブ、次に左側のナビゲーションバーの **ソフトウェアチャンネル管理** をクリックすると、**ソフトウェアチャンネル管理** のページに移動します。
2. 右上の **チャンネルのクローン** をクリックします。

クローン作成のオプションが3つ表示されます。チャンネルの現在の状態、チャンネルのオリジナルの状態、またはエラータ選択の3つです。詳細については **Web** ページに記載されていますが以下に要約します。

- **チャンネルの現在の状態 (Current state of the channel)** - 現在、ターゲットチャンネルにある最新の全パッケージと全エラータです。
  - **チャンネルのオリジナルの状態 (Original state of the channel)** - ターゲットチャンネルからの全オリジナルパッケージを含みます。ただし、エラータや関連の更新パッケージは含まれません。
  - **エラータ選択 (Select Errata)** - ターゲットチャンネルからの全オリジナルパッケージが含まれます。このオプションでは特定のエラータや関連の更新パッケージを除外することができます。
3. クローンフィールド内のラジオボタンを使用して該当オプションを選択します。**クローン作成元 (Clone From)** ドロップダウンメニューを使ってターゲットチャンネルを特定し、**チャンネルの作成** をクリックします。

4. 「ソフトウェアチャンネルの作成」で説明されているとおり、**新規のソフトウェアチャンネル (New Software Channel)** ページのフィールドを入力します。デフォルトの値のままでよいでしょう。
5. **チャンネルの作成** をクリックします。「オリジナルの状態」または「現在の状態」のいずれかを選択すると、**管理しているソフトウェアチャンネルの詳細** ページの **詳細** タブが表示されます。新しいチャンネルの設定を変更します。詳しくは、「**ベースチャンネルの詳細**」を参照してください。

チャンネルのクローン作成に「エラータ選択」のオプションを選んだ場合は、**管理しているソフトウェアチャンネルの詳細** ページの **クローン** サブタブにリダイレクトされます。クローンや新しいチャンネルへの組み込みを行うには、エラータやクローンに関連するパッケージを個別に選択する必要がある場合があります。手順は「**ベースチャンネルの詳細**」を参照してください。



### 注記

パッケージの整合性と再現性を維持するため、日付に基づいて指定チャンネル内に全エラータのクローンを作成するコマンドを使用できます。これは **spacewalk-clone-by-date** というコマンドです。

## 1.2.9. 特定の更新レベルからのカスタムチャンネルの作成

以下は、特定の更新レベルでのカスタムチャンネルが必要になる可能性がある場合です。

- 最新の更新ではなくマイナーリリースの更新のみを必要とするシステムを含む制御された環境
- 特定のパッケージセットを含むテスト環境
- 特定のバージョンの機能を要求するアプリケーションを含むシステム

### 前提条件

以下のソリューションを実装するには、**Satellite Server** が **Red Hat Network Tools** チャンネルにサブスクライブされており、**spacewalk-remote-utils** が **Satellite Server** 上にインストールされている必要があります。パッケージは **Red Hat Network Tools** チャンネルに含まれています。

1. **Satellite** サーバーに **root** としてログインします。
2. **Red Hat Satellite** の特定の更新レベルからカスタムチャンネルを作成します。

```
# spacewalk-create-channel --user=admin --server=localhost --
version=6 --update=GOLD --release=Server --arch=x86_64 --
destChannel=gold-rhel6
You have not specified a source channel, we will try to determine
it from inputs
Trying with source channel: rhel-x86_64-server-6
Creating channel, gold-rhel6, with arch x86_64 2797 packages in
source file to push.
Pushing 2797 packages, please wait.
Successfully pushed 2797 packages out of 2797

# spacewalk-create-channel -l admin -s localhost -d update1-rhel6 -D
/usr/share/rhn/channel-data/6-u1-server-x86_64
Password:
You have not specified a source channel, we will try to determine
```

```

it from inputs
Trying with source channel: rhel-x86_64-server-6
Creating channel, update1-rhel6, with arch x86_64 2857 packages in
source file to push.
Pushing 2857 packages, please wait.
Successfully pushed 2857 packages out of 2857

```

上記の設定で、

- **-IUSER, --user= USER** - サーバーへの接続に使用するユーザー名です。
- **-sSERVER, --server= SERVER** - 接続先の Satellite または Spacewalk サーバーのホスト名または IP アドレスです。デフォルトは `localhost` です。
- **-vVERSION, --version= VERSION** - 作成するチャンネルのバージョンです (例: 6、5、4)。
- **-rRELEASE, --release= RELEASE** - 作成するチャンネルのリリースです (例: AS、ES、WS、Server、Client、Desktop)。
- **-uUPDATE\_LEVEL, --update= UPDATE\_LEVEL** - 作成するチャンネルの更新レベルです (例: GOLD、U1、U2、U3、U4、U5、U6、U7、U8、U9)。ここで、GOLD は初期リリースを表します。
- **-aARCH, --arch= ARCH** - 作成するチャンネルのアーキテクチャーです (例: i386、ia64、ppc、s390、s390x、x86\_64)。
- **-dDEST\_CHANNEL, --destChannel= DEST\_CHANNEL** - 宛先チャンネルのラベルです。これは、表示されていない場合に作成されます。
- **-DDATAFILE, --data= DATAFILE** - 宛先チャンネルに移動させる RPM 一覧へのパスです。バージョン、リリース、更新、およびアーキテクチャーが指定されていない場合にのみ使用されます (オプション)。



### 注記

Red Hat Satellite 5.3 以降にのみ該当します。

## 1.2.10. ソフトウェアパッケージの削除

チャンネル内でパッケージを追加したり、削除したりするほかに、データベースおよびファイルシステムの両方からパッケージを完全に削除するオプションがあります。ファイルシステムからの削除には一時間ほどの遅れがあります。これは、**パッケージ管理** ページで行うことができます。このページには左側のナビゲーションバーにある **ソフトウェアパッケージの管理** をクリックしてアクセスします。



### 警告

データベースからパッケージを削除した場合、再びパッケージのアップロードを行えば元に戻すことはできますが、そうするとエラータとの関連付けは失われます。再ロードする際に手作業でエラータとの関連付けをやり直す必要があります。詳細は [5章 エラータ管理](#) を参照してください。

データベースからパッケージを削除するには:

1. **パッケージ管理** ページに移動し、ドロップダウンメニューからパッケージを含むオプションを選択し、**パッケージの表示** をクリックします。
2. 該当するチェックボックスを選択して **削除の確認** をクリックします。パッケージの一覧が記載された確認ページが表示されます。パッケージを完全に削除するには、**パッケージの削除** をクリックします。

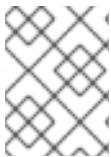


### 注記

Red Hat Satellite Proxy Server へのパッケージのアップロードに Red Hat Network Package Manager を使用している場合、実際のパッケージは Proxy Server に格納されます。カスタムパッケージは一覧表示されていても Red Hat Satellite からダウンロードすることはできません。ローカルディスクからパッケージを削除するには、**Satellite Proxy Server** にログインする必要があります。パッケージの格納場所について詳しくは、『Red Hat Satellite Proxy インストールガイド』を参照してください。

## 1.2.11. ソフトウェアチャンネルの削除

Red Hat Satellite Server や Red Hat Proxy Server の管理者の場合、未使用のチャンネルを削除することができます。この動作は **チャンネル → ソフトウェアチャンネルの管理** ページで行ないます。このページの右上にある **ソフトウェアチャンネルの削除** をクリックしてチャンネルを削除します。次のページで **チャンネルの削除** をクリックして動作を完了します。



### 注記

チャンネル → ソフトウェアチャンネルの管理 ページについては、「[ベースチャンネルの詳細](#)」で詳しく説明しています。



### 重要

チャンネルを削除しても一緒にパッケージが削除される訳ではありません。**Red Hat Satellite** からパッケージを削除する場合は、「[ソフトウェアパッケージの削除](#)」を参照してください。

Web サイトでチャンネルの削除を行なう場合は、まずは次の点をよく検討してから行なうようにしてください。

- チャンネルを削除してもそのチャンネルのパッケージはサーバー上に残ります。チャンネル削除後にパッケージも削除する方法があります。
- チャンネルを削除すると、そのチャンネルに関連するエラータの行き所がなくなり孤立する可能性があります。
- **Satellite** サーバーでは、子チャンネルがある場合には親チャンネルの削除は行なわれません。まず子チャンネルを削除してから親チャンネルを削除するようにしてください。
- チャンネルを削除する前に、キックスタートディストリビューションの関連付けの解除、またはキックスタートディストリビューションの削除を行なってください。
- **Proxy** で設定されたチャンネルが **Satellite** に接続されている場合は、**Red Hat Satellite Proxy Server** 上でチャンネルを削除してください。



## 1.2.12. カスタムパッケージのアップロードと保守

プライベートチャンネルへのパッケージのアップロードには 2 通りの仕組みがあり、使用している Red Hat Network サービスによって異なります。

Red Hat Satellite Proxy Server をご利用のお客様は **Red Hat Network Package Manager** アプリケーションを使用して頂くことになります。パッケージヘッダー情報を中央の **Satellite Server** に送り、**Red Hat Network Package Manager** を起動した Proxy のローカルリポジトリにそのパッケージ自体を格納します。

Red Hat Satellite Server をご利用のお客様は **Red Hat Network Push** アプリケーションを使用して頂くことになります。これは、パッケージのヘッダー情報をローカル Red Hat Satellite Server に送り、**Red Hat Network Push** を起動した **Satellite** のローカルリポジトリにそのパッケージを格納します。

本セクションでは、これら両方のツールについて詳しく見ていくことにします。



### 警告

Red Hat Satellite Proxy Server と Red Hat Satellite Server の両方を使用している場合は、**Red Hat Network Push** のみを使用するようにしてください。Proxy と Satellite を組み合わせて使用している場合には、カスタムパッケージおよびチャンネルのアップロードは Satellite に対してのみ行います。Proxy サーバーはここからパッケージを取得してクライアントシステム群に配信することになります。

### 1.2.12.1. Red Hat Satellite Proxy Server へのパッケージのアップロード

**Red Hat Network Package Manager** を使用すると、Red Hat Satellite Proxy Server 経由でプライベートの Satellite チャンネルに割り当てられたカスタムパッケージを提供することができます。Proxy 経由で Satellite に通信するクライアントシステムは、それらがチャンネルにサブスクライブされている場合はパッケージをダウンロードできます。Proxy 経由で Satellite に通信していないシステムは、チャンネルにサブスクライブされている場合は、yum パッケージのメタデータのみを受信し、パッケージを取得しようとする、Satellite はローカルリポジトリにパッケージコピーを持たないため、エラーが生じます。Red Hat Satellite Proxy Server で提供するパッケージを Red Hat Enterprise Linux の公式パッケージと組織が所有するパッケージのみにする場合は、**Red Hat Network Package Manager** をインストールしないようにしてください。

**Red Hat Network Package Manager** を使用する場合は、**spacewalk-proxy-package-manager** RPM パッケージとその依存パッケージをインストールします。このパッケージは登録している Red Hat Satellite Proxy Server のシステム群で使用することができます。yum install spacewalk-proxy-package-manager を実行してインストールを行います。



## 注記

Red Hat Satellite Server にアップロードされるのはパッケージのヘッダー情報のみです。Red Hat Network にクライアントシステム群のパッケージ依存関係を解決させるためにこのヘッダー情報が必要となります。実際のパッケージファイル(\*.rpm)は Red Hat Satellite Proxy Server に収納されます。このため、カスタムのパッケージは Red Hat Satellite Web サイトに表示されていてもダウンロードすることはできません。クライアントシステムにカスタムのパッケージを取得させる場合は **yum install** を使用してください。

### 1.2.12.1.1. Red Hat Network Package Manager の設定と使用

Red Hat Network Package Manager を使ってパッケージを Red Hat Satellite にアップロードする前に、そのパッケージを手作業で Proxy サーバー自体にコピーする必要があります。例えば、開発ホストから **scp** を使用します。

```
# scp foo.rpm root@rhnpkproxy.example.com:/tmp
```

Red Hat Network Package Manager を使用して Red Hat Satellite にパッケージをアップロードする場合、前のステップでサーバーにコピーしたファイルをポイントします。



## 注記

クライアントシステムにパッケージを取得させるにはチャンネルが必要になります。このため、Red Hat Satellite にカスタムのパッケージをアップロードする前に、まず、そのカスタムパッケージを割り当てるプライベートチャンネルを少なくとも1つ作成しておきます。

次のコマンドを Satellite Proxy Server で実行し、パッケージヘッダーを Red Hat Satellite Server にアップロードし、パッケージを Satellite Proxy Server リポジトリにコピーします。

```
# rhn_package_manager -c label_of_private_channel pkg-list
```

*label\_of\_private\_channel* は、これらのパッケージの受信用に作成されたカスタムチャンネルです。作成時に指定した正確なチャンネルラベルを必ず使用してください。1つまたは複数のチャンネルを指定すると (**-c** または **--channel** を使用)、アップロードしたパッケージのヘッダーは指定した全チャンネルにリンクされます。チャンネルを指定しないと、そのパッケージは **パッケージの管理** ページの **チャンネルがありません** のセクションに置かれます。パッケージを再割り当てする方法については「[パッケージのソフトウェアチャンネルへの割り当て](#)」を参照してください。

*pkg-list* はアップロードするパッケージの一覧を表示します。これらのパッケージはすでに Proxy ホストに物理的にコピーされていなければなりません。別の方法として、**-d** オプションを使用してチャンネルに追加するパッケージを含んだローカルのディレクトリを指定することもできます。Red Hat Network Package Manager は標準入力からパッケージの一覧を読み取ることもできます (**-stdin** を使用)。

Red Hat Satellite Server の URL、HTTP プロキシのユーザー名とパスワード (HTTP プロキシに認証を必要とする場合)、およびパッケージが存在する上部ディレクトリなど、その他のオプションは設定ファイルに指定されています。この特殊な設定は **編集しないでください**。またこの設定は **/etc/rhn/default/rhn\_proxy\_package\_manager.conf** に格納されています。このデフォルト設定ファイル内に指定されているオプションの値は、メインの設定ファイル **/etc/rhn/rhn.conf** の設定値や Red Hat Network Package Manager に渡すコマンドラインオプションなどで上書きすることができます。

このファイルに設定されていないパラメーターについては、現在ログインしているユーザーのホームディレクトリーにある `.rhn_package_manager` から読み込まれます。また、ここにもない場合は最終的には `/etc/rhn/rhn_package_manager.conf` から読み込まれます。これらのファイルが他の人から読み取られないよう必ず適切なパーミッションを持たせるようにしてください。

パッケージをアップロードしたらローカルディレクトリーが **Red Hat Satellite Server** のチャンネルイメージと同期しているかどうか確認します。

```
# rhn_package_manager -s -c name_of_private_channel
```

この `-s` オプションを使用することで、不足している全パッケージが一覧表示されます。このパッケージは、**Red Hat Satellite Server** にはアップロードされているものの、ローカルのディレクトリーにはないパッケージになります。このオプションを使用する場合は組織の管理者になる必要があります。アプリケーションにより **Red Hat Satellite** のユーザー名とパスワードの入力が求められます。

`--copyonly` オプションは引数に記載されているファイルを **Satellite** にはアップロードせずに指定のチャンネルにコピーします。**Red Hat Satellite Proxy Server** 上のチャンネルにパッケージが1つ不足しているものの、このチャンネル内の全パッケージ群の再インポートを行ないたくない場合に便利です。

```
# rhn_package_manager -c channel-name --copyonly /path/to/missing/file
```

パッケージ一覧は **Red Hat Satellite Server** に保存してあるため **Red Hat Network Package Manager** を使用してもチャンネル内のパッケージ一覧を検索することができます。

```
# rhn_package_manager -l -c name_of_private_channel
```

`-l` オプションを使用すると指定したチャンネル内にある各パッケージのパッケージ名、バージョン番号、リリース番号、アーキテクチャー、およびチャンネル名が表示されます。他のオプションについては「[Red Hat Network Package Manager の設定と使用](#)」を参照してください。

**Red Hat Network Package Manager (rhn\_package\_manager)** の全コマンドラインオプションの要約については、「[Red Hat Network Package Manager の設定と使用](#)」をご覧ください。

表1.1 rhn\_package\_manager オプション

オプション	説明
<code>-v, --verbose</code>	標準の出力メッセージの詳細レベルが冗長になります。
<code>-d, --dir DIRECTORY_NAME</code>	このディレクトリーからのパッケージを処理します。
<code>-c, --channel CHANNEL_NAME</code>	パッケージを検索するチャンネルを指定します。 <code>-c</code> を複数回使用すると複数のチャンネルを指定することができます (例、 <code>-c channel_one -c channel_two</code> )。
<code>-n, --count NUMBER</code>	呼び出しごとに指定したヘッダー数を処理します - デフォルトは 32 です。
<code>-l, --list</code>	指定したチャンネルのパッケージを一覧表示します。



オプション	説明
<b>-s, --sync</b>	ローカルのディレクトリーがサーバーと同期しているかどうか確認します。
<b>-p, --printconf</b>	現在の設定を表示して終了します。
<b>--newest</b>	サーバーにあるパッケージより新しいパッケージのみをプッシュします。ソースパッケージはバージョン同士の比較が行われないという点で特殊となります。新しいかどうかの定義は関連するバイナリパッケージに依存します。このオプションを <b>Red Hat Network Package Manager</b> とソースパッケージだけで使用すると、パッケージのアップロードは行なわれますが、関連バイナリパッケージがアップロードされるまでソースパッケージは <b>Red Hat Satellite Web</b> インターフェースに表示されません。 <b>--source</b> と比較して見てください。 <b>--source --newest</b> を一緒に使用すると、単独ソースパッケージが新しいパッケージで <b>更新される</b> ので、関連バイナリパッケージを先にアップロードしておく必要はありません。
<b>--source</b>	指示されたソースパッケージをアップロードします。この場合、ソースパッケージはプレーンで単独のパッケージとして扱われ、別途すでに存在するバイナリパッケージと関連する特殊なソースパッケージとしては <b>扱われません</b> 。例えば、通常のソース制御管理とは別に、開発者やテスターに向けてアプリケーションソースを配布したい場合などに使用できます。
<b>--stdin</b>	標準出力からパッケージ名を読み込みます。
<b>--nosig</b>	パッケージに署名がない場合も失敗しません。
<b>--no-ssl</b>	SSL をオフにします (推奨できません)。
<b>--stdin</b>	標準出力からパッケージ名を読み込みます。
<b>--username USERNAME</b>	<b>Red Hat Satellite</b> のユーザー名を指定します。指定しないと有効なチャンネル管理者のユーザー名の入力が必要です。
<b>--password PASSWORD</b>	<b>Red Hat Satellite</b> のパスワードを指定します。指定しないと有効なチャンネル管理者のパスワードの入力が必要です。
<b>--dontcopy</b>	アップロード後の手順で、パッケージをパッケージツリー内の最終配置場所にコピーしません。
<b>--copyonly</b>	パッケージのコピーのみを行い再インポートは行いません。
<b>--test</b>	プッシュするパッケージの一覧を出力するだけです。
<b>-, --help</b>	オプション一覧のヘルプ画面を表示します。

オプション	説明
<code>--usage</code>	使用可能なオプションの簡単な説明を表示します。
<code>--copyonly</code>	パッケージのコピーのみを行います。



### 注記

これらのコマンドラインオプションは **rhnpush** の man ページ (`man rhnpush`) でも説明されています。

## 1.2.12.2. Red Hat Satellite Server へのパッケージのアップロード

**Red Hat Network Push** アプリケーションを使用すると、Red Hat Satellite Server 経由でプライベートの Red Hat Network チャンネルに関連付けられたカスタムパッケージを提供することができます。Red Hat Satellite Server で提供するパッケージを Red Hat Enterprise Linux の公式パッケージのみにする場合は、**Red Hat Network Push** をインストールしないようにしてください。

**Red Hat Network Push** を使用する場合は、**rhnpush** パッケージとその依存パッケージをインストールします。このパッケージは登録している Red Hat Satellite Server のシステム群で使用することができます。`yum install rhnpush` を実行してインストールを行います。

**Red Hat Network Push** により RPM ヘッダー情報が Red Hat Satellite Server のデータベースにアップロードされ、その RPM は Red Hat Satellite Server のパッケージリポジトリに格納されます。Red Hat Satellite Proxy Server の **Red Hat Network Package Manager** とは異なり、**Red Hat Network Push** はパッケージ情報だけでなくヘッダーであっても Red Hat Satellite Server のデータベースから外部に配布することはありません。



### 注記

Satellite のインストールで Solaris OS のシステムのサポートを有効にしている場合は、Solaris クライアントから **Red Hat Network Push** を使用して Solaris パッケージのコンテンツを Solaris のカスタムチャンネルにアップロードすることができます。

### 1.2.12.2.1. Red Hat Network Push アプリケーションの設定

**Red Hat Network Push** のインストール時に中央設定ファイルが `/etc/sysconfig/rhn/rhnpushrc` にインストールされます。このファイルには「[Red Hat Network Push アプリケーションの設定](#)」に記載されているすべてのオプションの値が含まれています。

**rhnpush** コマンドを発行するディレクトリーに応じて設定を変更する場合、複数の異なる設定ファイルがあると便利です。現在のディレクトリー内の設定値 (`./rhnpushrc`) は、ユーザーのホームディレクトリー内の設定値 (`~/.rhnpushrc`) より優先され、中央設定ファイル (`/etc/sysconfig/rhn/rhnpushrc`) 内の設定値より先に使用されます。

例えば、現在のディレクトリーの設定ファイルは、以下を指定する場合に使用します。

- 移植するソフトウェアチャンネル
- 呼び出すユーザー名を組み込むためのホームディレクトリーの設定ファイル
- パッケージを受け取るサーバーを識別するための中央設定ファイル

「Red Hat Network Push アプリケーションの設定」には **rhnpush** コマンドのすべてのコマンドラインオプションが記載されています。

表1.2 rhnpush オプション

オプション	説明
<b>-v --verbose</b>	詳細レベルが冗長になります。 <b>-vv</b> 、 <b>-vvv</b> などのようにオプションは複数回使用することができます。
<b>-d, --dir DIRECTORY</b>	このディレクトリーからのパッケージを処理します。
<b>-c, --channel=CHANNEL_LABEL</b>	パッケージを受け取るチャンネルを指定します。チャンネルラベルの指定は必須となります。チャンネルラベルはチャンネル名とは異なります。 <b>-c</b> を複数回使用することで複数のチャンネルを指定することができます (例、 <b>-c=CHANNEL_ONE -c=CHANNEL_TWO</b> )。
<b>-n, --count N_HEADERS_PER_CALL</b>	呼び出しごとに処理するヘッダー数です。整数にしてください。デフォルトは 25 です。
<b>-l, --list</b>	指定したチャンネルのみを表示します。
<b>-r, --reldirRELATIVE_DIRECTORY</b>	各ファイルにこの相対ディレクトリーを関連付けます。
<b>-o, --orgidORGANIZATION_ID</b>	組織や企業の ID 番号を組み込みます。整数にしてください。
<b>-u, --username USERNAME</b>	指定したチャンネルに管理アクセス権を持つユーザーの Red Hat Satellite ユーザー名を組み込みます。ユーザー名を指定しないと <b>rhnpush</b> により有効なチャンネル管理者のユーザー名の入力求められます。ユーザー名とパスワードは一定期間 <b>~/.rhnpushcache</b> にキャッシュされます。デフォルトは 5 分です。新しいユーザー名とパスワードを強制する場合は <b>--new-cache</b> を使用します。
<b>-p, --password PASSWORD</b>	指定したチャンネルに管理アクセス権を持つユーザーの Red Hat Satellite パスワードを組み込みます。パスワードを指定しないと <b>rhnpush</b> により有効なチャンネル管理者のパスワードの入力が求められます。ユーザー名とパスワードは一定期間 <b>~/.rhnpushcache</b> にキャッシュされます。デフォルト値は 5 分です。新しいユーザー名とパスワードを強制する場合は <b>--new-cache</b> を使用します。
<b>-s, --stdin</b>	標準入力からパッケージ一覧を読み込みます。例えば、パイプされた <b>ls</b> コマンドなど。
<b>-X, --exclude GLOB</b>	この glob 式と一致するパッケージを除きます。

オプション	説明
<b>--force</b>	現在、チャンネル内にその名前とバージョンのパッケージが存在する場合でもパッケージのアップロードを強制します。このオプションを指定しないと既存のパッケージのアップロードはエラーを返すことになります。
<b>--nosig</b>	パッケージに署名がない場合も失敗しません。
<b>--new-cache</b>	<b>Red Hat Network Push</b> にキャッシュされているユーザー名とパスワードを破棄させ、新しいユーザー名とパスワードを受け取るか、またはその入力を求めるよう強制します。これは、最初にユーザー名とパスワードを間違えて入力した場合に便利です。
<b>--newest</b>	サーバーにあるパッケージより新しいパッケージのみをプッシュします。ソースパッケージはバージョン同士の比較が行われないという点で特殊となります。新しいかどうかの定義は関連するバイナリパッケージに依存します。このオプションを <b>Red Hat Network Push</b> とソースパッケージだけで使用すると、パッケージのアップロードは行なわれますが、関連バイナリパッケージがアップロードされるまでソースパッケージは <b>Red Hat Satellite Web</b> インターフェースに表示されません。 <b>--source --newest</b> と比較してみてください。 <b>--source --newest</b> を一緒に使用すると、単独ソースパッケージが新しいパッケージで更新されるので、関連バイナリパッケージを先にアップロードしておく必要はありません。
<b>--header</b>	ヘッダーのみをアップロードします。
<b>--source</b>	指示されたソースパッケージをアップロードします。この場合、ソースパッケージはプレーンで単独のパッケージとして扱われ、別の既存のバイナリパッケージと関連する特殊なソースパッケージとしては扱われません。例えば、通常のソース制御管理とは別に、開発者やテスターに向けてアプリケーションソースを配布したい場合などに使用できます。
<b>--server SERVER</b>	パッケージのアップロード先となるサーバーを指定します。現在、 <b>http://localhost/APP</b> の値が必要です。このパラメータは必須です。
<b>--test</b>	プッシュするパッケージの一覧のみを表示し、実際のプッシュは行いません。
<b>-h, --help</b>	オプションの簡潔な説明を表示します。
<b>-, --usage</b>	使用法の要約を表示します。

**注記**

これらのコマンドラインオプションについては **rhnpush** の **man** ページ (**man rhnpush**) でも説明されています。

**1.2.12.2.2. Red Hat Network Push アプリケーションの使用****注記**

クライアントシステムにパッケージを取得させるにはチャンネルが必要になります。このため、カスタムパッケージをアップロードする前に、そのカスタムパッケージを受け取るプライベートチャンネルを少なくとも1つ作成しておきます。

次のコマンドでパッケージヘッダーを **Red Hat Satellite Server** にアップロードし、パッケージを **Red Hat Satellite Server** のパッケージリポジトリにコピーします。

```
# rhnpush -c label_of_private_channel pkg-list
```

**Red Hat Network Push** 設定ファイルの設定値は、コマンドライン上でオプションと値を指定すると上書きすることができます。

```
# rhnpush -c label_of_private_channel --server=localhost pkg-list
```

**label\_of\_private\_channel** は、これらのパッケージの受信用に作成されたカスタムチャンネルです。作成時に指定した正確なチャンネルラベルを必ず使用してください。1つまたは複数のチャンネルを指定すると (**-c** または **--channel** を使用)、アップロードしたパッケージのヘッダーは指定した全チャンネルにリンクされます。チャンネルを指定しないと、そのパッケージは **パッケージの管理** ページの **チャンネルがありません** のセクションに置かれます。パッケージを再割り当てする方法については「[パッケージのソフトウェアチャンネルへの割り当て](#)」を参照してください。

**--server** オプションはパッケージのインストール先となるサーバーを指定するため必須となります。**Red Hat Network Push** は外部のシステムにインストールしても構いませんが、**Red Hat Network Push** の実行は **Red Hat Satellite Server** 上でローカルに行なうことをお勧めします。

**pkg-list** 参照はアップロードするパッケージの一覧を表示します。別の方法として、**-d** オプションを使用してチャンネルに追加するパッケージを含むローカルのディレクトリを指定することもできます。**Red Hat Network Push** は標準入力からパッケージ一覧を読み込むこともできます (**--stdin** を使用)。

**1.3. 設定チャンネルの管理**

**Red Hat Satellite** には、組織の設定ファイルを管理する機能が搭載されています。管理対象のファイルには、設定チャンネル内の中央で管理されるファイルと個別のシステムプロファイルでローカルに管理されるファイルが含まれます。これらのファイルを閲覧できるのは、設定管理者または **Satellite** 管理者のみです。

これにより、組織は同様の設定を複数のシステムからなるグループに配備したり、設定の変更を中央で管理したりすることができます。

中央管理のファイルとは、複数のシステムに対して使用可能なファイルのことです。中央の設定チャンネルにある単一のファイルに変更を加えることで複数のシステムに影響を与えることができます。また、ローカルの設定チャンネルもあります。**Provisioning** のエンタイトルメントを有する各システムに

は、ローカルの設定チャンネル (オーバーライドチャンネルとも呼ばれる) とサンドボックスチャンネルがあります。ローカルの設定チャンネルは **Red Hat Satellite** 内では作成されず、ローカルの設定チャンネルは、**Provisioning** エンタイトルメントが適用される各システムに対して自動的に存在します。

### 1.3.1. 設定管理のためのシステムの準備

**Satellite** 経由でシステムの設定を管理できるようにするには、システムに適切なツールと **config-enable** ファイルがインストールされている必要があります。システムを設定管理機能付きでキックスタートしている場合はとくに、これらのツールはすでにシステムにインストールされている可能性があります。インストールされていない場合は、これらのツールは **Red Hat Network Tools** の子チャンネル内にあります。以下のパッケージをダウンロードし、インストールしてください。

- **rhncfg - rhncfg-\*** のすべてのパッケージで必要となるベースライブラリと機能になります。
- **rhncfg-actions** - Red Hat Network Web サイト経由でスケジュールされた設定動作を実行するのに必要なコードになります。
- **rhncfg-client** - Red Hat Network 設定管理システムのクライアント機能に対するコマンドラインインターフェースです。
- **rhncfg-management** - Red Hat Network 設定を管理するために使用するコマンドラインインターフェースです。

### 1.3.2. 新規設定チャンネルの作成

設定チャンネルは、特定のシステムに配備する設定ファイルを格納するために作成する必要があります。新規の設定チャンネルを作成するには:

1. チャンネル管理者または組織管理者としてログインします。
2. **設定タブ**をクリックします。
3. **設定の動作**とマークされている右側のフレーム上で、**新規の設定チャンネルを作成**をクリックします。
4. 以下のフィールドに入力します。
  - **名前**
  - **ラベル** このフィールドには、英数字、「-」、「\_」、および「.」のみを入力してください。
  - **詳細** 詳細を入力してください。このフィールドには、このチャンネルを他と区別するための簡単な情報を入力できます。
5. **設定チャンネルの作成** をクリックします。

### 1.3.3. 設定チャンネルへの設定ファイルの追加

1. チャンネル管理者または組織管理者としてログインします。
2. **設定タブ**をクリックします。
3. 左側のサブメニューで、**設定チャンネル** をクリックします。

4. 設定ファイルが追加される設定チャンネルを選択します。
5. サブタブの **ファイルの追加** をクリックします。
6. 必須のフィールドに入力します。
  - **アップロードするファイル** - 設定ファイルの最大許容サイズは **128kb** です。
  - **ファイル名/パス** - 設定ファイルの配備先とする必要のあるターゲットシステムのパスです。
  - **所有権** - ファイルを所有するユーザーおよびグループです。フィールドに追加されたユーザーおよびグループが、ファイルが配備されるシステム上に存在しない場合は、配備は失敗します。
  - **ファイルパーミッションモード** - 変更可能なユーザーを基にしたパーミッションです。テキストファイルに「**644**」、ディレクトリーおよび実行可能ファイルに「**755**」を設定することで、グローバルなアクセスまたは実行が可能になります (変更は不可)。
  - **SELinux コンテキスト** - `user_u:role_r:type_t:s0-s15:c0.c1024` などの SELinux コンテキストを入力します。
  - **マクロデリミタ** - 使用できるマクロの一覧は、次のセクションの「[設定ファイルへのマクロの組み込み](#)」に記載されています。

#### 1.3.4. 設定ファイルへのマクロの組み込み

従来のファイル管理では、ファイルの差異がわずかであってもファイルの種類が数百または数千に上る場合に、それぞれのファイルを別々にアップロードし、配信する必要がありました。Red Hat Satellite では、Provisioning のエンタイトルメントが付与されたシステムを対象に管理する設定ファイル内へのマクロや変数の組み込みを許可することによってこれに対応します。カスタムシステム情報の変数のほかに、以下の標準的なマクロがサポートされています。

- `rhn.system.sid`
- `rhn.system.profile_name`
- `rhn.system.description`
- `rhn.system.hostname`
- `rhn.system.ip_address`
- `rhn.system.custom_info(key_name)`
- `rhn.system.net_interface.ip_address(eth_device)`
- `rhn.system.net_interface.netmask(eth_device)`
- `rhn.system.net_interface.broadcast(eth_device)`
- `rhn.system.net_interface.hardware_address(eth_device)`
- `rhn.system.net_interface.driver_module(eth_device)`

この機能を使用するには、**設定チャンネルの詳細 (Configuration Channel Details)** ページから、設定ファイルのアップロードまたは作成を行います。次に、その **設定チャンネルの詳細**

(**Configuration Channel Details**) ページを開いて、選択したサポート対象のマクロを組み込みます。変数を相殺するために使用したデリミタが、**マクロ開始デリミタ (Macro Start Delimiter)** と **マクロ終了デリミタ (Macro End Delimiter)** フィールドで設定されたデリミタと一致しており、ファイル内の他の文字とは競合しないことを確認してください。デリミタの長さは 2 文字であり、パーセント (%) 記号を使用しないことをお勧めします。

例のように、IP アドレスとホスト名以外に違いがないサーバーすべてに適用できるファイルを 1 つ持つことができます。各サーバーごとに別々の設定ファイルを管理する代わりに、**server.conf** などのような IP アドレスとホスト名のマクロを含む次のような単一ファイルを作成することができます。

```
hostname={| rhn.system.hostname |}
ip_address={| rhn.system.net_interface.ip_address(eth0) |}
```

ファイルを個別システムに配信する際に、Red Hat Network Web サイトのスケジュールされた動作であるか、または **Red Hat Network Configuration Client** のコマンドライン (**rhncfg-client**) によるかにかかわらず、変数は、**Satellite** のシステムプロファイルに記録されている、システムのホスト名と IP アドレスに置き換えられます。例えば、上記の設定ファイルでは、配備されたバージョンは以下のようになります。

```
hostname=test.example.domain.com
ip_address=177.18.54.7
```

カスタムのシステム情報をキャプチャーするには、キーラベルをカスタム情報マクロに挿入します (**rhn.system.custom\_info**)。例えば、「**asset**」というラベル名のキーを作成した場合、これを設定ファイル内のカスタム情報マクロに追加して、それを含むシステムではその値が置換されるようにすることができます。

```
asset={@ rhn.system.custom_info(asset) @}
```

そのキーの値が含まれているシステムに対してこのファイルを配備すると、マクロは変換されて次のような文字列になります。

```
asset=Example#456
```

デフォルト値を含める場合は、例えばエラーを防ぐためにデフォルト値が必要な場合などは、カスタム情報マクロにそれを追加することができます。次のようになります。

```
asset={@ rhn.system.custom_info(asset) = 'Asset #' @}
```

このデフォルトは、これを含むシステムならいずれでもその値で上書きされます。

**Red Hat Network Configuration Manager (rhncfg-manager)** はシステム不問のツールであるため、ファイルを変換したり、変更したりしません。**rhncfg-manager** はシステム設定に依存しません。



#### 注記

この機能は、マクロをテキストファイルにのみ挿入します。バイナリファイルの場合は挿入できません。

### 1.3.5. 設定チャンネルへのシステムのサブスクライブ

**Red Hat Satellite** が設定ファイルを使って設定チャンネルを中央で管理できるようにするには、2 つの要件を満たす必要があります。



- システムを設定チャンネルにサブスクライブさせる必要がある。
- 設定管理はシステム上で有効にする必要がある。手順については、「[システム上での設定管理の有効化](#)」を参照してください。

システムを設定チャンネルにサブスクライブさせるには:

1. **設定** をクリックします。左側のサブメニューで、**設定チャンネル** をクリックします。
2. システムを追加する必要のある設定チャンネルを選択します。
3. チャンネルページで、**システム** → **ターゲットシステム** サブタブの順にクリックします。
4. 設定チャンネルに追加するシステムを選択し、**システムのサブスクライブ** をクリックします。

### 1.3.6. システム上での設定管理の有効化

設定管理をシステム上で有効にする前に、以下の要件を満たす必要があります。

- ターゲットシステムには、**Provisioning** のエンタイトルメントが必要です。**Provisioning** のエンタイトルメントをシステムに追加するには、『システム』の章を参照してください。
  - **Red Hat Satellite Tools** チャンネルへのサブスクリプション。子チャンネルを変更する方法については、『システム』の章を参照してください。
1. チャンネル管理者または **Satellite** 管理者としてログインします。
  2. **設定** をクリックします。
  3. **設定の動作** というマークが付けられている右側のフレーム上で、**システムで設定管理を有効にする** をクリックします。
  4. 有効にするシステムを選択します。
  5. **rhcfg-\*** パッケージのパッケージインストールをスケジュールします。これらの設定パッケージをインストールする時間を選択します。
  6. **Red Hat Satellite 設定管理を有効にする (Enable Red Hat Satellite Configuration Management)** をクリックします。
  7. 個々のシステムでターミナルコンソールを開くか、**root** としてリモートからログインします。以下の動作を実行する必要があります。
    - a. このコマンドを実行して、保留中の **rhncfg-\*** パッケージのインストールを完了します。

```
# rhn_check
```

- b. 以下のコマンドを実行して **Red Hat Network** の動作を有効にします。

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

## 第2章 組織

Red Hat Satellite は、各配備を系統別に準備した入れ物に区分することができます。この入れ物(つまり **組織**)を活用してシステムの用途や所有権、および配備されているコンテンツ別に分けて保守を行うことができます。

Red Hat Network Satellite ひとつで複数の **組織** の作成と管理に対応しています。異なるグループのシステム群、コンテンツ、およびサブスクリプションなどを区分することができます。本章では、複数の組織の作成と管理に関する基本的な概念および作業について要約しています。

**組織** の Web インターフェースでは、管理者による複数の **Satellite** 組織の表示、作成、および管理が可能です。**Satellite** の管理者は異なる組織を横断してソフトウェアやシステムのエンタイトルメントを割り当てることができます。また、任意の組織によるシステム管理作業へのアクセス権を制御することもできます。

新しい組織を作成してその組織用の管理者やエンタイトルメントの割り当てを行うのが **Satellite** 管理者です。担当する組織のグループ、システム、およびユーザーの割り当てを行うのが組織の管理者です。こうした分担を行うことにより、組織単位の管理作業は他の組織に影響を与えることなくその組織の管理者で行うことができますようになります。

**組織** のページには、**Satellite** 全体での組織一覧が表示され、各組織に割り当てられたユーザー数とシステム数も表示されます。**組織** のページには、組織間で確立している「信頼」がある場合には**信頼** のページも備わっています。

### 2.1. 組織の作成

1. 新規の組織を作成するには、**管理 (Admin)** → **組織** → **新規の組織を作成** の順でをクリックします。
2. 所定のテキストボックスに組織名を入力します。この名前は、**3 ～ 128 文字**にしてください。
3. 以下のような情報を入力して組織の管理者を作成します。
  - 組織管理者用の **希望のログイン** を **5 ～ 64 文字**の長さで入力します。組織管理者のアカウント名にはその組織の管理ログイン名と一致する、わかりやすい名前にすることをお勧めします。
  - **希望のパスワード** を作成しそのパスワードの**確認**を行います。
  - 組織管理者の **Email** アドレスを入力します。
  - 組織管理者の **名前** と **姓** を記入します。
4. **組織の作成** ボタンをクリックしてこのプロセスを終了します。

新規の組織が作成されると、**組織** のページの一覧に新規の組織が表示されます。

**Satellite** 管理者の方には、**organization 1**の組織管理者アカウントを **Satellite** 管理用に確保しておくことをお勧めします。必要が生じた場合、組織へのログインが可能になります。



## 重要

Red Hat Network Satellite に PAM 認証が設定されている場合は、新しい組織の主要な組織管理者アカウントに PAM アカウントを使用するのは避けてください。代わりに、組織管理者用に Satellite ローカルのアカウントを作成し、Satellite ログイン用の PAM 認証アカウントは別途あまり特権を持たせずに確保します。間違った操作を行ってしまう可能性が高いため、この方法によりユーザーが上位の権限を持ったアカウントで Red Hat Network Satellite にログインしないようにします。

## 2.2. 組織別のエンタイトルメントの管理

新規の組織の作成が完了した後は、その組織へのエンタイトルメントを割り当てが重要な作業になります。以下のエンタイトルメントが必要です。

- **システムエンタイトルメント- Management**エンタイトルメントは、すべての組織が正しく機能するための基本的な要件となります。組織に割り当てられる **Management** エンタイトルメントの数は、利用可能なソフトウェアエンタイトルメントの数には関係なく、**Satellite** 上のその組織へ登録する可能性のあるシステムの最大数です。例えば、**Red Hat Enterprise Linux Client** のエンタイトルメントが **100** あるのに対し、その組織には管理システムエンタイトルメントが **50** しかない場合、この組織に登録できるシステムは **50** システムのみになります。組織で管理されるシステムの場合は、**Provisioning** も推奨されています。
- **ソフトウェアチャンネルのエンタイトルメント- Red Hat** チャンネルを使用するシステムの場合、**Red Hat Enterprise Linux Server** が必要になる場合があります。**Red Hat Network Tools** チャンネルも推奨されます。**Red Hat Network Tools** チャンネルには、設定管理やキックスタートのサポートなどの **Red Hat Satellite** の拡張機能に必要なクライアントソフトウェアのほか、**Xen** や **KVM** の仮想ゲストのエンタイトルメント数が正しくカウントされるために必要な **rhn-virtualization** パッケージなども含まれます。

### 2.2.1. 組織のシステムエンタイトルメントの変更

組織内でシステムエンタイトルメントを管理するには:

1. **管理** メニューをクリックし、**組織** を選択します。
2. 一覧から組織を選択し、**サブスクリプション** タブを選択します。ページは、**システムのエンタイトルメント** にデフォルト設定されています。
3. 組織に割り当てられる必要のあるエンタイトルメントの数に基づいて、システムエンタイトルメントの **提示合計数 (Proposed Totals)** を変更します。最大および最小の合計数は、フィールドのすぐ下に示されます。
4. すべての変更を更新するには、**組織の更新** をクリックします。

### 2.2.2. 組織のソフトウェアチャンネルのエンタイトルメントの変更

組織内のシステムエンタイトルメントを変更するには:

1. **管理** メニューをクリックし、**組織** を選択します。
2. 一覧から組織を選択し、**サブスクリプション** タブを選択します。
3. **サブスクリプション** インターフェースで、**ソフトウェアチャンネルのエンタイトルメント** タブをクリックし、すべての組織のすべてのエンタイトルメントとその使用状況を確認します。組織に割り当てる必要のある予定されたチャンネルのエンタイトルメントに応じて、**通常の提**

示合計数を変更します。最大および最小の合計数は、フィールドのすぐ下に示されます。

チャンネルエンタイトルメントは、通常または *Flex* のいずれかになります。通常のエンタイトルメントはどのシステムにでも使用できます。Flex のエンタイトルメントはサポートされている仮想化タイプのゲストとして検出されたシステムでのみ使用することができます。

4. すべての変更を更新するには、**組織の更新** をクリックします。



## 注記

カスタムチャンネルを使用できるのは、そのチャンネルを作成した組織管理者のみで、使用できるのは組織内に限られます。ただし、チャンネルを共有する予定のある複数の組織間で「組織の信頼」が確立されている場合を除きます。組織の信頼については「[組織の信頼の管理](#)」を参照してください。

## 2.3. 複数の組織の管理

Red Hat Satellite ひとつで 複数の組織の作成および管理に対応しています。これにより、異なる組織または特定グループのシステム群、コンテンツ、およびサブスクリプションなどを区分することができます。本セクションでは、Red Hat Satellite 内での複数の組織の作成と管理に関する基本的な概念およびセットアップ作業について要約します。

### 2.3.1. Satellite の複数組織向け使用のモデリング

以下の例では、複数の組織 (multi-organization) 機能を使った 2 つのシナリオを示しています。まずは、追加組織を 1 つ作成し、これを限定的なシステムやユーザーのセットに試用することで、適用しようとしている組織のプロセスおよび方針に複数組織の Satellite が与える影響について十分に理解することが推奨されます。

#### 2.3.1.1. 集中型管理の Satellite – 1 つの組織が複数の部署を持つ場合

以下の最初のシナリオでは、企業または他の組織内の 1 つの中央グループによって Red Hat Satellite の維持管理が行われています (「[集中型管理の Satellite – 1 つの組織が複数の部署を持つ場合](#)」を参照してください)。「組織 1」(Satellite の設定時に作成された管理組織)の Satellite 管理者は「組織 1」(「管理組織」)をソフトウェア、システムのサブスクリプション、エンタイトルメント用のステージングエリアとして扱います。

Satellite 管理者の責任には Satellite の設定 (Web インターフェースの **管理** 部分で行える作業)、追加の Satellite 組織の作成と削除、ソフトウェア、システムのサブスクリプション及びエンタイトルメントの割り当てと削除が含まれます。

この例では、追加の組織が企業内の複数の各部署にそれぞれマッピングされています。組織内の各部署を分割するレベルを決定する方法の 1 つとして、Red Hat Satellite と使用するために各部署が購入するサブスクリプションとエンタイトルメントを考えてみます。Satellite 内の複数の組織に対して集中型の制御を行う場合は、追加作成される各組織内にそれぞれ組織管理者アカウントを作成することにより、何らかの都合によりその組織にアクセスすることができます。

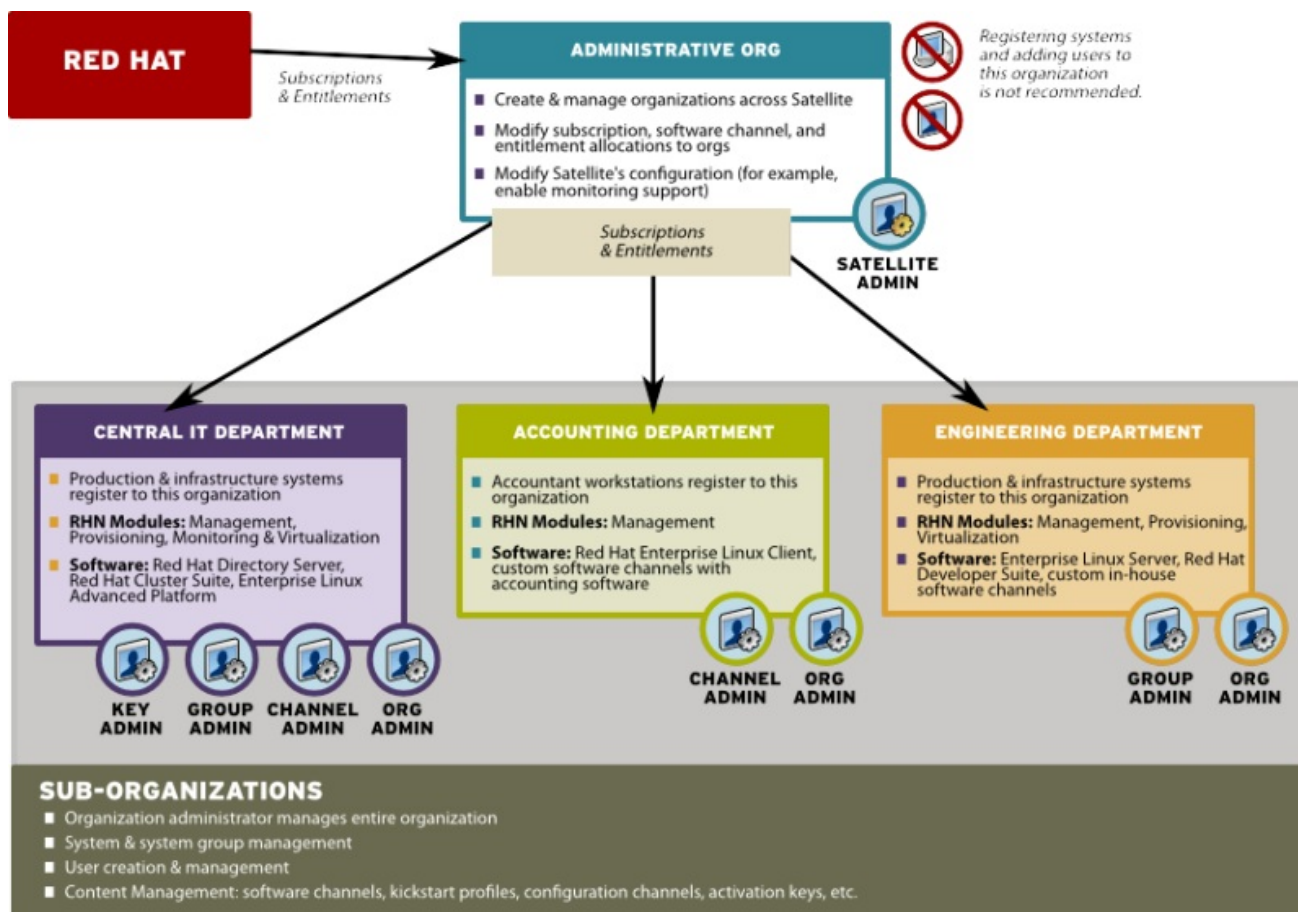


図2.1 集中型管理の Satellite – 1つの組織が複数の部署を持つ場合

### 2.3.1.2. 分散型管理 – 複数の第三者組織の場合

この例では、**Satellite** は中央のグループによって管理されていますが、各組織は **Satellite** の他の組織に対して互いに関連性や結び付きがなく別々に取り扱われます。各組織は、**Satellite** アプリケーション自体を管理するグループの顧客であるかも知れません。

同一企業のあらゆる部署となるサブの組織群で構成される **Satellite** は組織間でシステムやコンテンツの共有という点で優れた環境であるのに対し、この分散型の例では共有に適した環境とは言えません。管理者は各組織に対して特定数のエンタイトルメントを割り当てることができます。各組織はコンテンツに関するソフトウェアチャンネルのエンタイトルメントを持っていれば **Satellite** と同期している **Red Hat** の全コンテンツにアクセスできます。

ただし、任意の組織がカスタムのコンテンツをその組織にプッシュしても、そのコンテンツは他の組織では利用できません。コンテンツを各組織に再度プッシュしない限り、全組織あるいは選択した組織群に利用できるようなカスタムのコンテンツは提供できません。

このシナリオでは、**Satellite** 管理者がログインアクセスを得るため各組織にアカウントをひとつずつ確保したいとします。例えば、**Satellite** を使用して外部に向けて管理ホスティングサービスを提供する場合、その組織内のシステムにアクセスしてコンテンツをプッシュできるよう自分用のアカウントを確保することができます。



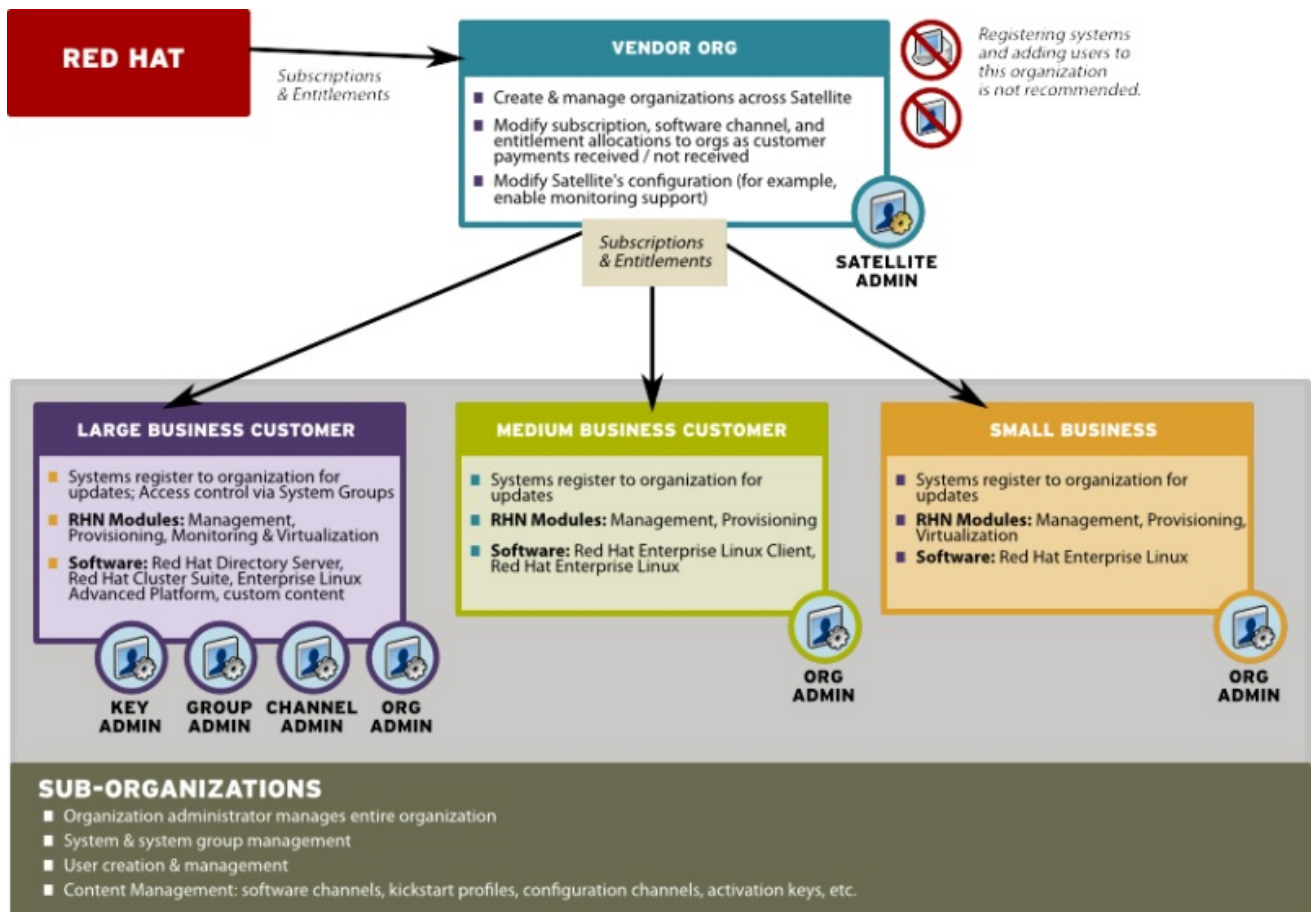


図2.2 分散型 Satellite 管理 – 複数の部署から構成される組織

### 2.3.1.3. 複数組織向け使用に関する推奨事項

複数組織向けの Satellite の管理に選択するモデルには関係なく、以下の推奨事項に留意してください。

管理組織 (組織 #1) は、以下を実行する場合を除き、どのような状況であってもシステムの登録や、ユーザーの作成に使用することは推奨できません。

- 単一の組織向け Satellite として使用する。
- Satellite が単一組織向け Satellite から複数組織向け Satellite に移行中である。

これは、以下の理由によります。

1. 管理組織はエンタイトルメントに関連した特殊ケースとして扱われます。Satellite にある他の組織でエンタイトルメントを追加したり削除したりすることによってのみ暗示的にこの管理組織に対するエンタイトルメントの追加または削除を行えます。
2. 管理組織は、サブスクリプションやエンタイトルメント用のステージングエリアとなります。Satellite を新しい証明書に関連付けると、新しいエンタイトルメントがデフォルトでこの組織に与えられます。この新しいエンタイトルメントを Satellite 上の他の組織に使用できるようにするには、エンタイトルメントを管理組織から明示的に他の組織に割り当てる必要があります。
3. Satellite サーバーには、Satellite 証明書にあるエンタイトルメントの数分のシステムを組み込むことができます。Satellite 上の各組織のエンタイトルメントの使用に関して評価を行い、各組織が適切に機能するために必要なエンタイトルメントの数を判断します。それぞれの組織管

理者は、エンタイトルメントの制約に留意し、システムプロファイルを必要に応じて管理する必要があります。万一問題があった場合には、**Satellite** 管理者はエンタイトルメントに関する懸念を調整するために介入することができます。



### 注記

**Satellite** 管理者としてログインすると、組織に割り当て済みのエンタイトルメントを、組織がシステムプロファイルに正しく関連付けたエンタイトルメントの数よりも少なくすることができません。

## 2.3.2. 組織内のシステム群の設定

組織を作成し必要なエンタイトルメントを割り当てたら、システムを各組織に割り当てることができるようになります。

特定の組織に対してシステムを登録する場合、基本的方法が2つあります。

1. ログインとパスワードを使用して登録する: 特定の組織用に作成したログインとパスワードを入力すると、その特定組織にシステムが登録されます。例えば、**user-123** が **Satellite** 上の **Central IT** 組織のメンバーである場合、次のコマンドはいずれのシステムで使用してもそのシステムを **Satellite** の **Central IT** 組織に登録させることになります。

```
# rhnreg_ks --username=user-123 --password=foobaz
```



### 注記

**rhnreg\_ks** の **--orgid** パラメーター (Red Hat Enterprise Linux 5 用) は、**Satellite** の登録や Red Hat Satellite の複数組織のサポートには関係ありません。

2. アクティベーションキーを使用して登録する: 組織のアクティベーションキーを使用してその組織にシステムを登録することもできます。アクティベーションキーはそのアクティベーションキーの作成元である組織へシステムを登録します。ユーザーに組織へのログインアクセスは与えずにその組織へのシステム登録を許可したい場合などに使用すると便利な登録方法です。組織間でシステムを移動したい場合も、アクティベーションキーを使用したスクリプトでシステムの移動を自動化することができます。

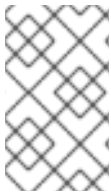


### 注記

アクティベーションキーは **Red Hat Satellite 5.1.0** 以降より新しい形式となり、アクティベーションキーの先頭の数文字がそのアクティベーションを所有する組織を表わすために使用されます (ID 番号)。

## 2.3.3. 組織の信頼の管理

組織は、**Satellite** 内に *組織の信頼関係* を確立することによって、リソースを相互に共有することができます。組織の信頼関係が双方向となる、つまり、**Satellite** 管理者が複数の組織間で信頼を確立すると、各組織の組織管理者は必要最大限または最小限のリソースを自由に共有することができるようになります。どのリソースを共有の対象とするか、信頼関係にある他の組織からの共有リソースの中でどれを使用するのかは、各組織管理者が決定します。



## 注記

カスタムのコンテンツを共有することができるのは組織管理者のみになります。**Satellite** 管理者は各組織へのシステムとソフトウェアエンタイトルメントの割り当てを行うだけです。

### 2.3.3.1. 組織間の信頼の作成

**Satellite** 管理者は、複数の組織間に信頼を作成することができます。**管理** のメインページでサイドメニューにある **組織** のリンクをクリックします。

複数ある組織の中の1つの組織の名前をクリックして **詳細** ページ内の **信頼** サブタブをクリックします。

**信頼** サブタブ上には、Red Hat Satellite のすべての「信頼」が表示されます。ここで、**組織別に分類 (Filter by Organization)** テキストボックスを使用して表示させる組織数を目的のグループに絞り込みます。

現在の組織と組織間の信頼を持たせたい組織名の横のチェックボックスをクリックしてから **信頼の変更** ボタンをクリックします。

### 2.3.3.2. 「信頼」内の複数の組織間でのコンテンツチャンネルの共有

組織間の信頼を作成したら、カスタムのソフトウェアチャンネルなどのコンテンツを信頼関係にある他の組織と共有できるようになります。また、チャンネルのアクセスについて詳細な制御ができるよう各チャンネルに適用できるチャンネル共有レベルが3通りあります。



## 注記

Red Hat チャンネルはそのエンタイトルメントを有する全組織で利用できるため、組織間での共有はできません。

別の組織とカスタムチャンネルを共有するには、以下の手順を実行します。

1. 組織管理者のユーザー一名で **Satellite** にログインします。
2. チャンネル → ソフトウェアチャンネルの管理の順にクリックします。
3. 他の組織と共有したいカスタムチャンネルをクリックします。
4. **詳細** ページの **チャンネルアクセス制御** セクションには **組織の共有 (Organizational Sharing)** 内に共有に関して3種類の選択肢があります。
  - **プライベート** - チャンネルをプライベートにすると、チャンネルの所有者以外はこの組織からもチャンネルにアクセスできなくなります。
  - **保護** - 選択した特定の信頼できる組織にチャンネルへのアクセスを許可します。



## 注記

**保護** 共有を選択すると、別のページが表示され **アクセスを許可して確認** をクリックすることによって組織へのチャンネルのアクセス権を許可する確認が求められます。



- **パブリック (Public)** - 信頼関係内にある全組織にカスタムチャンネルへのアクセスを許可します。

選択したレベルの横にあるラジオボタンをクリックして **チャンネルの更新** をクリックします。

これで、カスタムチャンネルへのアクセスを許可した信頼関係にある組織の組織管理者は、その組織内のクライアントシステムに共有チャンネルからのパッケージをインストールして更新させることができます。



#### 注記

共有チャンネルにサブスクライブさせているシステムがあり、その共有チャンネルの組織管理者がチャンネルへのアクセス権を変更してしまうと、システムはそのチャンネルを失うことになります。管理者がベースチャンネルのアクセス権を変更すると、システムは **システム** ページにベースチャンネルがなくなるため更新を受信しなくなります。

### 2.3.3.3. 信頼できる組織内でのシステムの移行

ソフトウェアチャンネルの共有のほかに、信頼関係にある組織はシステムを他の信頼できる組織に移行することができます。これを実行するには、**Satellite Web** インターフェースから行う方法と、**migrate-system-profile** というユーティリティを使用して行う方法の2種類があります。



#### 注記

**Satellite** 管理者は、信頼できる組織から「信頼」内のいずれの組織にシステムを移行させることができます。一方、組織管理者ができるのは自らの組織から「信頼」内の別の組織への移行のみになります。

#### 2.3.3.3.1. Satellite インターフェースを使用したシステムの移行

##### 手順2.1 組織間でのシステムの移行

1. **Systems** タブをクリックした後、移行するシステムの名前をクリックします。
2. **詳細 → 移行** とクリックし、システムの移行先となる組織の名前を選択します。
3. **システムの移行** をクリックします。

#### 2.3.3.3.2. migrate-system-profile の使用

**migrate-system-profile** はコマンドラインでの使用になります。移動対象と移動先の組織を指定するため **systemID** と **orgID** を引数として使用します。

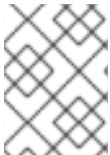
**migrate-system-profile** コマンドを使用する場合は、**spacewalk-utils** パッケージをインストールしておく必要があります。**migrate-system-profile** を使用するのに **Satellite** サーバーにログインする必要はありません。ただし、ログインしない場合はサーバーのホスト名か IP アドレスをコマンドラインスイッチとして指定する必要があります。



## 注記

**migrate-system-profile** コマンドを使用してシステムを移行する場合、移行前の組織でシステムが持っていたエンタイトルメントやチャンネルのサブスクリプションなどは移行されません。ただし、そのシステムの履歴は保存されるので、新しい組織管理者はこの履歴にアクセスして、ベースチャンネルへのサブスクライブやエンタイトルメントの付与などの移行プロセスを簡略化することができます。

移行するシステムの ID とシステムの移行先となる組織の ID、および別のマシンからコマンドを実行する場合は **Satellite** サーバーのホスト名または IP アドレスを確認します。



## 注記

Web UI または **spacewalk-report** ツールのいずれかを使用して、システム ID と組織 ID を見つけます。

このデータを取得した後のコマンドラインからの使用は以下のようになります。

```
# migrate-system-profile --satellite {SATELLITE HOSTNAME OR IP} --
systemId={SYSTEM ID} --to-org-id={DESTINATION ORGANIZATION ID}
```

### 例2.1 ある部署から別の部署への移行

経理部はエンジニアリング部からワークステーションの移行を希望しているものの、経理部の組織管理者には **Red Hat Satellite** サーバーへのシェルアクセスがない場合は、以下のデータを使用します。

- 経理部の組織 ID は **2**
- ワークステーションのシステム ID は **10001020**
- **Red Hat Satellite** ホスト名は **satserver.example.com**

経理部の組織管理者は、シェルスクリプトから以下のように入力を行います。

```
# migrate-system-profile --satellite=satserver.example.com --
systemId=10001020 --to-org-id=2
```

経理部の組織管理者は、ユーザー名とパスワードの入力を求められます (コマンドラインで **--username=** と **--password=** を使って指定しなかった場合)。

この後、**Red Hat Satellite** の Web インターフェースにログインすると経理部の組織管理者は **システム** ページでこのシステムが見れるようになります。経理部の組織管理者は、他のシステムを組織に登録したときと同様に、このクライアントにベースチャンネルを割り当ててエンタイトルメントを付与し、移行プロセスを完了します。これらはシステムの **イベント** サブタブ内の **履歴** ページで確認することができます。

一度に複数のシステムを移行する必要がある場合、**Satellite** 管理者は **migrate-system-profile** の **--csv** オプションを使用して、移行するシステムをすべてコマンドで区切って記載した一覧でこのプロセスを自動化することができます。

以下のような形式で、移行するシステムの ID のほかに、移行先の組織の ID を CSV ファイル内の 1 行に含める必要があります。

```
systemId,to-org-id
```

適切な CSV は以下のような記述になるはずです。

```
1000010000,3  
1000010020,1  
1000010010,4
```

**migrate-system-profile** の使い方については、**man migrate-system-profile** と入力して **man** ページを参照するか、または **migrate-system-profile -h** と入力して基本的なヘルプ画面を参照します。

## 第3章 システムのプロビジョニング

### 3.1. RED HAT SATELLITE を使用したプロビジョニング

プロビジョニングとは、物理また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 Satellite** は、キックスター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 台また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、または仮想マシンのいずれかになります。**Red Hat Satellite** の仮想マシンの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。

システム管理者は、キックスターインストールによって、管理するマシンに **Red Hat Enterprise Linux** をインストールする自動インストールを使用することができます。キックスターを使用すると、システム管理者は標準のインストール時に通常尋ねられるすべての質問への回答が含まれた単独ファイルを作成できます。

キックスターファイルは単独のサーバーシステム上で維持して、インストール時に個別のコンピューターによって読み込まれるようにできます。このインストール方法は、単独のキックスターファイルを使用して複数のマシンに **Red Hat Enterprise Linux** をインストールすることをサポートします。

ベースイメージや、キックスターファイル、およびその他のコンテンツは、**Satellite** サーバー URL を使って、HTTP からアクセスできます。たとえば、**Satellite** サーバーにある 64 ビット用の **Red Hat Enterprise Linux 6** のキックスターファイルにアクセスするには、**http://satellite.example.com/ks/dist/ks-rhel-x86\_64-server-6-6.4/GPL** のように、ベース URI の **http://satellite.example.com/ks/dist/ks-rhel-x86\_64-server-6-6.4** の後にダウンロードするパッケージ名が続きます。

『**Red Hat Enterprise Linux** インストールガイド』には、キックスターについての詳細な記載があります。

#### 定義

本セクション全体にわたって使用されている用語

#### キックスター

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 Enterprise Linux** 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、キックスタープロファイル (*Kickstart profile*) と呼ばれています。

#### キックスタープロファイル

キックスターファイルは、マシンのキックスター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** のキックスターへの機能強化をベースに構築されているため、**Red Hat Satellite** ではキックスタープロファイルは従来の **Anaconda** キックスターの定義のスーパーセットとなります。キックスタープロファイルはキックスターツリーが存在することを前提としています。

#### キックスターツリー

マシンをキックスター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ー構造とファイルです。Cobbler の用語では、キックスター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

Red Hat 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。

### 3.1.1. キックスタートの流れ

マシンにネットワークベースのキックスタートを行う場合、次のようなイベントが順次発生していくことになります。

1. ネットワーク上に配置して電源をオンにすると、マシンの PXE 論理がその MAC アドレスと発見されるべき要求をブロードキャストします。
2. 静的 IP アドレスを使用しない場合は、DHCP サーバーがその発見要求を認識してから新しいマシンを起動するために必要となるネットワーク情報を提供します。これには、IP アドレス、使用されるデフォルトのゲートウェイ、ネットワークのネットマスク、ブートローダープログラムを格納している TFTP または HTTP サーバーの IP アドレス、そのプログラムのフルパスとファイル名 (サーバーの root に相対的) などが含まれます。
3. マシンはネットワーキング情報を適用してブートローダープログラムを要求するためにサーバーとのセッションを開始します。
4. ブートローダーはロードされると、ブートローダー自体がロードされたサーバーからその設定ファイルを検索します。このファイルは、初期 RAM ディスク (initrd) イメージなどのブートしているマシン上で実行されるべきカーネルおよびカーネルオプションを規定します。ブートローダープログラムが SYSLINUX だとすると、このファイルは、サーバーの **pxelinux.cfg** ディレクトリー内にあり、新しいマシンの IP アドレスにあたる 16 進数の名前が付けられます。例えば、Red Hat Enterprise Linux 6 のブートローダー設定ファイルは次のようになります。

```
default=0
```

```
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.32-279.22.1.el6.x86_64)
root (hd0,0)
kernel /vmlinuz-2.6.32-279.22.1.el6.x86_64 ro root=/dev/sda2
crashkernel=auto ks=http://example.com/ks.cfg
initrd /initramfs-2.6.32-279.22.1.el6.x86_64.img
```

5. マシンは初期化イメージとカーネルを受け取り解凍すると、カーネルを起動して、キックスタータ設定ファイルを格納しているサーバーを含むブートローダー設定ファイル内にあるオプションを指定してキックスターインストールを開始します。
6. 次にこのキックスタータ設定ファイルがマシンにインストールファイルの場所を指示します。
7. 新しいマシンはキックスタータ設定ファイル内で設定されるパラメーターに基づいて構築されます。

### 3.1.2. 前提条件

**Provisioning** エンタイトルメントはシステムのプロビジョニングが機能するために必要です。このエンタイトルメントがないと、**Provisioning/Kickstarting** のタブは **Satellite** に表示されません。

キックスタータを処理するには、組織のインフラストラクチャーも準備する必要があります。キックスタータプロファイルの作成前に、以下の点を考慮することができます。

- **DHCP サーバー**。これはキックスタータには不要ですが、**DHCP** サーバーはキックスタータファイル内でのネットワーク設定の必要性を軽減します。また、ネットワークから起動することもできます。**DHCP** サーバーがなく、静的 IP アドレスを使用している場合、起動プロファイルの開発中に静的 IP を選択することが推奨されます。
- **FTP サーバー**。**HTTP** 経由でキックスタータディストリビューションツリーをホストする代わりに、**FTP** サーバーを使用することができます。

ベアメタルからのキックスタータを実行している場合は、以下を行います。

1. **DHCP** を設定して、必要なネットワーキングパラメーターとブートローダープログラムの場所を指定します。
2. ブートローダー設定ファイル内で使用するカーネルと適切なカーネルオプションを特定します。

#### 3.1.2.1. 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 (RHN)** 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。

Red Hat 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、アーキテクチャー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3.1 アーキテクチャー別の必要なディストリビューションファイル

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

### 3.1.2.2. 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3.1 キックスターツリーの自動インストール

Red Hat Satellite 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** を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 **Red Hat Network Tools** のチャンネルを探します。

例えば、**x86** アーキテクチャーを採用した **Red Hat Enterprise Linux 6** を使用する場合は、**rhel-x86\_64-server-6** チャンネルとそれに対応する **Red Hat Network Tools** チャンネルである **rhn-tools-rhel-x86\_64-server-6** 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** コマンドを使って **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3.2 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するには、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** 内にディレクトリーを作成することが奨励されます。例



例えば、Red Hat Enterprise Linux 6 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-x86_64-server-6-beta/` にコピーします。

3. Red Hat 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。上記で利用した例では、**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-6-beta' \ -d
/var/satellite/custom-distro/rhel-x86_64-server-6-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ーが異なります。

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-x86_64-server-6-
beta/Server/*.rpm
```

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なります。

6. Red Hat 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ーの生成 (例: **Red Hat Enterprise Linux 6**) 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 Red Hat Network 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作成できます。子チャンネルのクローン作成は以下の手順で行います。
  1. 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、クローンが作成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。

### 3.1.2.3. キックスターツリーのプロファイル

キックスターツリーのプロファイルは、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ますようになります。

### 手順3.3 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. 必要な **仮想化タイプ** を選択します。仮想タイプについて詳しい情報は「[キックスタートのプロファイル](#)」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3.4 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. 必要な **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、「[キックスタートのプロファイル](#)」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-x86_64-server-6-6.4
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. **Red Hat 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://$http_server/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。

Overview Systems **Errata** Channels Configuration Schedule Users Admin Help

1 SYSTEM SELECTED **MANAGE** CLEAR

Overview  
Systems  
System Groups  
System Set Manager  
Advanced Search  
Activation Keys  
Stored Profiles  
Custom System Info  
**Kickstart**  
Profiles  
Bare Metal  
GPG and SSL Keys  
Distributions  
File Preservation  
Kickstart Snippets

## Create Kickstart Profile

### Kickstart Details

Each kickstart file you upload to satellite will need a label so that you can refer to it later - please choose a label for this kickstart and enter it below. Entries marked with an asterisk (\*) are required.

**Label\*:**

**Kickstartable Tree\*:**

**Virtualization Type:**   
**NOTE:** Changing the Virtualization Type may require changes to the kickstart profile's bootloader and partition options, potentially overwriting user customizations. Please visit the [Partitioning](#) tab to verify the new settings.

**File Contents:**

1

Position: Ln 1, Ch 1 Total: Ln 1, Ch 0

☒ Toggle editor

**Tip:** Alternatively, you can upload a new version in the 'File to Upload' section below and select 'Upload File'.

**Tip:** If you wish to re-use an existing Kickstart Profile we recommend you copy-paste the template from `/var/lib/rhn/kickstarts` instead of from your browser. This will ensure that the template variables will be preserved and the registration process will work properly after the kickstart.

**Create**

**File to Upload**  **Browse...**

**Upload File**

32.06 KB/s 0.999 Done

図3.1 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3.2 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 または KVM 以外 (例 : VMware、Virtage など) の通常のプロビジョニング、ペアメタルインストール、および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。

タイプ	説明	用途
<b>Xen 完全仮想化ゲスト</b>	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> <b>注記</b> <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>
<b>Xen 準仮想化ゲスト</b>	Xen ゲスト	<p>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 のみが、準仮想化でのゲストをサポートしています。</p>
<b>Xen 仮想化ホスト</b>	Xen ホスト	<p>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。これは、Red Hat Enterprise Linux 5 のみでサポートされます。</p>

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ります。



#### 注記

キックスターに関する詳しい情報は、『Red Hat Enterprise Linux インストールガイド』の『キックスターのインストール』のセクションを参照してください。

### 3.1.2.4. テンプレーティング

キックスターの テンプレーティングにより、ご使用のキックスターファイル内に変数、スニペット、および **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、および 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 1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、1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また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を定義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、また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、Red Hat Satellite Server 上に書き込まれたか、または Red Hat 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、Red Hat 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。キックスタートプロファイルが作成された後に、スニペットがプロファイルにどのように組み込まれているかを確認するために **/var/lib/rhn/kickstarts/** の内容をレビューします。

**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 Red Hat 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、プロファイルまた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。Red Hat 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すでに入っています。

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し、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/` ディレクトリーに格納されます。Red Hat Satellite は、スニペットを組織別に異なるディレクトリーに格納するため、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、スニペット一覧また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仕方を変更することができます。

The screenshot shows the Red Hat Network Satellite web interface. The top navigation bar includes links for English (change), Knowledgebase, Documentation, and user/organization information. The main navigation menu has tabs for Overview, Systems, Errata, Channels, Configuration, Schedule, Users, Monitoring, and Help. The left sidebar contains a list of navigation options, with 'Kickstart Snippets' selected. The main content area is titled 'Kickstart Snippets' and includes a 'create new snippet' button. Below this, there are tabs for 'Default Snippets', 'Custom Snippets', and 'All Snippets'. A description explains that kickstart snippets are used to store common blocks of code. A filter section allows searching by snippet name and setting the number of items per page (currently 25). The main table lists snippets with columns for 'Snippet Name' and 'Snippet Macro'. One snippet, 'MyCustomSnippet', is listed with the macro '\$SNIPPET('spacewalk/28/MyCustomSnippet')'. A tip at the bottom advises copying and pasting the snippet macro into kickstart profiles.

図3.2 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、これらをエスケープして、変数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3.1 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。

\$ をエスケープする最も簡単な方法は、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。

2つ目の方法は、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、キックスタートテンプレートの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トにすでに含まれていますが、手動で作成する **raw** キックスタートにも組み入れが可能です。

#### 3.1.2.5. ベアメタルからのキックスタート

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
- PowerPC の Yaboot

### 手順3.5 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トと適合している必要があります。例えば、キックスタートが **ks-rhel-x86\_64-server-6-6.4** キックスタートツリーを使用するように構成されている場合は、Red Hat Enterprise Linux 6.4 64 ビットの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ト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、キックスタート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3.6 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。または、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 の設定ファイルの編集には、DHCP サーバーへの管理アクセスが必要となります。

#### 前提条件

最新の **cobbler-loaders** パッケージをインストールする必要があります。これは、PXE ブート前に **pxelinux.0** ブートイメージが **Satellite** にインストールされており、利用可能であることを確認するためです。最新のバージョンをインストールするには、以下を実行します。

```
# yum install cobbler-loaders
```

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)、すべてのルーターまたはスイッチがネットワーク境界を越えて DHCP プロトコルを渡すように設定することで可能になります。

Red Hat 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 サーバーをポイントするように構成します。

インストール時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 アドレスをポイントするように設定します。

```
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転送します。
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。

DHCP サーバーを再起動します。

```
# service dhcpd restart
```

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** サービスを有効にする一方で、**/sbin/service** コマンドは、**xinetd** を即時に有効にします。

### 手順3.7 Yaboot の起動

**Yaboot** は、**PowerPC** の **Open Firmware** 層内で動作するブートシステムです。キックスタートの実行には、ご使用の環境と **PowerPC** クライアントに設定が必要です。

1. PowerPC クライアントの起動順序を設定します。これには、**Open Firmware** インターフェースにアクセスし、プロンプトで以下のコマンドを実行する必要があります。

**devalias** コマンドを使用して、システム上のすべてのデバイスのエイリアスを表示します。

```
0 > devalias
ibm,sp                /vdevice/IBM,sp@4000
disk
/pci@8000000020000002/pci@2,4/pci1069,b166@1/scsi@1/sd@5,0
network               /pci@8000000020000002/pci@2/ethernet@1
net                   /pci@8000000020000002/pci@2/ethernet@1
network1              /pci@8000000020000002/pci@2/ethernet@1,1
scsi
/pci@8000000020000002/pci@2,4/pci1069,b166@1/scsi@0
nvram                 /vdevice/nvram@4002
rtc                   /vdevice/rtc@4001
screen                /vdevice/vty@30000000
ok
```

**boot-device** 環境変数をチェックして、現在の起動順序を表示します。

```
0> printenv boot-device
----- Partition: common ----- Signature: 0x70 -----
-----
boot-device           /pci@8000000020000002/pci@2,3/ide@1/disk@0
/pci@8000000020000002/pci@2,4/pci1069,b166@1/scsi@1/sd@5,0
ok
```

最初に **network** デバイスで **boot-device** 環境変数を設定した後に既存のブートデバイスで設定し、**network** デバイスを起動順序の一番上に追加します。

```
0 > setenv boot-device network
/pci@8000000020000002/pci@2,3/ide@1/disk@0
/pci@8000000020000002/pci@2,4/pci1069,b166@1/scsi@1/sd@5,0
```

これで、最初に **network** デバイスから起動するようシステムが設定されました。**network** に障害が発生した場合、システムの残りのデバイスは起動順序どおりに起動します。

2. **Satellite** サーバーでシステム設定プロパティを設定します。たとえば以下のコマンドを実行すると、ネットワーク上の特定の **MAC** アドレスを使用して **myppc01** という新しいシステムが作成されます。

```
# cobbler system add --name myppc01 --hostname myppc01.example.com -
-profile rhel6webserver--kopts "console=hvc0 serial" --interface 0 -
-mac 40:95:40:42:F4:46
```

さらに、指定の **MAC** アドレスを基に **Yaboot** ブートローダーと設定のディレクトリーのセットも作成されます。たとえば、前述のコマンドを実行すると、以下のディレクトリーが作成されます。

```
/var/lib/tftpboot/ppc/40-95-40-42-F4-46
/var/lib/tftpboot/etc/40-95-40-42-F4-46
```

最初のディレクトリー (**/var/lib/tftpboot/ppc/40-95-40-42-F4-46**) には、**Yaboot** に

使用される **ramdisk** および **vmlinux** ファイルが含まれます。2 つ目のディレクトリー (**/var/lib/tftpboot/etc/40-95-40-42-F4-46**) には、Yaboot の設定ファイル (**yaboot.conf**) が含まれます。

Cobbler を使用したシステムのプロビジョニングに関する詳細は、「[Cobbler へのシステムの追加](#)」を参照してください。

3. 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。Yaboot のインストールを実行するためのオプションが含まれる新しいエントリーを追加します。例を以下に示します。

```
allow booting;
allow bootp;
class "Yaboot" {
    match if substring(option vendor-class-identifier, 0, 9) =
"AAPLBSDPC";
    next-server 192.168.2.1;
    filename "yaboot";
}
```

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. **Yaboot** というクラスを作成します。起動順序で Yaboot が最優先になるようシステムが設定されている場合、**fAAPLBSDPC** として識別されます。
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転送します。
4. DHCP サーバーは、以前 **cobbler** で作成された Yaboot イメージファイルを参照します。

DHCP サーバーを再起動します。

```
# service dhcpd restart
```

4. Xinetd を設定します。Xinetd はサービスを管理するデーモンです。管理されるサービスには、ブートイメージを PowerPC クライアントに転送するために使用される 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。

5. Xinetd サービスを起動し、TFTP が Yaboot ブートイメージの処理を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** サービスを有効にする一方で、**/sbin/service** コマンドは、**xinetd** を即時に有効にします。

### 3.1.3. アクティベーションキーの使用

アクティベーションキーの管理者(Satellite 管理者を含む)のロールを持つ Red Hat Network Management および Provisioning のお客様は、Red Hat Satellite Web ページからアクティベーションキーを生成することができます。これらのキーを使用してコマンドラインユーティリティである `rhnmreg_ks` で Red Hat Enterprise Linux システムを登録し、このシステムに Red Hat Satellite サービスレベルのエントリーメントを付与し、特定のチャンネルやシステムグループにシステムをサブスクライブさせることができます。



#### 注記

システムの詳細 ページの再アクティベーション サブタブから作成したシステム固有のアクティベーションキーはシステム共通で再利用できないためこの一覧には含まれません。

#### 手順3.8 アクティベーションキーを管理する

アクティベーションキーを生成するには:

1. 左上のナビゲーションバーから システム → アクティベーションキー の順に選択します。
2. 右上の **新規のキーを作成** リンクをクリックします。



#### 警告

以下に表示されているフィールドのほかに、キー フィールド自体も入力することができます。このユーザー定義の文字列を `rhnmreg_ks` で指定すると、Satellite でクライアントシステムを登録できます。キーにはコンマを挿入しないでください。これ以外の文字はすべて使用できます。コンマは複数のアクティベーションキーを1度に指定するときに区切り文字として使用されるため、問題になります。詳しくは「[アクティベーションキーの使用](#)」を参照してください。

3. 次の情報を指定します。

- **詳細** - 生成されたアクティベーションキーを識別し易くするためにユーザーが定義する説明になります。
- **使用** - 一度にアクティベーションキーで登録できる登録システムの最大数です。使用を無制限にする場合は空白にします。システムプロファイルを1つ削除すると使用数が1つ減り、キーでシステムプロファイルを1つ登録すると、使用数が1つ増えます。
- **ベースチャンネル** - キーの主要チャンネルになります。ベースチャンネルを選択しない場合はすべての子チャンネルを選択できるようになりますが、適用可能なチャンネルにしかシステムをサブスクライブさせることができません。
- **付属エントリーメント** - キーの補足的なエントリーメントになります。Monitoring、Provisioning、Virtualization、および Virtualization Platform などが含まれます。このキーを使って、すべてのシステムにこれらのエントリーメントが付与されます。

- **Universal default** - このキーを組織のプライマリーアクティベーションキーとして考慮するかどうか。指定のアクティベーションキーを使用せずに **Satellite** に登録されるクライアントには **Universal Default** アクティベーションキーが使用されます。**Universal Default** アクティベーションキーには基本のシステム登録の標準または最低限のチャンネルとエンタイトルメントが含まれることが理想です。**Universal Default** アクティベーションキーの詳細は <https://access.redhat.com/solutions/1140083> を参照してください。

**アクティベーションキーの作成** をクリックします。

固有のキーを作成すると、それはアクティベーションキーの一覧にキーが使用された回数と共に表示されます。この一覧はアクティベーションキー管理者のみが閲覧できます。この時点で、子チャンネルとグループをキーと関連付けて、キーで登録したシステムが自動的にサブスクライブされるようにすることができます。

チャンネルまたはグループなどのキーについての情報を変更するには、キーの一覧の説明部分をクリックし、該当タブ内で修正を行ってから、**アクティベーションキーの更新** ボタンをクリックします。キーからチャンネルとグループの関連付けを解除するには、**Ctrl** キーを押しながら強調表示された名前をクリックしてそれぞれのメニューの選択を解除していきます。キーを完全に削除するには、編集ページの右上にある **キーの削除** リンクをクリックします。

アクティベーションキーで登録する際、システムがベースチャンネルにサブスクライブするように設定することができます。ただし、アクティベーションキーがシステムのオペレーティングシステムと互換性がないベースチャンネルを指定していると登録は失敗します。例えば、**Red Hat Enterprise Linux 6 for x86\_64** システムは、**Red Hat Enterprise Linux 5 for x86\_64** ベースチャンネルを指定しているアクティベーションキーでは登録できません。システムのカスタムベースチャンネルへのサブスクライブは常に許可されます。

キーを使ってシステムのアクティベーションを無効にするには、キーの一覧内の **有効** 列の下にある該当チェックボックスの選択を解除します。キーは、このチェックボックスを選択すると再度有効にできます。変更後は、このページの右下にある **アクティベーションキーの更新** ボタンをクリックします。

### 手順3.9 複数のアクティベーションキーを一度に使用する

**Provisioning** エンタイトルメントでは、複数のアクティベーションキーをコマンドラインまたは単独のキックスタートプロファイル内に組み込むことができます。これにより、必要なシステムに固有のキーを新たに再作成することなくそれぞれのキーの特徴を集約でき、登録およびキックスタートのプロセスを単純化できると共にキーの一覧が増えすぎないようにすることができます。

複数のアクティベーションキーで登録する際には注意が必要です。一部の値が競合すると登録が失敗する原因になります。ただし、ソフトウェアパッケージ、ソフトウェアの子チャンネル、および設定チャンネルの値が競合しても、登録が失敗する原因にはなりません。残りのプロパティ間の競合は次のような方法で解決されます。

- ベースソフトウェアチャンネル - 登録が失敗します。
  - エンタイトルメント - 登録が失敗します。
  - 設定フラグを有効にする (**enable config flag**) - 設定管理が設定されます。
1. 個々のアクティベーションキーを複数作成します。手順は「**アクティベーションキーの使用**」を参照してください。
  2. コマンドライン上では、**rhnreg\_ks** のオプションの場合のように、すべてのアクティベーションキーをコンマで区切って組み込みます。



```
# rhnreg_ks --  
activationkey=activationkey1,activationkey2,activationkey3
```

キックスタートの詳細 ページの **ポスト (Post)** タブ内に追加することも可能です。手順は「[キックスタートのプロファイル](#)」を参照してください。



#### 警告

システム固有のアクティベーションキーを他のアクティベーションキーと併用しないでください。これを行うと登録に失敗します。

### 3.1.4. Cobbler の使用

Red Hat Satellite には管理者がシステムのインストールとインフラストラクチャーの提供を集中管理できるようにする **Cobbler** サーバーが搭載されています。**Cobbler** は、無人のシステムインストールを行うさまざまな方法を提供するインストールサーバーであり、サーバーやワークステーションのインストール、完全仮想化または準仮想化の設定によるゲストシステムなどのインストールを行います。

**Cobbler** にはインストール前の準備、キックスタートファイルの管理、インストール環境の管理などに役立つツールがいくつか備えられています。



#### 警告

本セクションでは、**Cobbler** のサポートされている使用法について説明します。これには以下の機能が含まれます。

- **Cheetah** のテンプレートエンジンとキックスタートスニペットを使用したキックスタートのテンプレート作成および管理の機能
- クライアント側のツールである **koan** を使った仮想マシンゲストのインストールを自動化する機能
- **cobbler check** コマンドを使用したインストール環境の分析の機能
- **x86\_64** アーキテクチャーの **Satellite** システムの **cobbler buildiso** コマンドによる **PXE** のようなメニューを使用してインストール ISO を構築する機能

Red Hat は、正式なドキュメンテーションに記載されているか、**Satellite Web UI** および **API** または **Red Hat カスタマーポータル** の **Supported Cobbler Usage** という件名の下で記載されている記事で公開されている **Cobbler** 機能のみをサポートしています。

#### 3.1.4.1. Cobbler の要件

Cobbler を PXE ブートサーバーとして使用する場合は、以下のガイドラインを確認してください。

- PXE でのシステムインストールを行う際に Cobbler を使用する場合は、**tftp-server** パッケージをインストールし、これを設定します。
- インストールを行うために Cobbler を使用してシステムを PXE ブートを行うには、**Satellite** サーバーが Cobbler による PXE ブートを行うために DHCP サーバーとして動作するか、またはネットワーク DHCP サーバーにアクセスする必要があります。**/etc/dhcp.conf** を編集して **next-server** を Cobbler サーバーのホスト名または IP アドレスに変更します。
- 最新の **cobbler-loaders** パッケージをインストールする必要があります。これは、PXE ブートの前に **pxelinux.0** ブートイメージが **Satellite** にインストールされており、利用可能であることを確認するためです。最新バージョンをインストールするには、以下を実行します。

```
# yum install cobbler-loaders
```

#### 3.1.4.1.1. Cobbler と DHCP の設定

Cobbler は、PXE ブートサーバーを使ったネットワークによる起動を行うように設定されたベアメタルのキックスタートインストールに対応しています。Cobbler インストールサーバーを正しく実装するには、管理者はネットワークの DHCP サーバーにアクセス権を持っているか、または Cobbler サーバー自体に DHCP を実装する必要があります。

#### 手順3.10 既存の DHCP サーバーを設定する

ネットワーク上の別のシステムで DHCP サーバーをデプロイしている場合、DHCP 設定ファイルが Cobbler サーバーと PXE ブートイメージをポイントするよう変更を行うために DHCP サーバーへの管理アクセスが必要になります。

1. DHCP サーバーに **root** としてログインします。
2. **/etc/dhcpd.conf** ファイルを編集し、PXE ブートインストールを実行するためのオプションを指定して新規のクラスを追加します。例えば、以下のようになります。

```
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";
}
```

上記の例の各動作を以下に示します。

1. 管理者は **bootp** プロトコルを使用してネットワークによる起動を有効にします。
2. 次に、**PXE** というクラスを作成します。PXE が起動順序で一番目に設定されているシステムの場合は、それ自体を **PXEClient** として認識します。
3. DHCP サーバーはシステムを **192.168.2.1** にある Cobbler サーバーに転送します。
4. 最後に DHCP サーバーは **pxelinux.0** ブートローダーファイルを取り込みます。

### 3.1.4.1.2. Cobbler 用の Xinetd と TFTP の設定

Xinetd は、TFTP などのサービス群を管理するデーモンです。TFTP とは起動イメージを PXE クライアントに転送する際に使用される FTP サーバーです。TFTP を設定するには以下を実行します。

1. root としてログインします。
2. `/etc/xinetd.d/tftp` を編集して、以下のようにオプションを変更します。

```
disable = yes
```

変更後

```
disable = no
```

3. xinetd サービスを起動するには:

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

### 3.1.4.1.3. Cobbler サポート用の SELinux と IPTables の設定

Red Hat Enterprise Linux はデフォルトで SELinux にも対応できるようにインストールが行われており、セキュアなファイアウォールが有効になっています。Red Hat Enterprise Linux サーバーが Cobbler を使用するように正しく設定するには、まず Cobbler サーバーとの接続を許可するようシステムとネットワークセーフガードを設定する必要があります。

#### 手順3.11 SELinux で Cobbler サポートを有効にする

Cobbler のサポート用に SELinux を有効にするには:

1. root としてログインします。
2. SELinux の Boolean 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には:

```
# 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。

#### 手順3.12 IPTable の設定

SELinux を設定したら、IPTables を設定して Cobbler サーバー上での送受信のネットワークトラフィックを許可する必要があります。

1. root としてログインします。
2. 以下のルールを既存の IPTables ファイアウォールのルールセットに追加して、Cobbler 関連のポートを開きます。
  - TFTP 用:

```
# /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 用:

```
# /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 および Koan XMLRPC 用:

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

### 3. ファイアウォールの設定を保存するには:

```
# /sbin/iptables-save > /etc/sysconfig/iptables
```

#### 3.1.4.2. /etc/cobbler/settings での Cobbler の設定

Cobbler の設定は **/etc/cobbler/settings** ファイル内で実行されます。このファイルには数種の設定可能な設定が含まれ、Cobbler の機能に対して各設定がどのように影響を与えるか、またユーザーの環境に応じて設定の変更が推奨されるかどうかなど、各設定ごとに詳細な説明が記載されています。

ほとんどの設定はデフォルトのままで構いません。Cobbler は設計通りに実行されます。Cobbler の設定方法の詳細については、各設定が詳細に記載されている **/etc/cobbler/settings** ファイルをご覧ください。

#### 3.1.4.3. Cobbler サービスの同期と起動

1. Cobbler サービスを起動する前に、Cobbler サービスでチェックを実行し、すべての要件が組織のニーズに応じて設定されていることを確認します。

```
# cobbler check
```

2. 次のコマンドで Satellite サーバーを起動するには:

```
# /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。

Red Hat Satellite の起動/停止には、**/usr/sbin/rhn-satellite** を必ず使用してください。

#### 3.1.4.4. Cobbler へのディストリビューションの追加

すべての要件を満たされていて Cobbler が実行中の場合は、ディストリビューションを Cobbler に追加できるようになります。

**cobbler** を使用して、次の構文でディストリビューションを作成します。

```
# cobbler distro add --name=string --kernel=path --initrd=path
```

**--name=string** スイッチは1つのディストリビューションを他と区別するために使用するラベルです (例: **rhel5server**)。

**--kernel=path** スイッチは、カーネルイメージファイルへのパスを指定します。

**--initrd=path** スイッチは、初期の ramdisk (initrd) イメージファイルへのパスを指定します。

#### 3.1.4.5. Cobbler へのプロファイルの追加

Cobbler にディストリビューションを追加したら、今度はプロファイルを Cobbler に追加することができます。

Cobbler プロファイルはキックスターートのファイルのように追加オプションに対して任意のディストリビューションの関連付けを行います。プロファイルはプロビジョニングの核となるユニットであり、追加されるディストリビューションにはそれぞれ少なくとも 1 つの Cobbler プロファイルがなければなりません。例えば、Web サーバーとデスクトップの設定用に 2 種類のプロファイルが作成される場合があります。いずれのプロファイルも同じディストリビューションを使用しますが、インストールタイプは異なります。

キックスターートプロファイルを Red Hat Satellite インターフェースから作成し、設定する方法について詳しくは、「[キックスターートのプロファイル](#)」を参照してください。

**cobbler** を使用して、次の構文でプロファイルを作成します。

```
# cobbler profile add --name=string --distro=string [--kickstart=url] [--virt-file-size=gigabytes] [--virt-ram=megabytes]
```

**--name=string** は、**rhel5webserver** や **rhel4workstation** などプロファイルの固有のラベルとなります。

**--distro=string** スイッチは、この特定のプロファイルに使用されるディストリビューションを指定します。ディストリビューションは、「[Cobbler へのディストリビューションの追加](#)」で追加されました。

**--kickstart=url** オプションは、キックスタータファイルがある場合はその場所を指定します。

**--virt-file-size=gigabytes** オプションを使用すると仮想ゲストファイルイメージのサイズを設定することができます。指定しないとデフォルトの 5 ギガバイトになります。

**--virt-ram=megabytes** オプションは、仮想ゲストシステムが消費できる物理 RAM の量をメガバイトで指定します。指定がない場合はデフォルトの 512 メガバイトになります。

#### 3.1.4.6. Cobbler へのシステムの追加

Cobbler 用のディストリビューションとプロファイルを作成した後に、システムを Cobbler に追加します。システムの記録や、システムで実行するよう割り当てられた **cobbler** プロファイルに基づいて、クライアントにあるそれぞれのハードウェアのマッピングを行います。



#### 注記

**koan** と PXE メニューのみでプロビジョニングを行なっている場合、システム記録を作成する必要はありません。ただし、以下の場合にシステム記録は役に立ちます。

- システム固有のキックスターのテンプレーティングが必要である。
- 特定のシステムが常に特定のコンテンツのセットを受信する。
- 特定のクライアント用に特定のロールがある。

以下のコマンドでシステムを Cobbler 設定に追加できます。

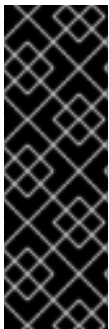
```
# cobbler system add --name=string --profile=string --mac-address=AA:BB:CC:DD:EE:FF
```

**--name=string** は **engineeringserver** や **frontofficeworkstation** などのシステムに固有となるラベルです。

**--profile=string** は、「[Cobbler へのプロファイルの追加](#)」で追加されたプロファイル名の 1 つを指定します。

**--mac-address=AA:BB:CC:DD:EE:FF** オプションは、指定 MAC アドレスを持つシステムがキックスタータされると、そのシステム記録に関連するプロファイルに自動的にプロビジョニングが行われます。

ホスト名や IP アドレスの設定などオプションの詳細については、シェルプロンプトで **man cobbler** と入力して Cobbler man ページを参照してください。



## 重要

**default** という名前を持つシステムには特別な機能があります。未定義のシステムをすべて設定し、PXE 経由で特定のプロファイルを使用します。**default** システムがないと、PXE は未設定のシステムに対してローカルブートになります。以下のように、名前とプロファイルとしてのみ **default** を含めます。

```
# cobbler system add --name=default --profile=rhel5webserver
```

### 3.1.4.7. Cobbler テンプレートの使用

Red Hat Satellite web インターフェース内で、キックスタートディストリビューションとプロファイルで使用する変数を作成するための各種の機能を使用できます。

キックスタートファイルでの テンプレート作成に対応するために **Satellite** 内のインフラストラクチャーに加えられた変更の一部として、キックスタートの変数があります。キックスタートファイルのコンテキストでは、テンプレートとは特定のキックスタートを作成するのではなく実際のキックスタートファイルをビルドするため使用される詳細内容を保持するファイルのことです。

これらのテンプレートは、独自の変数や対応する値を持つ各種のプロファイルとシステムで共有されます。これらの変数がテンプレートを修正し、テンプレートエンジンと呼ばれるソフトウェアがそのテンプレートと変数データを使用に適したキックスタートファイルに構文解析します。**Cobbler** は **Cheetah** と呼ばれる高度なテンプレートエンジンを使用し、テンプレート、変数、およびスニペットのサポートを提供します。

テンプレートを使用する利点は以下ようになります。

- 各状況に合わせて個別のキックスタートを手作業で作成したり重複した作業を行ったりすることなく、大量のプロファイルやシステムの作成と管理を行えるようにする強固な機能
- テンプレートは複雑性を増してループや条件、他の拡張機能、および構文などを伴う場合がある一方、こうした複雑性が伴うことなく単純にキックスタートファイルの作成にテンプレートを使用することもできます。

#### 3.1.4.7.1. テンプレートの使用

キックスタートテンプレートには、PXE イメージのファイル名、サブネットアドレス、**/etc/sysconfig/network-scripts/** などの共通パスなど特定の共通項目に対して静的な値を持たせることができます。ただし、テンプレートの標準のキックスタートファイルとの相違点はその変数の使用に見られます。

例えば、標準のキックスタートファイルには以下のようなネットワーク経路がある場合があります。

```
network --device=eth0 --bootproto=static --ip=192.168.100.24 --
netmask=255.255.255.0 --gateway=192.168.100.1 --nameserver=192.168.100.2
```

しかし、キックスタートテンプレートファイルでは、ネットワーク経路は以下になる可能性があります。

```
network --device=$net_dev --bootproto=static --ip=$ip_addr --
netmask=255.255.255.0 --gateway=$my_gateway --nameserver=$my_nameserver
```



これらの変数はキックスタートプロファイルの変数またはシステム詳細の変数で設定される値で置き換えられます。プロファイルとシステム詳細の両方で同じ変数が定義されている場合は、システムの変数が優先されます。

#### 3.1.4.7.2. キックスタートスニペット

キックスタートの全テンプレートとプロファイルで同一の共通の設定がある場合は、**Cobbler** の **スニペット** 機能を使ってコードの再使用という利点を活用することができます。

キックスタートのスニペットとはキックスタートコードのセクションであり、**Cobbler** によって構文解析される **\$SNIPPET()** 関数で呼び出し、その関数呼び出しをコードスニペットの内容に置き換えることができます。

例えば、次のように全サーバーに対して共通となるハードドライブのパーティション設定があるとし

```
clearpart --all
part /boot --fstype ext3 --size=150 --asprimary
part / --fstype ext3 --size=40000 --asprimary
part swap --recommended

part pv.00 --size=1 --grow

volgroup vg00 pv.00
logvol /var --name=var vgroup=vg00 --fstype ext3 --size=5000
```

このスニペットを利用して **my\_partition** などのファイルに保存し、**Cobbler** がアクセスできるようにこのファイルを **/var/lib/cobbler/snippets/** に配置します。

次に、キックスタートテンプレート内の **\$SNIPPET()** 関数でこの部分を利用することができます。例えば、以下ようになります。

```
$SNIPPET('my_partition')
```

この関数を呼び出した場合は常に、**Cheetah** 構文解析ツールによってこの関数が **my\_partition** ファイル内に含まれているコードのスニペットに置き換えられます。

#### 3.1.4.8. Koan の使用

仮想マシン上でゲストのプロビジョニングを行っているか、または実行中のシステムに新規のディストリビューションを再インストールしている場合、**koan** は **Cobbler** と連携して複数のシステムのプロビジョニングを行います。

##### 3.1.4.8.1. Koan の使用による仮想システム群のプロビジョニング

「[Cobbler へのプロファイルの追加](#)」の説明どおりに仮想マシンのプロファイルを作成した場合、**koan** を使用して、システム上での仮想ゲストのインストールを開始できます。

例えば、以下のような **Cobbler** プロファイルを作成したとします。

```
# cobbler add profile --name=virtualfileservr --distro=rhel-i386-server-5
--virt-file-size=20 --virt-ram=1000
```

このプロファイルは、20GB のゲストイメージサイズでシステム RAM には 1 GB が割り当てられた Red Hat Enterprise Linux 5 を実行しているファイルサーバー用となります。

仮想ゲストのシステムプロファイル名を検索する場合は **koan** で以下を実行します。

```
# koan --server=hostname --list=profiles
```

このコマンドは **cobbler profile add** で作成された利用可能なプロファイルをすべて一覧表示します。

次に、イメージファイルを作成して仮想ゲストシステムのインストールを開始するプロセスを開始します。

```
# koan --virt --server=cobbler-server.example.com --  
profile=virtualfileservier --virtname=marketingfileservier
```

このコマンドは、仮想ゲストシステムが **virtualfileservier** プロファイルを使用して Cobbler サーバー (ホスト名 **cobbler-server.example.com**) から作成されることを指定しています。 **virtname** オプションは仮想ゲストのラベルを指定しています。デフォルトではシステムの MAC アドレスでラベル付けされます。

仮想ゲストのインストールが完了すると、他の仮想ゲストシステムと同じように使用できます。

#### 3.1.4.8.2. Koan の使用による実行中システムの再インストール

マシンの実行中に別のオペレーティングシステムでそのマシンを再インストールする必要がある場合があります。 **koan** は利用可能な Cobbler プロファイルから新規インストールで実行中のシステムを置き換えることができます。

実行中のシステムを置き換えて新規のインストールを行う場合には、**そのシステム自体**で以下のコマンドを実行します。

```
# koan --replace-self --server=hostname --profile=name
```

置き換えを行う実行中のシステムでこのコマンドを実行すると、プロビジョニングのプロセスが開始され、 **--server=hostname** 内で指定した Cobbler サーバーの **--profile=name** にあるプロファイルを使ってそれ自体のシステムの置き換えを行います。

#### 3.1.4.9. Cobbler による ISO の構築

環境によっては、PXE サポートがない場合があります。この場合は、**cobbler buildiso** コマンドが、ディストリビューションとカーネルの一式、および PXE ネットワークインストールのようなメニューを含むブート ISO イメージを作成する機能を提供します。

**--iso** オプションを使用してブート ISO の名前と出力の場所を定義します。

```
# cobbler buildiso --iso=/path/to/boot.iso
```

デフォルトでは、ブート ISO にすべてのプロファイルとシステムが含まれます。 **--profiles** と **--systems** オプションを使用して、これらのプロファイルとシステムを制限します。

```
# cobbler buildiso --systems="system1,system2,system3" \  
--profiles="profile1,profile2,profile3"
```



## 注記

`cobbler buildiso` コマンドによる ISO の構築は、**s390x** アーキテクチャー以外のすべてのアーキテクチャーでサポートされます。



## 注記

`cobbler buildiso --standalone` オプションは Red Hat が提供するキックスタートツリーではサポートされません。この `standalone` オプションは、Cobbler サーバーへのネットワーク接続のないマシンのプロビジョニングに使用されますが、Satellite のプロビジョニングによって提供される追加機能にはすべて Satellite へのネットワーク接続が必要です。Red Hat Enterprise Linux をネットワーク接続のないマシンにインストールする必要がある場合は、ISO イメージをダウンロードして Red Hat Enterprise Linux をインストールしてください。

## 3.2. RED HAT SATELLITE PROXY を介したプロビジョニング

プロビジョニングは、すでにインストールされ、Red Hat Satellite に登録済みの Red Hat Satellite 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、またはシステムの再プロビジョニングを行う場合には、**Satellite Proxy を選択 (Select Satellite Proxy)** のドロップダウンメニューから必要な 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、Red Hat 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 が以下のようになります。

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

3. Red Hat Satellite 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
```

## 3.3. 仮想化されたゲストのプロビジョニング

仮想ゲストのプロビジョニングは、以下の仮想技術を使用する Red Hat Satellite 5 でサポートされます。

- 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3.13 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ー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、ディレクトリー、またはブロックデバイスへのパス。これには、**/dev/sdb**、**/dev/LogVol100/mydisk**、**VolGroup00**、または **/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。

### 3.4. クローンされたチャンネルまたはカスタムチャンネル経由のキックスタート

クローンされたチャンネルまたはカスタムチャンネルからシステムをインストールする方法は 2 つあります。

1. **%packages** の下に **@Base** がある **Red Hat Enterprise Linux** チャンネルからシステムをキックスタートします。同じキックスタートで、クローンされたチャンネルまたはカスタムチャンネルへサブスクライブするアクティベーションキーを指定します。また、クローンされたチャンネルまたはカスタムチャンネルからインストールするパッケージを、アクティベーションキーのパッケージセクションに示します。この方法でシステムをキックスタートすると、最初に **Red Hat Enterprise Linux Base** チャンネルから最低限のパッケージでインストールされ、アクティベーションキーを使用して **Satellite 5** サーバーのクローンされたチャンネルへシステムを登録します。登録後、キックスタートよりシステムを更新します。
2. この他に、クローンされたチャンネルまたはカスタムチャンネルから直接システムをキックスタートしたい場合は、チャンネルのディストリビューションツリーを作成することもできます。手順の例を以下に示します。

#### 手順3.14 クローンされたチャンネルのディストリビューションツリーの作成

1. クローンされたチャンネルをコピーします。Red hat Enterprise Linux 6.6 の場合は次のようになります。

```
# cd /var/satellite/rhn/kickstart/
# mkdir custom-distro-rhel-6.6
# cd custom-distro-rhel-6.6
# cp -rpv ../ks-rhel-x86_64-server-6-6.6/* .
```

2. ツリーのファイルすべてのパーミッションを **644** に設定し、所有者を **apache:apache** にします。

```
# find . -type f -print0 | xargs -0 chmod 644
# find . -type f -print0 | xargs -0 chown apache:apache
```

ツリーのディレクトリーすべてのパーミッションを **755** に設定し、所有者を **apache:root** にします。

```
# find . -type d -print0 | xargs -0 chmod 755
# find . -type d -print0 | xargs -0 chown apache:root
```

3. Satellite 5 サーバー web UI にログインし、**Systems** → **Kickstart** → **Distribution** → **create new distribution** と選択します。

4. 以下の値を設定します。

- **Distribution Label:** custom-distro-rhel-6.6
- **Tree Path:** /var/satellite/rhn/kickstart/custom-distro-rhel-6.6/
- **Installer Generation:** 6

終了したら、**Create Kickstart Distribution** をクリックします。

クローンされたチャンネルとキックスターツツリーに設定されたベースチャンネルでキックスターツプロファイルが作成されます。



### 注記

Red Hat Enterprise Linux 6 システムでは、クローンされたチャンネルのキックスターツされたシステムの登録およびサブスクリाइブに、クローンされたチャンネルをベースチャンネルとして使用するアクティベーションキーが必要となります。このアクティベーションキーをクローンされたチャンネルのキックスターツに追加します。新たにキックスターツされたシステムは、同じクローンされたチャンネル (キックスターツが実行された) に登録およびサブスクリाइブされます。

## 3.5. 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 **Red Hat 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、エンタイトルメントやシステムグループ、およびサブスクリाइブされているチャンネルなど、システムの情報や設定が維持されることになります。

システム内に現在インストールされているバージョンとは異なる主要な Red Hat Enterprise Linux バージョン

ジョンを使ってシステムが再プロビジョニングされる場合、システムプロファイルやエンタイトルメント、およびサーバーグループが維持されます。ただし、ベースチャンネルは変更されたため、すべての子チャンネルは失われます。

再プロビジョニング用に選択されたキックスタートプロファイルにアクティベーションキーが含まれる場合、再プロビジョニングによって、システムが持つ既存の関係を削除せずにアクティベーションキー内のすべての手順が実行されます。システムは、そのシステムプロファイル、チャンネルサブスクリプション、組織やグループの関連付けを維持します。システムは、単にアクティベーションキーに含まれる追加の組織、グループまたは追加のパッケージに追加されます。

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3.2 カーネルオプションとカーネルの後のオプションの設定

この例では、再プロビジョニングの設定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、**カーネルオプション** 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、**カーネルの後のオプション** の行に **noapic** を追加します。

### 手順3.15 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時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で復元します。



#### 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時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 → **プロファイル** 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。

## 3.6. スナップショットでのロールバックのプロビジョニング

システムでアクションが実行されるとスナップショットが作成されます。スナップショットは、グループ、チャンネル、パッケージ、および設定ファイルを特定します。**Provisioning** エンタイトルメントを持つ **Satellite** 管理者は、スナップショットロールバックでシステムのパッケージプロファイル、ローカル設定ファイル、および **RHN** 設定をロールバックできます。



## 注記

スナップショットロールバックはシステムの変更の一部を元に戻す機能をサポートしますが、すべての場合で適用されるわけではありません。たとえば、RPM パッケージのセットをロールバックできますが、複数の更新レベルにまたがるロールバックはサポートされません。

スナップショット サブタブはシステムの詳細→プロビジョニング→スナップショット にあります。スナップショット サブタブには、以下を含むシステムのすべてのスナップショットが表示されます。

- スナップショットが作成された理由
- 作成時間
- 各スナップショットに適用されたタグの数

### 手順3.16 スナップショットロールバックの実行

以下の手順にしたがって以前の設定に戻します。

1. システム をクリックします。
2. ロールバックする必要があるシステムをクリックします。
3. プロビジョニング→スナップショット タブを選択します。
4. 作成されたスナップショットの **理由** をクリックし、**Rollback** サブタブから順にサブタブ上で変更を確認します。
5. 各サブタブでロールバック中に加えられる変更を確認します。
  - グループメンバーシップ
  - チャンネルのサブスクリプション
  - インストールされたパッケージ
  - 設定チャンネルサブスクリプション
  - 設定ファイル
  - スナップショットタグ
6. 変更の内容が適切であれば **ロールバック** サブタブに戻り、**スナップショットにロールバックする** ボタンをクリックします。リストを再度表示するには、「スナップショットの一覧に戻る」をクリックします。

#### 3.6.1. スナップショットタグの使用

スナップショットタグは、意味のある説明をシステムのスナップショットに追加する方法です。スナップショットタグを使用すると、既知の作業設定、正常なアップグレード、またはその他の重要な出来事を記すことができます。

### 手順3.17 スナップショットタグの作成

スナップショット タブで以下を行います。

1. システムタグの作成 をクリックします。
2. タグ名 フィールドに説明を入力します。
3. 現在のスナップショットにタグ付け をクリックします。

タグを使用するには、スナップショットのタグリストでタグ名をクリックします。



## 第4章 システム管理

Red Hat Satellite では、システムレベルのサポートと Red Hat システムおよびシステムのネットワークの管理機能を提供しています。本章では、各種システムと、効果的な管理を行うために組織内の機能グループにこれらのシステムをどのように編成できるかについて説明します。

### 4.1. SATELLITE へのシステムの登録

システムは、Red Hat Satellite からパッケージ更新を要求するクライアントマシンです。これらのシステムは、Satellite からの更新を登録し、受信できるように設定された物理マシンまたは仮想化システムです。システムを Satellite に登録するのは、重要なステップです。クライアントシステムは、デフォルトでは組織の Satellite ではなく、Red Hat Network に登録されるためです。登録方法について詳しくは、『Red Hat Satellite クライアント設定ガイド』のクライアントの Satellite サーバーへの登録に関連する章を参照してください。

#### 4.1.1. Red Hat Network Bootstrap を使用したシステムの登録

Red Hat Network は、手作業によるシステムの再設定の多くを自動化できる Red Hat Network Bootstrap というツールを提供しています。Red Hat Network Bootstrap は Red Hat Satellite Server インストールプログラム で欠かせない重要な役割を担い、インストール時のブートストラップスクリプトの生成を可能にします。

Red Hat Satellite Proxy Server の管理者と更新された Satellite 設定をお使いの管理者には、単独で使えるブートストラップツールが必要になります。これはコマンド `/usr/bin/rhn-bootstrap` で呼び出される Red Hat Network Bootstrap により行われるため、Red Hat Satellite Server と Red Hat Satellite Proxy Server の両方にデフォルトでインストールされています。

正しく使用すれば、このツールが生成するスクリプトはどのクライアントシステムからも実行でき次のようなタスクを処理します。

- クライアントのアプリケーションを Red Hat Satellite Proxy または Satellite に転送します
- カスタムの GPG キーをインポートします
- SSL 証明書をインストールします
- アクティベーションキーを使ってシステムを Red Hat Network および特定のシステムグループとチャンネルに登録します
- パッケージの更新、再起動、Red Hat Network 設定の変更など、設定後のさまざまな作業を実行します



### 警告

ただし、設定作業にスクリプトを使用することに伴うリスクもあります。SSL 証明書などのセキュリティツールはスクリプト自体によってインストールされます。このため、証明書はシステム上にはまだ存在せず、トランザクションの処理に使用することはできません。これにより、**Satellite** になりすまして偽のデータを送信される可能性が生じます。実際にはすべての **Satellite** とクライアントのシステムは顧客のファイアウォールの背後で動作することになり、外部からのトラフィックは制限されることになるため、こうした可能性は軽減されることになります。登録は SSL 経由で行われるために保護されています。

ブートストラップスクリプトの **bootstrap.sh** は、Red Hat Network サーバーの **/var/www/html/pub/bootstrap/** ディレクトリーに自動的に配置されます。スクリプトはここからダウンロードされ、すべてのクライアントシステムで実行されます。次のセクションの説明どおり、準備や生成後の編集が必要になります。サンプルスクリプトについては「[Red Hat Network Bootstrap オプションの設定](#)」を参照してください。

#### 4.1.1.1. Red Hat Network Bootstrap インストールの準備

**Red Hat Network Bootstrap (rhn-bootstrap)** はクライアントシステムを正しく設定するために Red Hat Network のインフラストラクチャーを構成している他のコンポーネントに依存します。スクリプトを生成する前にまずこれらのコンポーネントの準備を行う必要があります。最初に行っておくべき準備を以下に示します。

- スクリプトで呼び出されるアクティベーションキーを生成します。アクティベーションキーは Red Hat Enterprise Linux システムの登録、Red Hat Network サービスレベルのエンタイトルメント付与、特定のチャンネルやシステムグループへのサブスクリाइプなどをすべて一度の動作で完了します。アクティベーションキーを使用するには利用可能な Management エンタイトルメントがあること、複数のアクティベーションキーを 1 度に組み込むには Provisioning エンタイトルメントが必要になる点に注意してください。Red Hat Satellite Web サイトの (Proxy 用の Red Hat Network 中央サーバーか Satellite の完全修飾ドメイン名のいずれかの) システムのカテゴリー内にある **アクティベーションキー** ページからアクティベーションキーを生成します。
- Red Hat では RPM をカスタムの GNU Privacy Guard (GPG) キーで署名しておくことを推奨しています。スクリプトから照合できるようにキーを使用可能にします。『Red Hat Satellite リファレンスガイド』の記載通りにキーを生成したら、そのキーを Red Hat Satellite Server の **/var/www/html/pub/** ディレクトリーに配置します。『Red Hat Satellite リファレンスガイド』の『カスタム GPG キーをインポートする』セクションを参照してください。
- 認証局の SSL パブリック証明書の配備にスクリプトを使用する場合は、その証明書またはその証明書を含むパッケージ (RPM) を該当する Red Hat Network サーバーで使用できるようにしてから、**--ssl-cert** オプションを使ってスクリプト生成時にこれを組み込みます。詳細は、『クライアント設定ガイド』の SSL インフラストラクチャーのセクションを参照してください。
- 再設定するシステムの種類に応じたブートストラップスクリプトを作成するために必要となる各種の値を手元に準備しておきます。**Red Hat Network Bootstrap** では再設定オプションの全セットが提供されるため、これを使用して、各種システムのタイプに適したブートストラップスクリプトをそれぞれ生成することができます。例えば、Web サーバーの再設定には

**bootstrap-web-servers.sh**、アプリケーションサーバーの場合は **bootstrap-app-servers.sh** を使用することができます。オプションの全一覧は「[Red Hat Network Bootstrap オプションの設定](#)」を参照してください。

#### 4.1.1.2. Bootstrap スクリプトの生成

これで必要なコンポーネントがすべて整いましたので、**Red Hat Network Bootstrap** を使用して必要なスクリプトを生成します。**Red Hat Satellite Server** または **Red Hat Satellite Proxy Server** に **root** としてログインし、**rhn-bootstrap** コマンドに必要なオプションと値を付けて実行します。オプションを付けずに実行すると、**bootstrap.sh** ファイルは **bootstrap/** サブディレクトリー内に作成されます。このサブディレクトリーにはホスト名、SSL 証明書、SSL と GPG の設定などのサーバーから派生した基本的な値や **client-config-overrides.txt** ファイルの呼び出しなどが含まれます。

少なくとも、次のような方法でアクティベーションキー、GPG キー、高度な設定オプションをスクリプトに組み込むことを強く推奨します。

- 「[Red Hat Network Bootstrap インストールの準備](#)」に記載されているエンタイトルメント要件を考慮し、**--activation-keys** オプションを使用してキーを含めます。
- スクリプトの生成時に、**--gpg-key** オプションを使ってキーのパスとファイル名を指定します。または、**--no-gpg** オプションを使うとクライアントシステム側でこの確認作業が行われなくなります。**Red Hat** ではセキュリティ対策として **--gpg-key** オプションの使用を推奨します。
- **--allow-config-actions** フラグを組み込むと、スクリプトで設定する全クライアントシステム上でリモートによる設定管理が可能になります。複数のシステムを同時に再設定する場合に便利です。
- **--allow-remote-commands** フラグを組み込むと、すべてのクライアントシステムでリモートによるスクリプトの使用が可能になります。設定管理と同様、複数システムの再設定に便利な機能です。

上記をすべて組み込むと次のようになります。

```
# rhn-bootstrap --activation-keys KEY1,KEY2 \
  --gpg-key /var/www/html/pub/MY_CORPORATE_PUBLIC_KEY \
  --allow-config-actions \
  --allow-remote-commands
```

実際のキーの名前が含まれるようにしてください。全オプションの一覧は、「[Red Hat Network Bootstrap オプションの設定](#)」を参照してください。

#### 4.1.1.3. Red Hat Network Bootstrap スクリプトの使い方

スクリプトの使用準備が完了したら、いつでもスクリプトを実行することができます。**Red Hat Satellite Server** または **Red Hat Satellite Proxy Server** にログインして、**/var/www/html/pub/bootstrap/** ディレクトリーに行き、次のコマンドを実行します。スクリプト名とホスト名はシステムタイプにあわせて適宜変更してください。

```
# cat bootstrap-EDITED-NAME.sh | ssh root@CLIENT_MACHINE1 /bin/bash
```

安全性は低くなりますが、**wget** または **curl** のいずれかを使用して各クライアントシステムからスクリプトを取り込み、実行していく方法もあります。各クライアントマシンにログインして次のコマンドを発行します。スクリプトとホスト名は適宜変更してください。

```
# wget -qO - \
    https://your-satellite.example.com/pub/bootstrap/bootstrap-EDITED-
NAME.sh \
    | /bin/bash
```

または、**curl** を使用した場合は次のようになります。

```
# curl -Sks \
    https://your-satellite.example.com/pub/bootstrap/bootstrap-EDITED-
NAME.sh \
    | /bin/bash
```

このスクリプトを各クライアントシステム上で実行した場合は、すべてが **Red Hat Network** サーバーを使用するように設定を行なう必要があります。

#### 4.1.1.4. Red Hat Network Bootstrap オプションの設定

**Red Hat Bootstrap** では、クライアントのブートストラップスクリプト作成に多くのコマンドラインオプションを提供しています。オプションの説明は次の表にありますが、これらのオプションが **Red Hat Network Server** にインストールしているツールのバージョンで利用できるか確認する必要があります。この確認を行うには、**rhn-bootstrap --help** を発行するか、または **man** ページを参照してください。

表4.1 Red Hat Network Bootstrap のオプション

オプション	説明
<b>-h、--help</b>	ブートストラップスクリプトの生成を行なう場合に使用するオプション一覧と共にヘルプ画面を表示します。
<b>--activation-keys=ACTIVATION_KEYS</b>	アクティベーションキーです。複数のエントリの場合はコンマで区切り空白は入れません。
<b>--overrides=OVERRIDES</b>	無視するファイル名です。デフォルトは <b>client-config-overrides.txt</b> です。
<b>--script=SCRIPT</b>	ブートストラップスクリプトのファイル名です。デフォルトは <b>bootstrap.sh</b> です。
<b>--hostname=HOSTNAME</b>	クライアントシステムの接続先となるサーバーの完全修飾ドメイン名 (FQDN) です。
<b>--ssl-cert=SSL_CERT</b>	組織のパブリック SSL 証明書へのパスで、パッケージまたは生の証明書になります。 <b>--pub-tree</b> オプションにコピーされます。"" の値で <b>--pub-tree</b> の検索を強制します。

オプション	説明
<b>--gpg-key=PGP_KEY</b>	企業または組織のパブリック GPG キーへのパスです (使用する場合)。 <b>--pub-tree</b> オプションによって指定した場所にコピーされます。
<b>--http-proxy=HTTP_PROXY</b>	<b>hostname:port</b> の形式のクライアントシステム用 HTTP プロキシの設定です。"" の値でこの設定が無効になります。
<b>--http-proxy-username=HTTP_PROXY_USERNAME</b>	HTTP プロキシの認証に使用する場合はユーザー名を指定します。"" の値を使うとこの設定は無効になります。
<b>--http-proxy-password=HTTP_PROXY_PASSWORD</b>	HTTP プロキシの認証に使用する場合はパスワードを指定します。
<b>--allow-config-actions</b>	Boolean です。このオプションを組み込むと、システムは Red Hat Network 経由の設定動作をすべて許可するようになります。このオプションを設定する場合は、アクティベーションキーなどを使って、特定の <b>rhncfg-*</b> パッケージをインストールする必要があります。
<b>--allow-remote-commands</b>	Boolean です。このオプションを組み込むと、システムは Red Hat Network 経由の任意のリモートコマンドを許可するようになります。おそらくアクティベーションキーを使って特定の <b>rhncfg-*</b> パッケージのインストールが必要になります。
<b>--no-ssl</b>	推奨できません。Boolean です。このオプションを組み込むと、クライアントシステムの SSL がオフになります。
<b>--no-gpg</b>	推奨できません。Boolean です。このオプションを組み込むと、クライアントシステム上で GPG のチェックがオフになります。
<b>--pub-tree=PUB_TREE</b>	変更は推奨できません。CA SSL 証書とパッケージが格納されるパブリックディレクトリーのツリーです (ブートストラップのディレクトリーとスクリプト群)。デフォルトは <b>/var/www/html/pub/</b> です。

オプション	説明
<b>--force</b>	推奨できません。Boolean です。このオプションを組み込むと、警告を無視してブートストラップスクリプトの生成を強制します。
<b>-v、--verbose</b>	詳細なメッセージを表示します。表示レベルが累積的に詳細となり、 <b>-vvv</b> で最大の詳細レベル表示になります。

#### 4.1.1.5. Red Hat Network Bootstrap 設定の手作業によるスクリプト化

本セクションでは、ブートストラップスクリプトを生成する際に **Red Hat Network Bootstrap** を使用しない方法について説明しています。記載されている説明に従うと、ゼロから独自のブートストラップスクリプトを作成することができるはずです。

集中管理を行なう場所に必要なファイル群を配備し、各クライアントで実行できるシンプルでスクリプト化が可能なコマンドを使ってそのファイルを取得し、インストールを行なう、というのが共通の目的です。本セクションでは、この目的に沿った単一のスクリプトで、企業や組織内のいずれのシステムからでも呼び出すことができるスクリプトの作成について見ていくことにします。

前のセクションで説明したコマンドをすべてまとめて適切な順序で並べると、以下のようなスクリプトを作成することができます。

```
# Reconfigure the clients to talk to the correct server.

perl -p -i -e 's/s/www\.rhns\.redhat\.com/proxy-or-sat\.example\.com/g' \
/etc/sysconfig/rhn/rhn_register \
/etc/sysconfig/rhn/up2date

# Install the SSL client certificate for your company's
# Red Hat Satellite Server or Red Hat Network Proxy Server.
rpm -Uvh http://proxy-or-sat.example.com/pub/rhn-org-trusted-ssl-cert-
*.noarch.rpm

# Reconfigure the clients to use the new SSL certificate.
perl -p -i -e 's/^sslCA/#sslCA/g;' \
/etc/sysconfig/rhn/up2date /etc/sysconfig/rhn/rhn_register
echo "sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT" \
>> /etc/sysconfig/rhn/up2date
echo "sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT" \
>> /etc/sysconfig/rhn/rhn_register

# Download the GPG key needed to validate custom packages.
wget -O - -q http://proxy-or-sat.example.com.com/pub/YOUR-RPM-GPG-KEY

# Import that GPG key to your GPG keyring.
rpm --import /path/to/YOUR-RPM-GPG-KEY
```

このスクリプトは完全で反復可能なプロセスで構成されています。**Red Hat Satellite** クライアントを設定して、**Red Hat Satellite Proxy Server** や **Red Hat Satellite** への登録の準備までを完了させることができます。Red Hat Satellite サーバーの URL、そのパブリックディレクトリー、および実際の GPG キーなど、必要な値をスクリプト内にあるプレースホルダーに必ず入力してください。また、使用環境に応じて追加の修正が必要となる場合があります。本スクリプトはほとんどこのままでも機能しますが参考として使用するようにしてください。

コンポーネントと同様に、このスクリプトも集中管理を行なう場所に配置することができます。スクリプトをサーバーの **/pub/** ディレクトリー内に配置し、そのサーバー上で **wget -O-** を実行し、その出力をシェルセッションに入力すると、各クライアントから単一のコマンドを使用してブートストラップの全プロセスを実行することができるようになります。

```
# wget -O - http://proxy-or-sat.example.com.com/pub/bootstrap_script |
bash
```



### 警告

Web 接続で取得して入力されるスクリプトの内容をシェルスクリプトとして直接実行することにはそれなりの危険も伴います。したがってソースとなるサーバーの安全性を必ず確保することが非常に重要となります。

次に、この1行のコマンドを、ネットワーク上の全システムに対して起動することができます。また、既存のキックスタートスクリプトの **%post** セクションにこのスクリプトを追加することもできます。

#### 4.1.1.6. キックスタートの実装

システムに設定変更を加えるタイミングとしては、そのシステムをはじめて構成する時が最も適しています。すでにキックスタートを効率的に使用している場合、ブートストラップを行うスクリプトをそのプロセスに追加するのが良いでしょう。

設定に関する問題がすべて解決したら、**rhn-setup RPM** に同梱されている **rhnreg\_ks** ユーティリティを使用して、ローカルの **Red Hat Network Server** にシステムを登録することができます。本セクションではシステムを登録するために **rhnreg\_ks** を正しく使用方法について説明しています。

**rhnreg\_ks** ユーティリティは、**アクティベーションキー** を使用して1回の操作でシステムの登録からエンタイトルメントの付与、指定チャンネルへのサブスクリプションまで一度に行います。アクティベーションキーについての詳細は『**Red Hat Network Management Reference Guide**』の「**RHN Website**」と「**Red Hat Update Agent**」のセクションを参照してください。

以下のコメント付きキックスタートファイルは、**Red Hat Satellite** を使用してどのようにシステムが設定されていくのを見る上で理想的な設定例となります。

```
# Generic 7.2 kickstart for laptops in the Widget Corporation (widgetco)

# Standard kickstart options for a network-based install. For an
# explanation of these options, consult the Red Hat Enterprise Linux
# Customization Guide.

lang en_US
langsupport --default en_US en_US
```

```

keyboard defkeymap
network --bootproto dhcp
install
url --url ftp://ftp.widgetco.com/pub/redhat/linux/7.2/en/os/i386
zerombr yes
clearpart --all
part /boot --size 128 --fstype ext3 --ondisk hda
part / --size 2048 --grow --fstype ext3 --ondisk hda
part /backup --size 1024 --fstype ext3 --ondisk hda
part swap --size 512 --ondisk hda
bootloader --location mbr
timezone America/New_York
rootpw --iscrypted $1$78Jnap82Hnd0PsjnC8j3sd2Lna/Hx4.
auth --useshadow --enablemd5 --krb5realm .COM --krb5kdc auth.widgetco.com \
    --krb5adminserver auth.widgetco.com
mouse --emulthree genericps/2
xconfig --card "S3 Savage/MX" --videoram 8192 --resolution 1024x768 \
    --depth 16 --defaultdesktop=GNOME --startxonboot --noprobe \
    --hsync 31.5-48.5 --vsync 40-70

reboot

# Define a standard set of packages. Note: Red Hat Network client
# packages are found in the Base channel. This is quite a minimal
# set of packages

%packages
@ Base
@ Utilities
@ GNOME
@ Laptop Support
@ Dialup Support
@ Software Development
@ Graphics and Image Manipulation
@ Games and Entertainment
@ Sound and Multimedia Support

%post
( # Note that we run the entire %post section as a subshell for logging.

# Use the one-line command for the bootstrap script. Assuming that the
# script has been properly configured, it should prepare the system
# fully for usage of local Red Hat Network Servers.

wget -O- http://proxy-or-sat.example.com/pub/bootstrap_script | /bin/bash

# The following is an example of rhnreg_ks usage, the kickstart
# utility for rhn_register. This demonstrates the usage of the
# --activationkey flag, which describes an activation key. For example,
# this activation key could be set up in the Web interface to join this
# system to the "Laptops" group and the local "Laptop Software"
# channel. Note that this section applies only to Proxy server users, as
# this step is handled by the Satellite bootstrap script.
#

```



```
# For more information about activation keys, consult the Red Hat Network
# Management Reference Guide.

/usr/sbin/rhnreg_ks --activationkey=6c933ea74b9b002f3ac7eb99619d3374

# End the subshell and capture any output to a post-install log file.
) 1>/root/post_install.log 2>&1
```

#### 4.1.1.7. サンプルのブートストラップスクリプト

Red Hat Satellite Server インストールプログラムによって生成される `/var/www/html/pub/bootstrap/bootstrap.sh` スクリプトでは、Red Hat Satellite Server にクライアントシステムをアクセスさせるための再設定を簡単に行なえる機能を提供しています。Red Hat Satellite Server および Red Hat Satellite Proxy Server をご利用のお客様は、**RHN Bootstrap** ツールでこの機能をご使用いただけます。使用状況に合わせてスクリプトを修正してから、各クライアントマシンでそのスクリプトを実行します。

詳細については、サンプルとハッシュマーク (#) で始まるコメントを確認してください。『スタートガイド』の手順に従ってスクリプトの使用準備を行います。

```
#!/bin/bash
echo "Red Hat Satellite Server Client bootstrap script v4.0"

# This file was autogenerated. Minor manual editing of this script (and
# possibly the client-config-overrides.txt file) may be necessary to
# complete
# the bootstrap setup. Once customized, the bootstrap script can be
# triggered
# in one of two ways (the first is preferred):
#
# (1) centrally, from the RHN Satellite Server via ssh (i.e., from the
#     RHN Satellite Server):
#     cd /var/www/html/pub/bootstrap/
#     cat bootstrap-<edited_name>.sh | ssh root@<client-hostname>
/bin/bash
#
# ...or...
#
# (2) in a decentralized manner, executed on each client, via wget or
# curl:
#     wget -qO- https://<hostname>/pub/bootstrap/bootstrap-
#     <edited_name>.sh | /bin/bash
#     ...or...
#     curl -Sks https://<hostname>/pub/bootstrap/bootstrap-
#     <edited_name>.sh | /bin/bash

# SECURITY NOTE:
# Use of these scripts via the two methods discussed is the most
# expedient
# way to register machines to your RHN Satellite Server. Since "wget" is
# used
# throughout the script to download various files, a "Man-in-the-middle"
# attack is theoretically possible.
#
# The actual registration process is performed securely via SSL, so the
```

```
risk
# is minimized in a sense. This message merely serves as a warning.
# Administrators need to appropriately weigh their concern against the
# relative security of their internal network.

# PROVISIONING/KICKSTART NOTE:
# If provisioning a client, ensure the proper CA SSL public certificate
is
# configured properly in the post section of your kickstart profiles
(the
# RHN Satellite or hosted web user interface).

# UP2DATE/RHN_REGISTER VERSIONING NOTE:
# This script will not work with very old versions of up2date and
# rhn_register.

echo
echo
echo "MINOR MANUAL EDITING OF THIS FILE MAY BE REQUIRED!"
echo
echo "If this bootstrap script was created during the initial installation"
echo "of an RHN Satellite, the ACTIVATION_KEYS, and ORG_GPG_KEY values
will"
echo "probably *not* be set (see below). If this is the case, please do
the"
echo "following:"
echo " - copy this file to a name specific to its use."
echo "   (e.g., to bootstrap-SOME_NAME.sh - like bootstrap-web-
servers.sh.)"
echo " - on the website create an activation key or keys for the
system(s) to"
echo "   be registered."
echo " - edit the values of the VARIABLES below (in this script) as"
echo "   appropriate:"
echo "   - ACTIVATION_KEYS needs to reflect the activation key(s)
value(s)"
echo "     from the website. XKEY or XKEY,YKEY"
echo "   - ORG_GPG_KEY needs to be set to the name(s) of the corporate
public"
echo "     GPG key filename(s) (residing in /var/www/html/pub) if
appropriate. XKEY or XKEY,YKEY"
echo
echo "Verify that the script variable settings are correct:"
echo "   - CLIENT_OVERRIDES should be only set differently if a
customized"
echo "     client-config-overrides-VER.txt file was created with a
different"
echo "     name."
echo "   - ensure the value of HOSTNAME is correct."
echo "   - ensure the value of ORG_CA_CERT is correct."
echo
echo "Enable this script: comment (with #'s) this block (or, at least
just"
echo "the exit below)"
echo
```

```

exit 1

# can be edited, but probably correct (unless created during initial
install):
# NOTE: ACTIVATION_KEYS *must* be used to bootstrap a client machine.
ACTIVATION_KEYS=
ORG_GPG_KEY=

# can be edited, but probably correct:
CLIENT_OVERRIDES=client-config-overrides.txt
HOSTNAME=yoursatellite.hostname.com

ORG_CA_CERT=RHN-ORG-TRUSTED-SSL-CERT
ORG_CA_CERT_IS_RPM_YN=0

USING_SSL=1
USING_GPG=1

REGISTER_THIS_BOX=1

ALLOW_CONFIG_ACTIONS=1
ALLOW_REMOTE_COMMANDS=1

FULLY_UPDATE_THIS_BOX=1

# Set if you want to specify profilename for client systems.
# NOTE: Make sure it's set correctly if any external command is used.
#
# ex. PROFILENAME="foo.example.com" # For specific client system
#     PROFILENAME=`hostname -s`      # Short hostname
#     PROFILENAME=`hostname -f`      # FQDN
PROFILENAME="" # Empty by default to let it be set automatically.

#
# -----
# DO NOT EDIT BEYOND THIS POINT -----
# -----
#
# an idea from Erich Morisse (of Red Hat).
# use either wget *or* curl
# Also check to see if the version on the
# machine supports the insecure mode and format
# command accordingly.

if [ -x /usr/bin/wget ] ; then
    output=`LANG=en_US /usr/bin/wget --no-check-certificate 2>&1`
    error=`echo $output | grep "unrecognized option"`
    if [ -z "$error" ] ; then
        FETCH="/usr/bin/wget -q -r -nd --no-check-certificate"
    else
        FETCH="/usr/bin/wget -q -r -nd"
    fi

```

```
else
    if [ -x /usr/bin/curl ] ; then
        output=`LANG=en_US /usr/bin/curl -k 2>>&1`
        error=`echo $output | grep "is unknown"`
        if [ -z "$error" ] ; then
            FETCH="/usr/bin/curl -Sks0"
        else
            FETCH="/usr/bin/curl -Ss0"
        fi
    fi
fi

HTTP_PUB_DIRECTORY=http://${HOSTNAME}/pub
HTTPS_PUB_DIRECTORY=https://${HOSTNAME}/pub
if [ $USING_SSL -eq 0 ] ; then
    HTTPS_PUB_DIRECTORY=${HTTP_PUB_DIRECTORY}
fi

INSTALLER=up2date
if [ -x /usr/bin/zypper ] ; then
    INSTALLER=zypper
elif [ -x /usr/bin/yum ] ; then
    INSTALLER=yum
fi
echo
echo "UPDATING RHN_REGISTER/UP2DATE CONFIGURATION FILES"
echo "-----"
echo "* downloading necessary files"
echo "  client_config_update.py..."
rm -f client_config_update.py
$FETCH ${HTTPS_PUB_DIRECTORY}/bootstrap/client_config_update.py
echo "  ${CLIENT_OVERRIDES}..."
rm -f ${CLIENT_OVERRIDES}
$FETCH ${HTTPS_PUB_DIRECTORY}/bootstrap/${CLIENT_OVERRIDES}

if [ ! -f "client_config_update.py" ] ; then
    echo "ERROR: client_config_update.py was not downloaded"
    exit 1
fi
if [ ! -f "${CLIENT_OVERRIDES}" ] ; then
    echo "ERROR: ${CLIENT_OVERRIDES} was not downloaded"
    exit 1
fi

echo "* running the update scripts"
if [ -f "/etc/sysconfig/rhn/rhn_register" ] ; then
    echo "  . rhn_register config file"
    /usr/bin/python -u client_config_update.py
    /etc/sysconfig/rhn/rhn_register ${CLIENT_OVERRIDES}
fi
echo "  . up2date config file"
/usr/bin/python -u client_config_update.py /etc/sysconfig/rhn/up2date
${CLIENT_OVERRIDES}

if [ ! -z "$ORG_GPG_KEY" ] ; then
    echo
```

```

    echo "* importing organizational GPG key"
    for GPG_KEY in $(echo "$ORG_GPG_KEY" | tr ", " " "); do
rm -f ${GPG_KEY}
$FETCH ${HTTPS_PUB_DIRECTORY}/${GPG_KEY}
# get the major version of up2date
# this will also work for RHEL 5 and systems where no up2date is
installed
res=$(LC_ALL=C rpm -q --queryformat '%{version}' up2date | sed -e
's/\..*/g')
if [ "x$res" == "x2" ] ; then
    gpg $(up2date --gpg-flags) --import $GPG_KEY
else
    rpm --import $GPG_KEY
fi
done
fi

echo
echo "* attempting to install corporate public CA cert"
if [ $ORG_CA_CERT_IS_RPM_YN -eq 1 ] ; then
    rpm -Uvh --force --replacefiles --replacepkgs
${HTTPS_PUB_DIRECTORY}/${ORG_CA_CERT}
else
    rm -f ${ORG_CA_CERT}
    $FETCH ${HTTPS_PUB_DIRECTORY}/${ORG_CA_CERT}
    mv ${ORG_CA_CERT} /usr/share/rhn/

fi
if [ "$INSTALLER" == zypper ] ; then
    if [ $ORG_CA_CERT_IS_RPM_YN -eq 1 ] ; then
        # get name from config
        ORG_CA_CERT=$(basename $(sed -n 's/^sslCACert *= */p'
/etc/sysconfig/rhn/up2date))
    fi
    test -e "/etc/ssl/certs/${ORG_CA_CERT}.pem" || {
        test -d "/etc/ssl/certs" || mkdir -p "/etc/ssl/certs"
        ln -s "/usr/share/rhn/${ORG_CA_CERT}"
"/etc/ssl/certs/${ORG_CA_CERT}.pem"
    }
    test -x /usr/bin/c_rehash && /usr/bin/c_rehash /etc/ssl/certs/ | grep
"${ORG_CA_CERT}"
fi

echo
echo "REGISTRATION"
echo "-----"
# Should have created an activation key or keys on the RHN Satellite
Server's
# website and edited the value of ACTIVATION_KEYS above.
#
# If you require use of several different activation keys, copy this file
and
# change the string as needed.
#
if [ -z "$ACTIVATION_KEYS" ] ; then
    echo "**** ERROR: in order to bootstrap RHN clients, an activation key

```

```

or keys"
    echo "                must be created in the RHN web user interface, and
the"
    echo "                corresponding key or keys string (XKEY,YKEY,...) must
be mapped to"
    echo "                the ACTIVATION_KEYS variable of this script."
    exit 1
fi

if [ $REGISTER_THIS_BOX -eq 1 ] ; then
    echo "* registering"
    files=""
    directories=""
    if [ $ALLOW_CONFIG_ACTIONS -eq 1 ] ; then
        for i in "/etc/sysconfig/rhn/allowed-actions
/etc/sysconfig/rhn/allowed-actions/configfiles"; do
            [ -d "$i" ] || (mkdir -p $i && directories="$directories $i")
        done
        [ -f /etc/sysconfig/rhn/allowed-actions/configfiles/all ] ||
files="/etc/sysconfig/rhn/allowed-actions/configfiles/all"
        [ -n "$files" ] && touch $files
    fi
    if [ -z "$PROFILENAME" ] ; then
        profilename_opt=""
    else
        profilename_opt="--profilename=$PROFILENAME"
    fi
    /usr/sbin/rhnreg_ks --force --activationkey "$ACTIVATION_KEYS"
$profilename_opt
    RET="$?"
    [ -n "$files" ] && rm -f $files
    [ -n "$directories" ] && rmdir $directories
    if [ $RET -eq 0 ]; then
        echo
        echo "**** this system should now be registered, please verify ****"
        echo
    else
        echo
        echo "**** Error: Registering the system failed."
        echo
        exit 1
    fi
else
    echo "* explicitly not registering"
fi

if [ $ALLOW_CONFIG_ACTIONS -eq 1 ] ; then
    echo
    echo "* setting permissions to allow configuration management"
    echo "  NOTE: use an activation key to subscribe to the tools"
    if [ "$INSTALLER" == zypper ] ; then
        echo "        channel and zypper install/update rhncfg-actions"
    elif [ "$INSTALLER" == yum ] ; then
        echo "        channel and yum upgrade rhncfg-actions"
    else
        echo "        channel and up2date rhncfg-actions"
    fi
fi

```

```

fi
if [ -x "/usr/bin/rhn-actions-control" ] ; then
    rhn-actions-control --enable-all
    rhn-actions-control --disable-run
else
    echo "Error setting permissions for configuration management."
    echo "    Please ensure that the activation key subscribes the"
if [ "$INSTALLER" == zypper ] ; then
    echo "    system to the tools channel and zypper install/update
rhncfg-actions."
elif [ "$INSTALLER" == yum ] ; then
    echo "    system to the tools channel and yum updates rhncfg-
actions."
else
    echo "    system to the tools channel and up2dates rhncfg-
actions."
fi
fi
exit
fi
fi

if [ $ALLOW_REMOTE_COMMANDS -eq 1 ] ; then
    echo
    echo "* setting permissions to allow remote commands"
    echo "    NOTE: use an activation key to subscribe to the tools"
    if [ "$INSTALLER" == zypper ] ; then
        echo "        channel and zypper update rhncfg-actions"
    elif [ "$INSTALLER" == yum ] ; then
        echo "        channel and yum upgrade rhncfg-actions"
    else
        echo "        channel and up2date rhncfg-actions"
    fi
    if [ -x "/usr/bin/rhn-actions-control" ] ; then
        rhn-actions-control --enable-run
    else
        echo "Error setting permissions for remote commands."
        echo "    Please ensure that the activation key subscribes the"
        if [ "$INSTALLER" == zypper ] ; then
            echo "    system to the tools channel and zypper updates rhncfg-
actions."
        elif [ "$INSTALLER" == yum ] ; then
            echo "    system to the tools channel and yum updates rhncfg-
actions."
        else
            echo "    system to the tools channel and up2dates rhncfg-
actions."
        fi
    fi
    fi
fi

echo
echo "OTHER ACTIONS"
echo "-----"
if [ $FULLY_UPDATE_THIS_BOX -eq 1 ] ; then
    if [ "$INSTALLER" == zypper ] ; then

```

```

        echo "zypper --non-interactive up zypper zypp-plugin-spacewalk;
rhn-profile-sync; zypper --non-interactive up (conditional)"
        elif [ "$INSTALLER" == yum ] ; then
            echo "yum -y upgrade yum yum-rhn-plugin; rhn-profile-sync; yum
upgrade (conditional)"
        else
            echo "up2date up2date; up2date -p; up2date -uf (conditional)"
        fi
    else
        if [ "$INSTALLER" == zypper ] ; then
            echo "zypper --non-interactive up zypper zypp-plugin-spacewalk;
rhn-profile-sync"
        elif [ "$INSTALLER" == yum ] ; then
            echo "yum -y upgrade yum yum-rhn-plugin; rhn-profile-sync"
        else
            echo "up2date up2date; up2date -p"
        fi
    fi
echo "but any post configuration action can be added here. "
echo "-----"
if [ $FULLY_UPDATE_THIS_BOX -eq 1 ] ; then
    echo "* completely updating the box"
else
    echo "* ensuring $INSTALLER itself is updated"
fi
if [ "$INSTALLER" == zypper ] ; then
    zypper ref -s
    zypper --non-interactive up zypper zypp-plugin-spacewalk
    if [ -x /usr/sbin/rhn-profile-sync ] ; then
        /usr/sbin/rhn-profile-sync
    else
        echo "Error updating system info in RHN Satellite."
        echo "    Please ensure that rhn-profile-sync is installed and
rerun it."
    fi
    if [ $FULLY_UPDATE_THIS_BOX -eq 1 ] ; then
        zypper --non-interactive up
    fi
elif [ "$INSTALLER" == yum ] ; then
    /usr/bin/yum -y upgrade yum yum-rhn-plugin
    if [ -x /usr/sbin/rhn-profile-sync ] ; then
        /usr/sbin/rhn-profile-sync
    else
        echo "Error updating system info in RHN Satellite."
        echo "    Please ensure that rhn-profile-sync is installed and
rerun it."
    fi
    if [ $FULLY_UPDATE_THIS_BOX -eq 1 ] ; then
        /usr/bin/yum -y upgrade
    fi
else
    /usr/sbin/up2date up2date
    /usr/sbin/up2date -p
    if [ $FULLY_UPDATE_THIS_BOX -eq 1 ] ; then
        /usr/sbin/up2date -uf

```



```
fi
fi
echo "-bootstrap complete-"
```

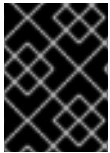
## 4.2. SATELLITE を使ったシステムの管理

システムが Red Hat Satellite に登録されると、これらのシステムは **システム** タブの **概要** ページに表示されます。**概要** ページには、システムの要約が記載されます。これには、システムの状態、関連するエラータとパッケージの数、システムが属するベースチャンネル、およびエンタイトルメントのレベルなどが含まれます。さらに、システムはいくつかの方法で管理することができます。

### 4.2.1. 個別システムの管理

Satellite 管理者は、**システム** タブの **概要** ページで、システム名をクリックすることにより、システムを個別に管理できます。ここから、システムの **詳細** タブに移動します。「詳細」ページには、特定のシステム情報や、システムに固有の他の識別子を記載する数多くのタブがあります。

#### 4.2.1.1. システムの電力管理の制御



#### 重要

この機能は、x86\_64 アーキテクチャーの Red Hat Satellite サーバーのみで利用できません。

Satellite は、**cobbler** の IPMI との統合によって物理マシンの電源管理機能を提供します。電源管理機能を使用する前に、サーバーに IPMI フェンシングエージェントをインストールする必要があります。以下のコマンドを使用してフェンシングエージェントをインストールします。

```
# yum install fence-agents
```

このコマンドを実行すると、IPMI エージェントを含むフェンシングエージェントのセットがインストールされます。

フェンシングエージェントのインストール完了後、IPMI フェンシングエージェントを使用するための設定が必要になります。`/etc/rhn/rhn.conf` ファイルを編集し、以下のプロパティをファイルの最後に追加します。

```
java.power_management.types = ipmilan
```

ファイルを保存し、Satellite サーバーを再起動します。

```
# rhn-satellite restart
```

以下の手順にしたがって、Satellite の電源管理機能を使用します。

1. **システム** → **システム** と移動します。
2. 管理するシステムを選択します。
3. **詳細** → **プロパティ** と選択し、システムの **Provisioning** エンタイトルメントが有効であることを確認します。**付属エンタイトルメント** の **Provisioning** をチェックし、**プロパティの更新** をクリックします。

4. **Provisioning** → **電力管理** と選択します。
5. フェンシングエージェント **タイプ** の **IPMI** を選択します。
6. **IPMI 電源管理ボードのネットワークアドレス** を入力します。**Satellite** はフェンシングエージェントを使用して電源管理ボードと通信します。
7. 電源管理ボードの **ユーザー名** および **パスワード** を入力します。
8. **保存** をクリックします。

**電源オン**、**電源オフ**、および **再起動** を含むシステムの基本的な電源管理を制御できるようになります。

#### 4.2.1.2. システムのハードウェアプロファイルの表示

**ハードウェア** タブは、システムのハードウェアプロファイルを表示します。これには、**DMI**、**ネットワーク**の**詳細**、およびマシンが使用している**ドライバーの詳細**などの情報が含まれます。これは、マシンを識別する際や、ベンダー情報が必要なときに役に立ちます。

システム上のハードウェアのいずれかが変更されるか、またはリストが未完成な場合に、**ハードウェア更新のスケジュール** をクリックして、このページのハードウェアリストを更新することもできます。

1. **システム** → **システム** の順にクリックします。
2. システムの名前をクリックします。
3. **ハードウェア** をクリックします。

#### 4.2.1.3. Satellite からのシステム再起動のスケジュールリング

ここでは、システムのリモート再起動をスケジュールします。これは、システム管理者が **Satellite** の近くにおらず、トラブルシューティングの目的で再起動を必要とする場合に適しています。

1. **システム** → **システム** の順にクリックします。
2. システムの名前をクリックします。
3. **システムイベント** 以下のページの右側にある、**システムの再起動をスケジュール** をクリックします。

#### 4.2.1.4. システムのベースチャンネルのサブスクリプションの変更

ベースチャンネルのサブスクリプションは、登録時に **Red Hat Satellite** に送信されたハードウェアとシステムのプロファイルに基づいてデフォルトで選択されます。ただし、この設定は、問題が発生する場合には変更されることがあります。システムのベースチャンネルを変更するには、以下を実行します。

1. **システム** → **システム** の順にクリックします。
2. システムの名前をクリックします。
3. ページの左側にある **サブスクライブしているチャンネル** の下で、**チャンネルサブスクリプションの変更** をクリックします。

4. ページの下部にある **ベースソフトウェアチャンネルセクション**にスクロールし、適用するベースチャンネルを選択します。
5. **確認** をクリックします。



### 注記

システムのベースチャンネルサブスクリプションを変更する場合、**Satellite** は以前にサブスクライブしたチャンネルすべてからシステムのサブスクライブを中止し、選択した新規のベースソフトウェアチャンネルにシステムをサブスクライブさせます。また、追加の子チャンネルを再度追加する必要があります。

#### 4.2.1.5. システムの子チャンネルのサブスクリプションの変更

追加のチャンネルは、**Red Hat Network Tools** チャンネルのオプションパッケージや仮想化支援機能などの、システムが持つ固有の要件に基づいて追加される場合があります。システムを追加のチャンネルにサブスクライブさせるには、以下の手順に従います。

1. システム → システムの順にクリックします。
2. システムの名前をクリックします。
3. ページの左側にある **サブスクライブしているチャンネルの下で、チャンネルサブスクリプションの変更** をクリックします。
4. チェックボックスにチェックマークを付けることにより、システムが必要な子チャンネルを選択します。必要な数だけ選択することができます。一部のチャンネルは、選択されると追加のソフトウェアエンタイトルメントを使用する可能性があることに注意してください。
5. **サブスクリプションの変更** をクリックします。

#### 4.2.1.6. Provisioning/Monitoring エンタイトルメントのシステムへの追加

**Provisioning** または **Monitoring** エンタイトルメントは、これらのエンタイトルメントが利用可能な場合は、**Satellite** 内のシステムに追加できます。これらのエンタイトルメントは、以下の要件を持つシステムと関連します。

- **Provisioning**- このエンタイトルメントは、キックスタートを使用する機能、パッケージのロールバックおよび設定ファイル管理などを必要とするシステムで要求されます。
- **Monitoring**- このエンタイトルメントにより、**Monitoring** のエンタイトルメントを付与されたクライアントを持つ **Satellite** が、管理者に対して、システムパフォーマンスの問題が重大になる前に通知できるようにします。

1. システム → システムの順にクリックします。
2. システムの名前をクリックします。
3. **詳細** → **プロパティ** の順にクリックします。
4. **付属エンタイトルメント** セクションの必須のエンタイトルメントを選択します。
5. **プロパティの更新** をクリックします。

#### 4.2.1.7. 新規パッケージのリモートインストール

リモートでインストールするパッケージを選択する機能は、システム内の新規または変更された要件を管理するために使用できます。リモートインストールのスケジュール時にシステムの電源が切れている場合、この動作は、システムがバックアップされてから実行されます。

Red Hat Satellite からパッケージをインストールするには、以下の手順に従います。

1. システム → システムの順にクリックします。
2. システムの名前をクリックします。
3. ソフトウェア → パッケージ → インストールの順にクリックします。
4. システムにインストールするパッケージを選択します。
5. 選択したパッケージをインストールをクリックします。
6. 特定の時間か、またはできるだけ早くインストールするようスケジュールすることを選択します。
7. **確認** をクリックします。



#### 注記

パッケージをシステムにインストールするために追加のコマンドが必要な場合、**パッケージインストールへのリモートコマンドの追加** をクリックします。これを起動するのに必要なスクリプトを追加し、**リモートコマンドの確認とパッケージインストールのスケジュール** をクリックします。この追加の手順は、特殊な設定要件を持つシステムには役に立つ場合があります。

#### 4.2.1.8. パッケージのリモートでのアップグレード

Red Hat Satellite からパッケージをリモートでアップグレードするには:

1. システム → システムの順にクリックします。
2. システムの名前をクリックします。
3. ソフトウェア → パッケージ → アップグレードの順にクリックします。
4. アップグレードするパッケージを選択します。
5. 特定の時間か、またはできるだけ早くインストールするようスケジュールすることを選択します。
6. **確認** をクリックします。



#### 注記

パッケージをシステムにインストールするために追加のコマンドが必要な場合、**パッケージインストールへのリモートコマンドの追加** をクリックします。これを起動するのに必要なスクリプトを追加し、**リモートコマンドの確認とパッケージインストールのスケジュール** をクリックします。この追加の手順は、特殊な設定要件を持つシステムには役に立つ場合があります。

#### 4.2.1.9. 変更からのシステムのロック

システムを変更しないまま維持することが要件となる場合、システムをロックすることができます。ロックされたシステムは、システムがシステムグループ内にある場合でも、パッケージのアップグレードやシステムで変更される動作のいずれも受信しません。

システムをロックするには:

1. システム → システムの順にクリックします。
2. システムの名前をクリックします。
3. システム情報 セクションの **ロックの状態** フィールドで、**システムのロック** をクリックします。



#### 注記

保留中の動作を持つシステムはロックできません。スケジュールされた動作をキャンセルするには、**イベント → イベントの取り消し**の順にクリックします。

#### 4.2.1.10. システム状態のウェイト/乗数の設定

Satellite に含まれるシステム状態レポートは、登録されたシステムをスコア順に表示します。スコアは、システムに関連するエラータの合計によって決定されます。デフォルトの重さでは、重大なセキュリティエラータに最大重量が追加され、エンハンスエラータに最小重量が追加されます。このレポートを使用して、Satellite に登録されたシステム上のメンテナンス作業の優先順位をつけることもできます。以下の値は、`/usr/share/rhn/config-defaults/rhn_java.conf` に格納されたデフォルトのシステム状態の値になります。

```
# multiplier for critical security errata
java.sc_crit = 32
# multiplier for important security errata
java.sc_imp = 16
# multiplier for moderate important security errata
java.sc_mod = 8
# multiplier for low important security errata
java.sc_low = 4
# multiplier for bugfix errata
java.sc_bug = 2
# multiplier for enhancement errata
java.sc_enh = 1
```

`/etc/rhn/rhn.conf` ファイルに追加すると上書きできます。スコアが高いほど、レポートのシステムの重要度が高くなります。

各エラータタイプのデフォルトの重み/乗数を変更するには:

1. Satellite サーバーに **root** としてログインします。
2. 選択したテキストエディターで、`/etc/rhn/rhn.conf` を編集します。

```
# vim /etc/rhn/rhn.conf
```

3. `/etc/rhn/rhn.conf` ファイルに変更したいカスタム値を追加します。

```
java.sc_imp = desired_value
```

```
java.sc_mod = desired_value
java.sc_low = desired_value
java.sc_bug = desired_value
java.sc_enh = desired_value
```

4. `/etc/rhn/rhn.conf` ファイルを保存します。

5. Satellite のサービスを再起動します。

```
# rhn-satellite restart
```

## 注記

Red Hat Satellite 5.5 では、状態レポートと重み/乗数の値は Red Hat Network API 呼び出しを使用して取得することもできます。以下の呼び出しについてさらに詳しくは、『API の概要 (API Overview)』を参照してください。

- **system.getSystemCurrencyMultipliers** - この API 呼び出しは、現在の重み/乗数の設定についての情報を提供します。
- **system.getSystemCurrencyScores** - この API 呼び出しは、システムのリスト、スコアおよびそれぞれのタイプの適用されるエラータの数を返します。

### 4.2.2. システムグループの管理

システムの使用またはロールに関連して必要となるかどうかにかかわらず、管理者が組織のシステムの論理単位を作成しようとする場合、Red Hat Satellite は、「システムグループ」機能からこれを実行する機能を提供します。システムを論理単位に分離することにより、管理者はグループ内の全システムにまたがって同様のパッケージバージョンを維持し、より管理の容易な標準ビルドを維持することができます。

本セクションの手順では、システムグループ管理の基本的な手順を提供すると共に、システムグループのセットアップを支援します。

#### 4.2.2.1. システムのシステムグループへの追加

以下の手順は、個々のシステムをシステムグループに追加する方法について示しています。

1. システム → システムの順にクリックします。
2. システムの名前をクリックします。
3. グループ → 参加 の順にクリックします。
4. システムを追加する必要のある 1 つまたは複数のグループを選択します。
5. 選択したグループに参加をクリックします。

#### 4.2.2.2. 複数のシステムのシステムグループへの追加

複数のグループを、個別にではなく一度にシステムグループに追加することができます。システムグループに追加するシステムが複数ある場合は、これが最も効果的な動作になります。

1. システム → システムグループをクリックします。

2. 必要なシステムグループをクリックします。
3. 目的のシステムサブタブをクリックします。



#### 注記

目的のシステムは、**Satellite** 上にある有効なシステムであり、グループに追加することができます。

4. システムの追加 をクリックします。

#### 4.2.2.3. グループ管理者のグループへの追加

以下の手順は、特定のグループの管理者としてユーザーを追加する方法について示しています。

1. システム → システムグループ をクリックします。
2. 必要なシステムグループをクリックします。
3. 管理者 サブタブ をクリックします。
4. グループ管理者にする必要のあるユーザー名をクリックします。
5. 更新 をクリックします。



#### 注記

組織の管理者は、**Satellite** 内のすべてのシステムを管理することができます。

#### 4.2.2.4. システムグループからのシステムの削除

以下は、特定のシステムグループからシステムを削除する手順です。

1. システム → システムグループ をクリックします。
2. 必要なシステムグループをクリックします。
3. システム をクリックします。
4. システムグループから削除するすべてのシステムを選択します。
5. システムの削除 をクリックします。

#### 4.2.2.5. システム内の影響を受けたシステムへのエラータの適用

以下の方法を使用して、エラータを複数のシステムにリモートで適用することができます。

1. システム → システムグループ をクリックします。
2. 必要なシステムグループをクリックします。
3. エラータ サブタブ をクリックします。
4. 影響を受けたシステムに適用する アドバイザリ を選択します。

5. **影響を受けるシステム**をクリックして、エラータの影響を受けるシステムの一覧を表示します。
6. エラータを適用するシステムを選択するか、または**すべてを選択**を選択します。
7. エラータの**適用**をクリックします。

#### 4.2.3. システム設定マネージャーでのシステムの管理

システム設定マネージャー (System Set Manager: SSM) は、管理者が Red Hat Satellite サーバー内の多数のシステムを扱い、かつシステム管理、維持管理および配備を実行できるようにします。SSM は、異なるシステムグループにある複数のシステムが同様の維持管理、設定または配備に関連した変更を必要とする場合に使用されます。

システム設定マネージャーが提供する他のさまざまな機能には以下があります。

- エラータ更新のスケジュール、およびパッケージのアップグレード、インストールおよび削除
- システムのチャンネルメンバーシップ、設定チャンネルの配備および設定チャンネルのサブスクリプションの管理
- システムのプロビジョニング、リモートコマンドの実行、およびスナップショットのロールバック用のシステムのタグ付け
- システムの別の組織への移行、および Satellite 内の選択したシステムのカスタム値の設定

SSM を使用するために、システムが SSM に追加されていることを確認します。SSM の各種機能を使用する前に、まずこれを行う必要があります。

##### 4.2.3.1. システムの SSM への追加

システムを SSM に追加すると、管理者は、システムがどのシステムグループに属しているかに関わらず、システムの特定グループへの更新、設定の変更などを管理することができます。システムを SSM に追加するには、以下を実行します。

1. システム → システムの順にクリックします。
2. システムの名前をクリックします。
3. ページ右上にある **ssm へ追加** をクリックします。

これにより、システムが SSM 内のシステム一覧に追加されるはずです。

##### 4.2.3.2. SSM 内でのエラータ更新のスケジュールリング

SSM 内の作業グループでシステムのエラータ更新をスケジュールするには:

1. システム → システム設定マネージャー をクリックします。
2. メインページで **エラータの更新をスケジュール** をクリックするか、または **エラータ** サブタブをクリックします。
3. 影響を受けるシステムに適用する **アドバイザリ** を選択します。システムのセットに有効な分だけ選択します。
4. エラータの**適用** をクリックします。



#### 4.2.3.3. チャンネルメンバーシップの管理

チャンネル管理者は、システムがサブスクライブさせているベースチャンネルを変更できます。有効なチャンネルは、組織によって作成されたチャンネルか、または使用しているオペレーティングシステムのバージョンとプロセッサタイプのデフォルトの **Red Hat** ベースチャンネルのいずれかです。新規のベースチャンネルを選択すると、すべてのチャンネルからのシステムのサブスクライブが中止されます。そのため、すべての子チャンネルを再びサブスクライブし直す必要があります。

チャンネルメンバーシップを変更するには:

1. システム → システム設定マネージャー → チャンネル の順にクリックします。
2. 前述したように、ベースチャンネルは、子チャンネルにサブスクライブする前に選択している必要があります。子チャンネルのサブスクリプションのみを変更する場合は、次の手順に進みます。ベースチャンネルを変更する必要がある場合、以下の手順に従って、ベースチャンネルにサブスクライブします。
  - a. ベースチャンネルサブタブをクリックします。
  - b. 必要なベースチャンネルからサブスクライブするベースチャンネルを選択します。
  - c. サブスクリプションの確認 をクリックします。
  - d. 子チャンネルサブタブをクリックして、「子チャンネル」ページに戻ります。
3. 選択したシステムをチャンネルにサブスクライブさせるには、**サブスクライブ (Subscribe)** を選択します。選択したシステムのチャンネルへのサブスクライブを中止するには、**サブスクライブを中止** を選択し、チャンネルのサブスクリプションに変更を一切加えない場合には**何も実行しない** を選択します。
4. 変更を保存するには、**サブスクリプションの変更** をクリックします。
5. 前の画面で行われた変更を確認するために、変更の要約が表示されます。これらの変更を確認してから、変更が正しい場合は **サブスクリプションの変更** を選択します。

#### 4.2.3.4. SSM による設定管理の有効化

システム内の異なる設定ファイルを管理するには、設定管理を有効にする必要があります。

##### 要件

システム内で設定管理を有効にするには、以下の要件が必要です。

- **Provisioning** エンタイトルメント。**Provisioning** エンタイトルメントをシステムに追加する方法については、『システム』の章を参照してください。
- **Red Hat Satellite Tools** チャンネルへのサブスクリプション。子チャンネルを変更する方法については、『システム』の章を参照してください。
- 組織管理者。
- **SSM** にサブスクライブされたシステム。**SSM** がこの動作を実行するには、システムを **SSM** にサブスクライブさせる必要があります。

1. システム → システム設定マネージャー → 設定 の順にクリックします。
2. 設定を有効にする サブタブをクリックします。

3. **rhcfg-\*** パッケージのパッケージインストーラーをスケジュールします。これらの設定パッケージをインストールする時間を選択します。
4. **Red Hat Satellite 設定管理を有効にする (Enable Red Hat Satellite Configuration Management)** をクリックします。
5. 個々のシステムでターミナルコンソールを開くか、**root** としてリモートからログインします。以下の動作を実行する必要があります。
  - a. このコマンドを実行して、保留中の **rhncfg-\*** パッケージのインストールを完了します。

```
# rhn_check
```

- b. 以下のコマンドを実行して **Red Hat Network** の動作を有効にします。

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

#### 4.2.3.5. SSM を使った設定チャンネルのサブスクリプション

設定チャンネルは、『チャンネル管理』の章でさらに詳しく説明します。以下の手順は、SSM を介してシステムを設定チャンネルにサブスクリプションさせる方法のみを扱います。

##### 要件

SSM を使ってシステムをチャンネルにサブスクリプションさせるには、以下の要件を満たしている必要があります。

- システムを **SSM** に追加している必要があります。本セクションの『システムの **SSM** への追加』の手順を参照してください。
  - 設定管理を有効にしておく必要があります。本章の『SSM を使って設定管理を有効にする』手順を参照してください。
  - 設定管理では、システムが **Provisioning** エンタイトルメントを持っている必要があります。**Provisioning** エンタイトルメントをシステムに追加する方法については、『システム』の章を参照してください。
1. システム → システム設定マネージャー → 設定 または システム → システム設定マネージャー → 設定チャンネルサブスクリプションの順にクリックします。
  2. システムをサブスクリプションさせるチャンネルを選択します。
  3. **続行** をクリックします。
  4. 一覧にあるチャンネルを選択し、上下の矢印を使って順位を変更します。これにより、チャンネルにランクが割り当てられます。チャンネルのランク付けを行うと、順位の高いチャンネルが、順位の低いチャンネル内の同じパスを持つファイルの設定変更をオーバーライドできます。
  5. 端にあるラジオボタンを使用して、設定チャンネルの順位を選択します。これにより、システム上の現在の設定チャンネルに対して、リストされる設定チャンネルのランク付けが行われます。
  6. **サブスクリプションの適用** をクリックします。

7. サブスクリाइブさせる設定チャンネルに対してシステムを確認します。すべての情報を確認した後に **確認** をクリックします。

#### 4.2.3.6. SSM による設定チャンネルの配備

SSM からシステムに関連付けられた変更済みの設定ファイルを配備するには:

1. システム → システム設定マネージャー → **設定** または システム → システム設定マネージャー → **設定チャンネルの配備** をクリックします。
2. 配備するファイルのファイル名を選択します。
3. 設定ファイルの配備をできるだけ早く行うように選択するか、または特定の日時を選択することによって、設定ファイルの配備をスケジュールします。
4. 設定配備を確認するために **ファイルの配備を確認** をクリックします。

#### 4.2.3.7. プロビジョニング用のシステムのタグ付け

システムのタグ付けにより、タグが作成された際のシステムの最新のスナップショットにロールバックできます。システムのタグ付を行うには、以下を実行します。

1. システム → システム設定マネージャー → **タグ** の順にクリックします。
2. タグ名 フィールドに入力します。
3. **現在のスナップショットにタグ付け** をクリックします。

#### 4.2.3.8. SSM を使用したリモートコマンドの実行

SSM を使用してリモートコマンドを実行するには:

1. システム → システム設定マネージャー → **リモートコマンド** をクリックします。
2. 以下のフィールドに入力します。
  - ユーザーとして実行
  - グループとして実行
  - タイムアウト (秒単位)
  - スクリプト
3. シェルスクリプトをターゲットシステムで実行できるようにスケジュールされた日時を設定します。
4. **リモートコマンドのスケジュール** をクリックします。

#### 4.2.4. 動作チェーンの管理

Satellite は、複数の動作を順番に実行する機能を提供します。たとえば、web サーバーを設定するために、**httpd** パッケージをシステムにインストールして **/etc/httpd/conf.d/** の設定ファイルをアップロードし、最後にスクリプトを実行して **httpd** サービスを開始する必要がある場合があります。Satellite では動作チェーンの機能を使用して、これらの動作をスケジュールに組み込み、順番に実行することができます。

動作チェーンは、以下の動作を順番にリンクできます。

- パッケージのインストール
- パッケージの更新
- パッケージの削除
- パッケージの検証
- エラータの適用
- リモートコマンドの実行
- 設定ファイルの配備
- システムの再起動

個別のシステムとシステムのセットの両方に適用されます。

動作を動作チェーンに追加するには、動作を開始して確認ページで**動作チェーンに追加**を選択した後、**確認**をクリックして動作を追加します。



☐ Schedule no sooner than:

☒ Add to Action Chain:

#### 図4.1 動作チェーンの追加

スケジュール→動作チェーンと選択し、動作チェーンのリストを表示します。動作チェーンのラベルをクリックして動作を表示し、順番を編集したり、チェーンから個々の動作を削除します。さらに、スケジュールセクションは、動作のスケジュールを指定するために日付と時間を選択するフィールドを提供します。適切な日付と時間を選択して動作のスケジュールを指定した後に**保存とスケジュール**ボタンをクリックすると、動作チェーンが指定された日付および時間に実行されます。

### 4.3. 仮想化クライアントシステムの管理

クライアントシステムの管理やプロビジョニングを行うためには、Red Hat Network の中央サーバーのコンテンツを Red Hat Satellite に同期させます。

少なくとも次のチャンネルを同期します。

Red Hat Enterprise Linux 5 の場合

- Red Hat Enterprise Linux Server (32 ビット x86 対応の v.5) - rhel-i386-server-5 (およびすべての子チャンネル)
- RHEL サーバー向け Red Hat Network Tools (32 ビット x86 対応の v.5) - rhn-tools-rhel-i386-server-5
- Red Hat Enterprise Linux Server Virtualization (32 ビット x86 対応の v.5) - rhel-i386-server-vt-5 (およびすべての子チャンネル)

## Red Hat Enterprise Linux 6 の場合

- Red Hat Enterprise Linux Server (64 ビット x86\_64 対応の v.6) - rhel-x86\_64-server-6 (およびすべての子チャンネル)
- RHEL サーバー向けの Red Hat Network Tools (64 ビット x86\_64 対応の v.6) - rhn-tools-rhel-x86\_64-server-6

## 4.3.1. 仮想システム用ホストシステムのセットアップ

ゲストシステムの作成前に、まずはホストシステムの準備が必要です。Red Hat Enterprise Linux Server のキックスタープロファイルを作成し、次にそのキックスタープロファイルを使ってオペレーティングシステムをホストにインストールします。これらのステップが完了したら仮想ゲストのプロビジョニングに進みます。

## 4.3.1.1. ゲストシステム用のキックスタープロファイルの作成

1. Satellite の Web インターフェースにログインします。ユーザーの **Red Hat Network (Your Red Hat Network)** にある **タスク ウィジェット** 内の **キックスターの管理 (Manage Kickstarts)** リンクをクリックして **キックスター概要** の画面まで行きます。または **システム** タブをクリックしてから左のナビゲーションバーの **キックスター サブタブ** をクリックしても **キックスター概要** の画面まで行くことができます。
2. **キックスター概要** ページの右上部にある **キックスターのアクション ウィジェット** 内の **新規のキックスタープロファイルを作成** をクリックします。
3.
  - a. 他のプロファイルと区別できるようプロファイルのラベルを入力します。ここでは説明しやすいようラベルを **host-system-for-virtual-guests** にしたと仮定して進めます。
  - b. **ベースチャンネル** フィールドには、Red Hat Enterprise Linux (\$ARCH 対応の v.5 または 6) を選択します (\$ARCH はご使用のホストシステムのアーキテクチャーです)。



## 注記

32 ビットの Red Hat Enterprise Linux 5 または 6 を 64 ビットのホストシステムにインストールすることは可能です。ただし、これを行う場合はゲストシステムも 32 ビットバージョンの Red Hat Enterprise Linux を稼動しなければならないので注意してください。

- c. **キックスター可能なツリー** フィールドでは、**ks-rhel-\$ARCH-server-5 (または 6)** を選択します。\$ARCH はご使用のホストシステムのアーキテクチャーになります。
- d. **仮想化タイプ** を選択します。



## 注記

既存のキックスターの **仮想化タイプ** を変更する場合、ブートローダーやパーティションのオプションも変更してしまう可能性があり、ユーザーの行ったカスタマイズ部分を上書きする可能性があります。**仮想化タイプ** を変更する場合は **パーティション (Partitioning)** タブを見て設定が適切であることを確認してください。

- e. 最後に、画面の右下にある **次へ** ボタンをクリックして次のステップに移動します。



### 注記

上記のオプションがフィールド内がない場合、Red Hat のサーバーから Satellite へのソフトウェアチャンネルのコンテンツの同期が正常に行われていない可能性があります。

4. ホストシステムのインストール用の配信ファイルの場所を選択します。この画面では、**デフォルトのダウンロード場所** がすでに入力されています。**次へ** ボタンをクリックしてステップ 3 に移動します。



### 注記

デフォルトのダウンロード場所がない場合、Red Hat のサーバーから Satellite へのソフトウェアチャンネルのコンテンツ同期が正常に行われていない可能性があります。

5. プロビジョニングを行うホストシステムに設定する root パスワードを選択し、**完了** をクリックしてプロファイルの作成を終了します。
6. 新たに作成されたキックスタートプロファイルが表示されます。プロファイルの各種タブを確認して設定を修正することができますが、ほとんどの場合デフォルト設定で正しく動作するので、修正は特に必要はありません。

Satellite Web インターフェースを使ってゲストをリモートで開始し、停止できるようにするには、**acpid** パッケージを含める必要があります。

#### 4.3.1.2. ホストシステムのキックスタート

新たに作成されたキックスタートプロファイルを使用してホストシステムをキックスタートします。ホストシステムをキックスタートする方法として 3 つのシナリオがあります。以下のシナリオを読んで、最適なシナリオの手順に従ってください。

##### 4.3.1.2.1. ホストシステムに Red Hat Enterprise Linux がインストールされていない

ブート CD を作成してホストシステムでキックスタートを開始します。ホストのプロビジョニングを行うために、先ほど作成したキックスタートプロファイルを使用することができます。次の手順を行うには、使用するマシンに物理的なアクセスが必要なので注意してください。

1. **ssh** を使って Satellite にログインし、ホスト用のブート CD を作成するために ISO を探します。Satellite は次の場所にあります。

```
/var/satellite/rhn/kickstart/ks-rhel-i386-server-5.3/images/boot.iso
```



### 注記

キックスタートするのにフラッシュメモリ USB キーを使ってシステムを起動することができます。これを実行するためのヒントについては、『Red Hat Enterprise Linux インストールガイド』を参照してください。お使いのホストシステムのハードウェアがこれらのデバイスからの起動に対応している必要がありますので注意してください。

2. ドライブにブート CD を挿入してシステムを再起動し、CD-ROM ドライブがシステムの BIOS で1 番目の起動デバイスに設定されていることを確認します。
3. 再起動したら、ブートプロンプトが表示されるはずです。次のコマンドをこのブートプロンプトに入力してキックスターを開始します。

```
linux \
ks=http://your-satellite.example.com/ks/label/the_profile_label_you
created_earlier
```



#### 注記

システムによっては、上記のコマンドに **ksdevice=eth0** を追加する必要がある場合や、キックスタープロセス中の混同を防ぐためシステムの BIOS で1つまたは複数の NIC を無効にする必要がある場合などがあります。

4. ホストシステムのキックスターを開始されます。完了するまでに約15分ほどかかります。このキックスターが正常に完了すると、仮想ゲスト用のホストシステムのプロビジョニングが完了し Satellite に登録されたことになります。

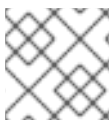
#### 4.3.1.2.2. ホストシステムに Red Hat Enterprise Linux 6 または 7 がインストールされている

ホストシステムを Satellite に登録し、システム上に必要な KVM パッケージがインストールされているかどうかを確認します。インストールされていない場合は、Satellite を使ってインストールします。



#### 注記

Red Hat Enterprise Linux 6 では、仮想化に対応しているのは、64 ビットの Intel および AMD マシンのみです。

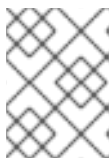


#### 注記

Xen 仮想化ホストは現在、Red Hat Enterprise Linux 6 ではサポートされていません。

1. まず、ホストシステムを Red Hat Satellite に登録します。**ssh** を使ってホストシステムに接続します。その後、**root** で次のコマンドを発行します。

```
# rhnreg_ks --serverUrl=http://your-satellite.example.com/XMLRPC \
--username=username --password=password
```



#### 注記

ホストシステムが別の Red Hat Network サーバーにすでに登録されている場合、上記のコマンドに **--force** オプションを追加します。

2. 次に、Satellite の Web インターフェースでホストシステムのプロファイルを開きます。<https://your-satellite.example.com/> で Satellite の Web インターフェースにログインします。上部のナビゲーションバーにある **システム** タブをクリックします。登録したばかりのホストシステムが表示されるはずです。プロファイル名をクリックしてそのシステムのプロファイルページにアクセスします。

3. 仮想ゲストをホストするために必要となるソフトウェアにアクセスする必要があるため、そのソフトウェアチャンネルへのアクセス権を有していることを確認してください。ホストシステムのプロファイルページから、サブスクライブされているチャンネルヘッダーの **チャンネルサブスクリプションの変更 (Alter Channel Subscriptions)** リンクをクリックします。**RHEL Virtualization** と **RHEL Server 用の Red Hat Network ツール (Red Hat Network Tools for RHEL Server)** のそれぞれのチェックボックスにチェックを入れ、チャンネル一覧の下にある **サブスクリプションの変更** ボタンをクリックします。
4. システムで仮想ゲストをホストするために必要なソフトウェアがインストールされていることを確認します。ホストシステム上で、次のコマンドを **root** として実行します。

Red Hat Enterprise Linux 6 の場合

```
# rpm -q qemu-kvm rhn-virtualization-host python-virtinst
```

Red Hat Enterprise Linux 7 の場合

```
# rpm -q qemu-kvm rhn-virtualization-host virt-install
```

**rpm** がこれらのパッケージはインストールされていないことを示す場合、次のコマンドを **root** としてシステム上で実行してパッケージをインストールする必要があります。

Red Hat Enterprise Linux 6 の場合

```
# yum install qemu-kvm rhn-virtualization-host python-virtinst
```

Red Hat Enterprise Linux 7 の場合

```
# yum install qemu-kvm rhn-virtualization-host virt-install
```

5. マシンを再起動させて変更を反映するか、またはプロセッサに適合した **modprobe** コマンドを使います。

```
# modprobe kvm_intel
```

または

```
# modprobe kvm_amd
```

6. また、**start**、**pause**、**resume**、および **shutdown** などの **Satellite** から送られてくるコマンドに対してホストシステムが応答するようにするためには、**osad** パッケージをインストールして実行する必要があります。インストールするには、以下のコマンドを実行します。

```
# yum install -y osad
```

インストールしたら **osad** プロセスを起動します。

```
# /sbin/service osad restart
```

7. これで **Red Hat Network** 仮想ゲストのプロビジョニングに関するホストシステムの準備が整ったはずです。



### 4.3.1.3. ホストシステムに Red Hat Enterprise Linux 5 がインストールされている

ホストシステムを **Satellite** に登録し、必要な **Xen** または **KVM** パッケージがシステムにインストールされているかどうかを確認します。インストールされていない場合は、**Satellite** を使ってパッケージのインストールを行います。

1. まず、ホストシステムを **Satellite** に登録します。**ssh** を使ってホストシステムに接続します。**root** で、次のコマンドを発行してホストシステムを **Satellite** に登録します。

```
# rhnreg_ks --serverUrl=http://your-satellite.example.com/XMLRPC \
--username=username --password=password
```



#### 注記

ホストシステムが別の **Red Hat Network** サーバーにすでに登録されている場合、上記のコマンドに **--force** オプションを追加します。

2. **Satellite** の Web インターフェースでホストシステムのプロファイルを開きます。<https://your-satellite.example.com/> で **Satellite** の Web インターフェースにログインします。上部のナビゲーションバーにある **システム** タブをクリックします。登録したばかりのホストシステムが表示されます。プロファイル名をクリックしてそのシステムのプロファイルページにアクセスします。
3. 仮想ゲストをホストするために必要となるソフトウェアにアクセスする必要があるため、そのソフトウェアチャンネルへのアクセス権を有していることを確認してください。ホストシステムのプロファイルページから、**サブスクライブされているチャンネル** ヘッダーの **チャンネルサブスクリプションの変更 (Alter Channel Subscriptions)** リンクをクリックします。**RHEL Virtualization** と **RHEL Server 用の Red Hat Network ツール (Red Hat Network Tools for RHEL Server)** のそれぞれのチェックボックスにチェックを入れ、チャンネル一覧の下にある **サブスクリプションの変更** ボタンをクリックします。
4. システムで仮想ゲストをホストするために必要なソフトウェアがインストールされているかどうかを確認します。ホストシステム上で、次のコマンドを **root** として発行します。

```
# rpm -q xen kernel-xen rhn-virtualization-host
```

**KVM** の場合は、次のコマンドを **root** で発行します。

```
# rpm -q kvm kmod-kvm rhn-virtualization-host python-virtinst
```

**rpm** がこれらのパッケージはインストールされていないことを示す場合、次のコマンドを **root** としてシステム上で実行してパッケージをインストールする必要があります。

```
# yum install xen kernel-xen rhn-virtualization-host
```

**KVM** ユーザーの場合は、次のコマンドを **root** で実行してインストールを行います。

```
# yum install kvm kmod-kvm rhn-virtualization-host python-virtinst
```

**Xen** の場合、デフォルトで新しい **xen** カーネルが起動するよう **/etc/grub.conf** 設定ファイルを編集する必要があります。**grub.conf** 内で **xen** カーネルに関連する行となる **title** 行の冒頭から **initrd** 行の末尾までを選択してコピーし、いったん削除してから **grub.conf** 内の

最初のカーネルエントリとなるように貼り付けし直します。また、**grub.conf**の上部にあるデフォルト変数の値が「0」に設定されていることも確認してください。



### 注記

ホストシステム上でカーネルを更新しようとする、再起動時のデフォルト選択は標準カーネルになります。必ず **Xen** カーネルがデフォルトで選択されるようにするには、**/etc/sysconfig/kernel** ファイル内で以下の値を変更します。

```
DEFAULTKERNEL=kernel
```

値を **kernel-xen** に変更します。

```
DEFAULTKERNEL=kernel-xen
```

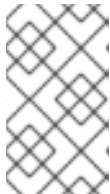
- マシンを再起動させて変更を反映するか、またはプロセッサに適合した **modprobe** コマンドを使います。

```
# modprobe kvm_intel
```

または

```
# modprobe kvm_amd
```

- システムを再起動して、**xen** カーネルで起動します。**uname -r** コマンドを使って実行中のカーネルが **xen** カーネルであるかどうかを確認します。カーネル名に **Xen** の文字列が見当たらない場合は、正しいカーネルで起動していないことになります。



### 注記

システムに **Xen** および **kernel-xen** がすでにインストールされている場合は、**rhn-virtualization-host** のインストール後に再起動を行う必要はありません。

- また、**start**、**pause**、**resume**、および **shutdown** などの **Satellite** から送られてくるコマンドに対してホストシステムが応答するようにするためには、**osad** パッケージをインストールして実行する必要があります。インストールするには、以下のコマンドを実行します。

```
# yum install -y osad
```

インストール後に **osad** プロセスを開始します。

```
# /sbin/service osad restart
```

- これで **Red Hat Network** 仮想ゲストのプロビジョニングに関するホストシステムの準備が整ったはずです。

## 4.3.2. 仮想システムのセットアップ

仮想ゲストシステムで作業を行うには、仮想ゲストを簡単にプロビジョニングできるようにするキックスタートプロファイルを作成します。

#### 4.3.2.1. ゲストシステム用のキックスタートプロファイルの作成

1. Satellite の Web インターフェースにログインします。**概要**にある **タスク ウィジットの キックスタートの管理** リンクをクリックするか、または上部ナビゲーションバーの **システム**、左のナビゲーションバーの **キックスタート** の順にクリックして **キックスタート概要** 画面まで移動します。
2. **キックスタート概要** のページで、右上にある **キックスタートのアクション ウィジットの 新規のキックスタートプロファイルを作成** リンクをクリックします。
3. 次に表示されるページがキックスタートプロファイル作成プロセスのステップ1になります。
  - a. 他のプロファイルと区別しやすいプロファイルラベルを入力します。**guest-system** などがわかりやすいでしょう。
  - b. ベースチャンネル フィールドでは、**Red Hat Enterprise Linux \$PRODUCT (\$ARCH 対応の v.5 または 6)** を選択します。**\$ARCH** はホストシステムのオペレーティングシステムのアーキテクチャーになり、**\$PRODUCT** は **Server** か **Client** のいずれかになります。



#### 注記

Client ソフトウェアチャンネルを **Satellite** に同期していなかった場合、**Red Hat Enterprise Linux Client 5** または **6** は選択できない場合があります。



#### 注記

**Red Hat Enterprise Linux 5** または **Red Hat Enterprise Linux 6** と、**Red Hat Enterprise Linux 5** または **Red Hat Enterprise Linux 6 Desktop** のチャンネルラベルは、それぞれ「**server**」と「**client**」になることに注意してください。

- c. **キックスタート可能なツリー** のフィールドでは、**ks-rhel-\$ARCH-\$PRODUCT-5.3** を選択してください。**\$ARCH** はホストシステムのアーキテクチャーになります。**\$PRODUCT** はゲストをプロビジョニングする製品に応じて「**server**」か「**client**」のいずれかになります。
- d. **仮想化タイプ** を選択します。



#### 注記

既存のキックスタートの **仮想化タイプ** を変更する場合、ブートローダーやパーティションのオプションも変更してしまう可能性があり、ユーザーの行ったカスタマイズ部分を上書きする可能性があります。**仮想化タイプ** を変更する場合は **パーティション (Partitioning)** タブを見て設定が適切であることを確認してください。

- e. 最後に、画面の右下にある **次へ** ボタンをクリックして次のステップに移動します。
4. **キックスタートプロファイル作成プロセスのステップ2** では、ゲストシステムのインストール用の配信ファイルの場所を選択します。すでに **デフォルトのダウンロード場所** が画面上で選択

され、データが記入されているはずです。この画面の **次へ** ボタンをクリックしてステップ 3 に進みます。



#### 注記

Client ソフトウェアチャンネルを **Satellite** に同期していなかった場合、**Red Hat Enterprise Linux Client 5** または **6** は選択できない場合があります。

5. キックスタートプロファイル作成プロセスのステップ 3 では、プロビジョニングしているゲストシステムの **root** パスワードを選択し、**次へ** をクリックしてプロファイルの作成を完了します。

これでキックスタートプロファイルの作成が完了します。ステップ 3 が完了するとプロファイルの詳細に移動します。プロファイルの各種タブをブラウズして設定を変更することができますが、ほとんどの場合、デフォルトの設定で正常に動作するはずなので必ずしも変更する必要はありません。インターフェースは少な目に割り当てられています。このキックスタートプロファイルでのゲストシステムには最低でも **3 GB** の領域を割り当てることを強く推奨します。

#### 4.3.2.2. ゲストシステムのプロビジョニング

1. **Satellite** の **Web** インターフェースにログインします。上部ナビゲーションバーにある **システム** タブをクリックしてホストシステムのプロファイルに移動し、システム名をクリックします。
2. ゲストシステムのキックスタートをスケジュールするには、ホストシステムのプロファイルで **仮想化** → **プロビジョニング** タブの順に進みます。**ゲスト名** フィールドには **guest1** を選択します。**メモリ割り当て**、**仮想 CPU**、および **ストレージ** の各フィールドについては、デフォルトの値で良いでしょう。必要であれば変更を加えても構いません。インターフェースの各フィールドに入力されたアドバイスをメモしておきます。**キックスタートプロファイル** フィールドには前のステップで作成したゲストのシステムプロファイルを選択します。
3. 画面の右下にある **キックスタートをスケジュールしてから終了する** ボタンをクリックすると、ゲストのゲストのキックスタート進捗状況が確認できる **キックスタートの状態** ページに移動します。状態画面はキックスタートが正常に完了したことを表示します。新しいゲストを表示するには、**Satellite** 上のホストシステムのプロファイルの **仮想化** タブをクリックします。仮想システム一覧を表示するには、**システム** → **システム** → **仮想システム** の順に移動します。

## 注記

ゲストのキックスタートをスケジュールしてからしばらく経っても **キックスタートの状態** ページに **キックスタートの開始** というゲストメッセージが表示されない場合、ホストに **osad** がない可能性があります。

ホストシステムには、**start** (開始)、**pause** (停止)、**resume** (再開)、および **shutdown** (シャットダウン)などの **Satellite** から送られてくるコマンドに対して応答できるよう **osad** パッケージが必要です。**osad** がインストールされ稼働されていないと、2.5 時間の間、または次の **Red Hat Network** デーモンが実行されるまで、ホストシステムは **Web** インターフェースからこれらのコマンドを受信しなくなります。

**Satellite** でホストシステムのプロファイルにある **OSA の状態** フィールドを見ると **osad** のインストールおよび実行が行われているかどうかを確認することができます。**OSA の状態** フィールドが存在しないか、またはこのフィールドがシステムが数分間 **Satellite** に接続していないという障害を示している場合、このホスト上で正常にゲストをプロビジョニングする前に、**osad** をインストールする必要があります。**root** として **yum install -y osad** を実行します。

## 注記

ゲストのキックスタート時に、**キックスタートの状態** ページから次のメッセージを受け取る場合があります。

```
The install process on the guest system has not
communicated to Red Hat Network in
the past n minutes. This may be due to a hung install
process, or it
may just be due to a slow install because of hardware
constraints. A
log of the installation process is available, you may
wish to review
it to troubleshoot this issue.
```

このメッセージが表示されてから 20 分以上経過していない場合は、心配せずに待機してください。キックスタートが進行しているかどうかを確認するには、インストールログにエラーがないか確認し、またキックスタートの状態ページを再ロードするときに「最後に要求されたファイル」フィールドの更新が継続されているかどうかを確認します。

4. ホストに追加のゲストを登録したい場合は上記のステップを繰り返します。ただし、一度にプロビジョニングできるのは1ゲストのみになることに注意してください。1ゲストのキックスタートが進行している間に別のゲストのキックスタートをスケジュールしようとすると、現在進行しているゲストのキックスタートプロセスが取り消され、新しいゲストのキックスタートプロセスが開始します。
5. ホストシステムのプロファイルで**仮想化** タブをクリックして、**Satellite** の **Web** インターフェース内に新たに作成した仮想ゲストのシステムを表示させます。次に、仮想システムのプロファイル名をクリックし、その **Satellite** システムのプロファイルを表示します。

### 4.3.2.3. 仮想ゲストのエンタイトルメントの管理

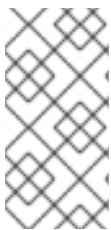
Red Hat Satellite には Flex ゲストエンタイトルメントがあり、これにより、物理システム用に確保された標準エンタイトルメントを使用せずに、仮想ゲストにエンタイトルメントを割り当てることができるようになります。

Flex ゲストのエンタイトルメントを管理する場合は **概要 -> サブスクリプション管理 -> 仮想化エンタイトルメント -> Flex ゲストエンタイトルメントコンシューマー** の順にクリックしていきます。このページには Flex ゲストのエンタイトルメントを使用している全仮想ゲスト一覧が表示されます。

標準のエンタイトルメントを使用している仮想ゲストを検索して変換するには **標準のエンタイトルメントを使用しているゲスト** サブタブをクリックします。

### 4.3.3. 仮想システムでの作業

仮想システムを設定したら、ホストシステム上で SSH や仮想化管理のインターフェースを介した接続などのさまざまな方法で仮想システムの管理およびカスタマイズが可能になります。



#### 注記

このセクションでは主に、Xen ホストについて説明します。Red Hat Enterprise Linux 6 では、Xen は現在サポートされておらず、推奨される仮想化メソッドは KVM となっています。KVM の使用方法について詳しくは、『Red Hat Enterprise Linux 仮想化ガイド』を参照してください。

#### 4.3.3.1. SSH 経由での仮想システムへのログイン

1. 仮想システムの IP アドレスを検索する必要があります。この検索は、システム → **仮想システム** タブの順に移動し、仮想システムのプロファイル名をクリックして行います。
2. 仮想システムプロファイルページ上の、**IP アドレス** フィールドの左側の列に IP アドレスがあります。
3. **root** として **ssh** を使ってこの IP アドレスに接続します。その際、さきほどのキックスタートプロファイル作成で仮想システム用に設定したパスワードを使用します。

#### 4.3.3.2. ホスト経由でのコンソールへのアクセス

1. まずホストシステムに接続してから作業するゲストの ID 番号を確認します。 **ssh** でホストシステムに接続し次のコマンドを実行します。

```
# xm list
```

これにより、Satellite 上で作成した全ゲストの一覧が ID 番号も含めて表示されるはずです。さきほど作成したゲストの **guest1** も ID 番号と共にこの一覧にあります。例えば、このゲストに 2 の ID が割り当てられていた場合、

2. この仮想システムのコンソールにアクセスするには次のコマンドを実行します。

```
# xm console 2
```

即時に **guest1** 上のログインプロンプトが見れるはずです。

3. システムのプロビジョニングに使用したキックスタートプロファイルで設定したのと同じパスワードを使い、**root** として **guest1** にログインします。

(画面に特定のメッセージが表示されるかもしれません。この場合、使用しているキーボードの **Enter** キーを押して新しいログインプロンプトにします。)

4. ゲストコンソールを終了してホストシステムのコマンドプロンプトに戻るには、使用しているキーボードで **Ctrl** キーと **]** キーを同時に押します。

#### 4.3.3.3. Satellite の Web インターフェースを使ったソフトウェアのインストール

1. ログインしてシステム → システム → 仮想システムに移動し、仮想システムのプロファイル名をクリックし、仮想システムのプロファイルをブラウズします。
2. 仮想システムのプロファイルで、ソフトウェア+パッケージ タブをクリックします。
3. パッケージ タブメニュー内の **新しいパッケージのインストール** をクリックします。
4. インストールしたいパッケージを選択して画面下部の右側にある **選択したパッケージをインストール** ボタンをクリックします。
5. パッケージインストールの詳細を確認してから画面下部の右側にある **確認** ボタンをクリックします。
6. パッケージのインストールはゲストシステムが次回 **Satellite** にチェックインしたときに行われます。直ちにインストールの実行を強制するには、ゲストシステムで **rhn\_check** コマンドを実行します。

#### 4.3.3.4. 仮想システムから yum を使用したソフトウェアのインストール

仮想システムはそのゲストのプロビジョニングプロセスの一部として **Satellite** に登録されているので、**yum** コマンドを使用してソフトウェアのインストールおよび更新を行えます。例えば、テキストエディターの **vim** をインストールするには、**root** として次のコマンドを発行します。

```
# yum install -y vim-enhanced
```

#### 4.3.3.5. ホスト再起動時のゲストの再起動

ホストシステムが再起動するときに、デフォルトではゲストは再起動しないため管理者により手動で起動する必要があります。

ただし、**rhn-virtualization-host** サービスはホストシステムの再起動が発生する際に自動的にゲストを再起動することができます。

このサービスを使用するには、以下の手順に従います。

1. ホスト上にあるゲストの設定ファイルを **/etc/sysconfig/rhn/virt/** の中から見つけます。UUID で名前が付けられることになりましたが、正しいファイルを見つけるには **grep** コマンドを使用して UUID ファイル内でゲスト名を検索できます。
2. 使用しているゲストシステムに対応する UUID ファイルを見つけたら、その UUID ファイルから **/etc/sysconfig/rhn/virt/auto/** ディレクトリーにシンボリックリンクを作成します。

```
# ln -s /etc/sysconfig/rhn/virt/GUEST_UUID.xml
/etc/sysconfig/rhn/virt/auto/
```

#### 4.3.3.6. 仮想システムの削除

仮想システムの削除は複数のステップを実行して行います。

1. まず、削除する仮想システムをシャットダウンします。**Satellite Web** インターフェース内のホストシステムのプロファイルをブラウズし、仮想化タブをクリックし、削除する仮想システムを選択するとシャットダウンできます。画面の下部にある **システムのシャットダウン** ボタンをクリックしてシャットダウンを完了します。
2. **Satellite** から仮想システムを削除します。仮想システムのチェックボックスを選択し、画面の下部にある **システムの削除** ボタンをクリックして削除することができます。



#### 注記

仮想システムのシャットダウン操作と削除操作は少なくとも 2 分ほど間隔を置いてから行ってください。間隔が短すぎると仮想システムが正しくシャットダウンせず、この稼働中の仮想システムを削除することになります。仮想システムが稼働している間にこれを削除しても、そのシステムは次のチェックインで **Satellite** 上に再び表示されます。これが発生する場合、単純にシステムをシャットダウンした後に 2 分間待機してから再度削除を実行します。

3. 削除する仮想システムのディスクイメージを削除します。例えば、**guest1** のディスクイメージはホストシステム上の次の場所にあります。

```
/var/lib/xen/disk-images/guest1.disk
```

次のコマンドでこれを削除します。

```
# rm /var/lib/xen/disk-images/guest1.disk
```

**guest1** が KVM ホストシステム上にある場合、ディスクイメージは次の場所にあるはずです。

```
/var/lib/libvirt/images/guest1.img
```

4. 最後に、ホストシステムから **Red Hat Network** 設定ファイルを削除しなければなりません。**guest1** の **Red Hat Network** 設定ファイルを見つけるには、次のコマンドを実行します。

```
# grep guest1 /etc/sysconfig/rhn/virt/*.xml
```

指定されたファイルを削除します。例えば、以下ようになります。

```
# rm /etc/sysconfig/rhn/virt/14e5cfbf72342515236ad74b260c2f6b.xml
```

5. これで、ゲストシステムをホストシステムおよび **Satellite** から正常に削除できました。



## 第5章 エラータ管理

カスタムエラータを使用すると、カスタムチャンネル内のパッケージにエラータアラートを発行し、エラータ配備をスケジュールし、さらに複数の組織間でエラータを管理することができるようになります。



### 警告

Red Hat Satellite Proxy と Red Hat Satellite の両方を使用する場合、Proxy サーバーの役割は Satellite から更新を直接受信することですので、エラータの管理作業は Satellite 上でのみ行なうようにしてください。両方を使用する構成でエラータの管理作業を Proxy で行うと、サーバーが同期しなくなる恐れがあります。

エラータセクションで管理されるエラータには 2 つの種類があります。

1. **発行済みエラータ** - 組織が作成し、配信したエラータアラートを表示します。既存の発行済みエラータを編集するには、「[エラータの作成と編集](#)」に記載されている手順に従います。エラータを配信するには、**エラータの詳細 (Errata Details)** ページの右上にある **通知の送信** をクリックします。エラータアラートは影響を受けるすべてのシステムの管理者に送信されます。
2. **未発行のエラータ** - 組織が作成し、他のエラータアラートでまだ配信していないものが表示されます。既存の未発行のエラータを編集するには、「[エラータの作成と編集](#)」に記載されている手順に従います。エラータを発行するには、**エラータの詳細 (Errata Details)** ページの右上にある **エラータの発行** をクリックします。次にエラータに関連付けられたチャンネルを確認してから右下にある **エラータの発行** ボタンをクリックします。エラータアラートが **発行済み** ページに移動して配信待ちになります。

### 5.1. エラータの作成と編集

以下の手順に従ってカスタムのエラータアラートを作成します。

1. ナビゲーションバーで、**エラータ**をクリックしてから、左側のナビゲーションバーで**エラータの管理**を選択します。**エラータ管理**ページから、**新規エラータの作成**をクリックします。
2. **アドバイザー** フィールドでエラータ用のわかりやすいラベルを入力します。企業や組織内で採用している命名規則などに従うのがよいでしょう。カスタムのエラータと Red Hat から発行されるエラータとを混同しないようラベルの先頭に「RH」を付けないようにしてください(大文字小文字を問わず)。
3. 次に、残りの必須フィールドをすべて入力してから **エラータの作成** ボタンをクリックします。Red Hat の標準的なエラータを表示して適切にフィールド入力されているか確認してください。

Red Hat Satellite 管理者は既存のエラータのクローンを作成することでエラータを作成することもできます。この方法でエラータのクローンを作成するとパッケージの関連付けを維持するためエラータの発行が容易になります。詳細については「[エラータのクローン作成](#)」を参照してください。

既存のエラータアラートの詳細を編集するには、**エラータ管理** ページの**アドバイザー**をクリックして

**詳細** タブの該当フィールド内で変更を加え **エラータ更新** ボタンをクリックします。エラータのチャンネル関連付けを変更する場合は **チャンネル** タブをクリックします。**パッケージ** タブをクリックしてそのパッケージを表示させ変更を行います。

エラータを削除する場合は、**エラータ管理** ページ内のチェックボックスを選択し **エラータの削除** ボタンをクリックしてから削除動作の確認を行います。発行済みのエラータの削除には数分かかることがあります。



### 注記

システムに対してエラータアラートが発行されたときに Email を受信したい場合は、Red Hat Network Management Web サイトで **ユーザーの Red Hat Network => 個人設定** に進み、**電子メールの通知を受け取る** を選択します。サブスクライブさせている各システムの管理者に便利な設定となります。

## 5.2. パッケージのエラータへの割り当て

次の手順に従ってパッケージをエラータに割り当てます。

1. 編集するエラータを選択したら **パッケージ** タブをクリックしてから **追加** サブタブをクリックします。
2. パッケージを編集中的エラータに関連付けるには、関連付けを行いたいパッケージを持っているチャンネルを **表示** ドロップダウンメニューから選択して **表示** をクリックします。編集中的エラータにすでに関連付けされているパッケージは表示されません。**管理している全パッケージ (All managed packages)** を選択すると利用できるパッケージがすべて表示されます。
3. **表示** をクリックすると選択したオプションのパッケージ一覧が表示されます。ページヘッダーはまだ編集中的エラータを表示しているので注意してください。
4. この一覧で編集中的エラータに割り当てるパッケージのチェックボックスを選択し、ページの右下にある **パッケージの追加** をクリックします。
5. パッケージ一覧が記載された確認ページが表示されます。**確認** をクリックしてパッケージをエラータに関連付けます。**管理しているエラータの詳細** ページの **一覧表示/削除** サブタブに新しいパッケージ一覧が表示されます。

エラータへのパッケージの割り当てが完了すると、エラータキャッシュが更新され変更が反映されます。すべての変更が適用されるまでにユーザーがエラータの編集を完了できるようにこの更新には若干のずれがあります。手動でキャッシュの変更を開始するには、ページ上部の **直ちに変更をコミット** の指示に従います。

## 5.3. エラータの発行

パッケージをエラータに追加した後に、影響を受けたシステムに配信するためにエラータを発行する必要があります。エラータを発行するには以下の手順に従います。

1. 上部ナビゲーションバーで、左側のナビゲーションバーにある **エラータ → エラータの管理** をクリックします。
2. **エラータの発行** をクリックします。エラータを使用できるチャンネルを選択するよう指示する確認ページが表示されます。関連するチャンネルを選択します。

3. **エラータの発行** をクリックします。発行されたエラータは、**エラータの管理** の **発行済み** ページに表示されるようになります。

## 5.4. エラータのクローン作成

Red Hat Satellite の一部として、エラータをクローン作成し配信することができます。クローン作成できるのは、いずれかのチャンネルに適用できる可能性があるエラータのみです。エラータを適用しているチャンネルからチャンネルをクローン作成した場合、そのエラータはクローン作成したチャンネルにも適用可能となります。エラータのクローン作成機能を利用するには、上部ナビゲーションバーの **エラータ** をクリックしてから、左側のナビゲーションバーの **エラータのクローン作成** をクリックします。このボタンは Red Hat Satellite をご利用のユーザーにしか表示されません。

**エラータのクローン作成** ページに移動し、**表示** のドロップダウンメニューからエラータを含むチャンネルを選択して **表示** をクリックします。エラータ一覧が表示されたらクローン作成するエラータのチェックボックスを選択して **エラータのクローン作成** をクリックします。エラータの一覧が記載された確認ページが表示されます。**確認** をクリックしてクローンの作成を終了します。

クローンのエラータは未発行のエラータ一覧に表示されます。ここでエラータのテキストや関連付けられているパッケージを確認することができます。準備が整ったらそのエラータを発行して該当するユーザーが利用できるようにします。

## 付録A ブートデバイス

自動インストール(または キックスタート)は効率的なシステムのプロビジョニングに欠かせません。本付録では、クライアントのキックスタートに使用する様々なタイプのブートメディアを作成する方法について説明します。

Red Hat Enterprise Linux CD のブートイメージとなる **boot.iso** はブートデバイスの作成に必須の要件です。このイメージがシステムのどこにあるのかを確認して、その場所を書き留めてください。



### 重要

**syslinux** および **syslinux-extlinux** パッケージをインストールして、以下の手順を実行します。

```
# yum install syslinux syslinux-extlinux
```

**syslinux** パッケージは、Red Hat Enterprise Linux 6 用の **/usr/share/syslinux/** にファイルをインストールします。Red Hat Enterprise Linux 5 を使用している場合、このディレクトリーを **/usr/lib/syslinux/** に置き換えてください。

**syslinux-extlinux** パッケージは、USB ブートメディア作成用のツールをインストールします。

### 手順A.1 CD および DVD ブートメディア



### 注記

以下で使用するバックスラッシュ「\」はコマンドが一行で構成されているためシェルプロンプトでも改行せず一行として入力するという意味です。

1. ブートイメージ用の作業ディレクトリーを作成します。

```
# mkdir -p temp cd/isolinux
```

2. ブートイメージを **temp** ディレクトリーにマウントします。

```
# mount -o loop boot.iso temp
```

3. ブートメディアデバイスに必要なファイルを、上記で作成したディレクトリーにコピーします。

```
# cp -aP temp/isolinux/* cd/isolinux/
```

4. **temp** ディレクトリーをアンマウントし、**cd** ディレクトリーのパーミッションをユーザーに対して読み取りと書き込みに変更します。

```
# umount temp  
# chmod -R u+rw cd
```

5. **./cd** ディレクトリーに移動します。

■

```
# cd ./cd
```

6. **/usr/share/syslinux/menu.c32** ファイルを **./cd** ディレクトリーにコピーします。

```
# cp -p /usr/share/syslinux/menu.c32 isolinux
```

7. 必要に応じて、**CD** による起動用に **isolinux.cfg** 内のブートパラメータとターゲットをカスタマイズします。

8. **mkisofs** を使用して **ISO** を作成し、**CD** または **DVD** に焼き付けます。

```
# mkisofs -o ./custom-boot.iso -b isolinux/isolinux.bin -c
isolinux/boot.cat -no-emul-boot \
    -boot-load-size 4 -boot-info-table -J -l -r -T -v -V "Custom Red
Hat Enterprise Linux Boot" .
```

9. ディレクトリーを **CD** または **DVD** に焼き付けたら手順は終了です。

## 手順A.2 PXE ブート

1. ブートイメージ用の作業ディレクトリーを作成します。

```
# mkdir -p temp pxe/pxelinux.cfg
```

2. ブートイメージを **temp** ディレクトリーにマウントします。

```
# mount -o loop boot.iso temp
```

3. **PXE** ブートデバイスに必要なファイルを、上記で作成したディレクトリーにコピーします。

```
# cp -aP temp/isolinux/* pxe/
```

4. **temp** ディレクトリーをアンマウントし、**cd** ディレクトリーのパーミッションをユーザーに対して読み取りと書き込みに変更します。

```
# umount temp
# chmod -R u+rw pxe
```

5. **/pxe** ディレクトリーに移動します。

```
# cd ./pxe
```

6. **/usr/share/syslinux/menu.c32** ファイルを **/pxe** ディレクトリーにコピーします。

```
# cp -p /usr/share/syslinux/menu.c32 .
```

7. **isolinux.cfg** ファイルを **pxelinux.cfg/default** に移動します。

```
# mv isolinux.cfg pxelinux.cfg/default
```

8. 一時ファイルを削除します。

```
# rm -f isolinux.bin TRANS.TBL
```

9. `/usr/share/syslinux/pxelinux.0` ファイルを `/pxe` ディレクトリーにコピーします。

```
# cp -p /usr/share/syslinux/pxelinux.0 .
```

10. テキストエディタで `pxelinux.cfg/default` ファイルを開いて PXE ブートに必要なブートパラメータやターゲットをカスタマイズします。

### 手順A.3 USB ブートメディア



#### 警告

`root` としてこれらのコマンドを実行する際には (最も重要な部分で必要とされます)、極めて慎重に行ってください。これらのコマンドは、デバイスファイルにアクセスするため、これらのコマンドを正しく使用しないと、システムが回復不可能なダメージを受ける可能性があります。以下の例では、マウントに `/dev/loop0` を使用しています。`losetup -f` のコマンドを使用すると、どれが正しいデバイスかを確認することができます。

1. ブートイメージ用の作業ディレクトリーを作成します。

```
# mkdir -p temp usb/extlinux
```

2. ブートイメージを `temp` ディレクトリーにマウントします。

```
# mount -o loop boot.iso temp
```

3. USB メディアブートデバイスに必要なファイルを、上記で作成したディレクトリーにコピーします。

```
# cp -aP temp/isolinux/* usb/extlinux/
```

4. `temp` ディレクトリーをアンマウントし、`cd` ディレクトリーのパーミッションをユーザーに対して読み取りと書き込みに変更します。

```
# umount temp  
# chmod -R u+rw usb
```

5. `/usr/share/syslinux/menu.c32` ファイルを `./usb/extlinux/` ディレクトリーにコピーします。

```
# cp -p /usr/share/syslinux/menu.c32 ./usb/extlinux/
```

6. **usb/extlinux/isolinux.cfg** ファイルを **usb/extlinux/extlinux.conf** に移動します。

```
# mv usb/extlinux/isolinux.cfg usb/extlinux/extlinux.conf
```

7. 一時ファイルを削除します。

```
# rm -f usb/extlinux/isolinux.bin usb/extlinux/TRANS.TBL
```

8. **custom-boot.img** ファイルを変換して、コピーします。

```
# dd if=/dev/zero of=./custom-boot.img bs=1024 count=3000000
```

9. ループバックデバイスをマウントするための正しい場所を確認します。

```
# losetup -f  
/dev/loop0
```

ブートイメージを使用して、ループバックデバイスを設定します。

```
# losetup /dev/loop0 ./custom-boot.img
```

10. **fdisk** ユーティリティーを開きます。

```
# fdisk /dev/loop0
```

デバイス上にブート可能なプライマリーパーティションを1つ作成します。次のキーの組み合わせを押すと、これを作成できます。 **n p 1 Enter Enter a 1 p w**

11. マスターブートレコード (MBR) をループバックデバイスにコピーします。

```
# dd if=/usr/share/syslinux/mbr.bin of=/dev/loop0
```

12. ループバックデバイスにパーティションマップを追加します。

```
# kpartx -av /dev/loop0
```

13. ファイルシステムを作成します。

```
# mkfs.ext2 -m 0 -L "Custom Red Hat Enterprise Linux Boot"  
/dev/mapper/loop0p1
```

14. デバイスをマウントします。

```
# mount /dev/mapper/loop0p1 temp
```

15. 一時的なファイル群を削除します。

```
# rm -rf temp/lost+found
```

16. **usb/extlinux/** ディレクトリーを一時ディレクトリーにコピーします。

```
# cp -a usb/extlinux/* temp/
```

17. ブートローダーを一時ディレクトリにインストールします。

```
# extlinux -i temp
```

18. 一時ディレクトリーをアンマウントします。

```
# umount temp
```

19. ループバックデバイス上のパーティションマップを削除します。

```
# kpartx -dv /dev/loop0
```

20. ループバックデバイスを削除します。

```
# losetup -d /dev/loop0
```

ファイルシステムの変更を同期します。

```
# sync
```

21. テキストエディタで **extlinux.conf** ファイルを開いて USB ブートに必要なブートパラメータとターゲットをカスタマイズします。

22. イメージを USB デバイスに移動したら手順は完了です。デバイスを挿入して、**dmesg** のコマンドを実行しマウントの場所を確認します。この例では **/dev/sdb** になります。

USB デバイスをアンマウントします。

```
# umount /dev/sdb
```

イメージを USB デバイスにコピーします。

```
# dd if=./custom-boot.img of=/dev/sdb
```



## 付録B カスタムパッケージの管理

本章では Red Hat Network 経由で正しく配信できるパッケージの構築方法の概要を記載しています。RPM を使用する理由、Red Hat Network 用のパッケージの構築方法、およびパッケージの正しい署名法などについて説明します。

### B.1. RED HAT NETWORK のパッケージの構築

Red Hat Network は RPM パッケージマネージャ (RPM) の技術を使用して、各クライアントシステムに適用できる追加ソフトウェアや更新などを確認します。Red Hat Network から取り込まれるパッケージは通常 RPM 形式になりますが、ISO 全体のイメージは Red Hat Network Web サイトの **ソフトウェア** タブで入手可能です。ただし、Red Hat Satellite インストールの場合には利用できません。Satellite サーバーで Solaris のサポートを有効にしている場合は、Red Hat Network Push を使用して Solaris のクライアント群で使用されるカスタムチャンネルに Solaris のパッケージ群をアップロードしてください。

RPM はソフトウェアパッケージのインストール、アンインストール、アップグレードおよび検証などをユーザーが簡単に行うことができるツールです。また、ソフトウェア開発者なら、このツールを使用してプログラムのコンパイル版とそのソースコードをエンドユーザーおよび開発者向けにパッケージ化することもできます。

#### B.1.1. RPM の利点

RPM には次のような利点があります。

##### 簡単なアップグレード

RPM を使用すると、新たに再インストールを行わなくてもシステムの個別コンポーネントをアップグレードすることができます。Red Hat から Red Hat Enterprise Linux の新しいバージョンがリリースされる際には、ユーザーによるアップグレードのための再インストールは必要ありません。RPM によって完全に自動制御されているシステムアップグレードが用意されています。パッケージ内の設定ファイルはアップグレード後も保持されているためカスタムの設定が失われることはありません。パッケージのインストールおよびアップグレードには同じ RPM ファイルが使用されるため、パッケージの更新に特別なアップグレードファイルは必要ありません。

##### パッケージのクエリ

RPM はクエリのオプションを用意していますので、全パッケージまたは特定のファイルの検索を RPM データベース全体に対して行うことができます。また、任意のファイルの所属先またはパッケージの所属先を簡単に見つけることもできます。パッケージに含まれているファイルは圧縮されたアーカイブに入っています。そのカスタムバイナリヘッダーには役に立つパッケージの情報やその内容が含まれています。RPM によりヘッダーのクエリを素早くかつ簡単に行うことができます。

##### システムの検証

パッケージを検証する機能も備わっています。パッケージに関連したファイルが削除されている可能性を懸念している場合、パッケージの検証を行い、パッケージが提供するファイルの状態を確認することができます。この検証ではすべての異常が報告されます。エラーが存在する場合はそのファイルを簡単に再インストールすることができます。修正した設定ファイルは再インストール時にも維持されます。

##### 純粋なソース

RPM の重要な設計目標のひとつは、オリジナルのソフトウェア著者により配布された通りの **純粋な** ソフトウェアソースを使用できるようにすることです。RPM を使用すると、純粋なソースに使用されたパッチや構築方法に関する詳細な解説を付けてパッケージ化することができます。これは複数の理由で重要な利点となります。例えば、プログラムの新しいバージョンがリリースされた場合、

コンパイルを完全に最初から始める必要はなく、パッチを見て何が**必要そうなのか**を判定することができます。ソフトウェアを正常に構築できるようコンパイルされている全てのデフォルト設定や変更を、この技術を使用して簡単に確認することができます。

ソースを純粋に保持することは開発者以外の人にとっては重要には見えないかもしれませんが、純粋なソースの維持は高品質なソフトウェアにつながるためエンドユーザーにとっても重要なものとなります。

### B.1.2. Red Hat Network RPM ガイドライン

RPM の長所は、正確に依存関係を定義し、競合を識別する能力にあります。Red Hat Network は RPM のこうした側面に依存しています。Red Hat Network は自動化した環境を提供することでパッケージインストール中に手動による介入が必要ないようにしています。そのため、Red Hat Network による配信用 RPM を構築する場合には次のルールに従うことが重要となります。

1. RPM についてよく理解しておいてください。パッケージを正しく構築するには、RPM の重要な機能について基本的に理解しておくことが大切になります。RPM に関する詳細については次のリソースをまずご覧ください。
  - [http://docs.fedoraproject.org/en-US/Fedora\\_Draft\\_Documentation/0.1/html/RPM\\_Guide/index.html](http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/RPM_Guide/index.html)
  - [http://docs.fedoraproject.org/en-US/Fedora\\_Draft\\_Documentation/0.1/html/Packagers\\_Guide/index.html](http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/Packagers_Guide/index.html)
  - <http://www.gurulabs.com/GURULABS-RPM-LAB/GURULABS-RPM-GUIDE-v1.0.PDF>
2. 子チャンネル用に RPM を構築する場合は、子チャンネルのベースチャンネルと同じバージョンの Red Hat Enterprise Linux の新規インストール上でそのパッケージを構築します。最初に Red Hat Network からのすべての更新を適用することを忘れないようにしてください。
3. RPM パッケージは **--force** または **--nodeps** のオプションを使用しないでインストールしてください。ビルドシステムで RPM を正常にインストールできない場合、Red Hat Network でもその RPM をシステムに自動的にインストールすることはできません。
4. RPM パッケージのファイル名は **NVR** (名前、バージョン、リリース) 形式にしてください。また、パッケージのアーキテクチャも含ませる必要があります。**name-version-release.arch.rpm** が正しい形式となります。例えば、有効な RPM パッケージのファイル名が **pkgname-0.84-1.i386.rpm** とすると、パッケージ名は **pkgname**、バージョンは **0.84**、リリースは **1**、アーキテクチャは **i386** になります。
5. RPM パッケージはパッケージのメンテナが署名する必要があります。署名のないパッケージも Red Hat Network で配信することができますが、そのパッケージを受信するには **yum** アップデーターに強制的に受取させる必要があります。このため、パッケージへの署名を強く推奨します。パッケージに署名する方法については「[Red Hat Network パッケージ用のデジタル署名](#)」に記載されています。
6. 署名が変更されたり再コンパイルが行われたなど、パッケージが何らかの形で変更された場合にはそのバージョンまたはリリースを増分させる必要があります。つまり、Red Hat Network から配信する各 RPM の **NVRA** (アーキテクチャを含む) は、混乱を避けるため固有のビルドにそれぞれ対応していなければなりません。
7. RPM パッケージがそれ自体を廃止予定にすることはできません。
8. 一つのパッケージを複数のパッケージに分割させる場合は、依存関係に十分に注意してください。どうしても必要な理由がない限り、既存のパッケージを分割するのは避けるようにしてください。

ださい。

9. インタラクティブなプレインストール、ポストインストール、プレアンインストール、またはポストアンインストールなどのスクリプトにパッケージを依存させることはできません。インストール中にユーザーによる直接介入を必要とする場合は **Red Hat Network** で動作させることはできません。
10. プレインストール、ポストインストール、プレアンインストールおよびポストアンインストールなどのいずれのスクリプトにも **stderr** や **stdout** に書き込みを絶対行わせないようにしてください。メッセージが必要なければ **/dev/null** にリダイレクトさせてください。これ以外はファイルに書き込みを行なわせてください。
11. **spec** ファイルを作成する場合は **/usr/share/doc/rpm-<version>/GROUPS** のグループ定義を使用します。完全に一致するものがない場合は 2 番目に適合するものを選択します。
12. **RPM** の依存性機能を利用して、プログラムの実行が必ずインストールの後に行われるようにします。



### 重要

ファイルをアーカイブしてからポストインストールのスクリプトで解凍する手順で **RPM** は作成しないでください。 **RPM** の利点が失われます。

アーカイブ内のファイルがファイル一覧に含まれていないと、競合に関する検証や確認を行うことができません。ほとんどの場合、**RPM** 自体がアーカイブを効率的に圧縮したり展開したりすることができます。たとえば、**%postun** セクションでクリーンアップを行わない **%post** の中にはファイルを作成しないでください。

## B.2. RED HAT NETWORK パッケージ用のデジタル署名

**Red Hat Network** から配信されるパッケージにはすべて デジタル署名がなければなりません。デジタル署名は固有となるプライベートキーで作成され、対応するパブリックキーを使って検証することができます。パッケージを作成したら、**SRPM** (ソース **RPM**) と **RPM** は **GnuPG** キーでデジタルに署名することができます。パッケージのインストールが行われる前に、信頼できる機関によってパッケージが署名されているか、また署名後にパッケージに変更が行われていないことがパブリックキーを使用して検証されます。

### B.2.1. GnuPG キーペアの作成

**GnuPG** キーペアはプライベートキーとパブリックキーで構成されます。以下のようにしてキーペアを生成します。

1. シェルプロンプトで **root** ユーザーになり次のコマンドを入力します。

```
gpg --gen-key
```

**GPG** キーペアは **root** ユーザー以外のユーザーが作成するべきではありません。 **root** ユーザーは、**root** ユーザー以外のユーザーとは異なりメモリページをロックできるため、情報がディスクに書き込まれることはありません。

2. キーペア生成のコマンドを実行すると、次のようなキーペア生成に関するオプション選択を求める画面が表示されます。

```
gpg (GnuPG) 2.0.14; Copyright (C) 2009 Free Software Foundation,
```

Inc.  
This is free software: you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:

- (1) RSA and RSA (default)
- (2) DSA and Elgamal
- (3) DSA (sign only)
- (4) RSA (sign only)

Your selection?

3. オプション **1** を選択し、**Enter** を押します。
4. 次にキーの長さを指定するキーサイズを選択します。キーが長いほどユーザーのメッセージに対する攻撃への対抗力が増します。少なくとも **2048** ビットの長さのキーを作成することを推奨します。
5. 次のオプションはキーの有効期間を指定するように求めます。有効期限の日付を選択する場合には、そのパブリックキーを使用するユーザーにも有効期限の日付を知らせ、新しいパブリックキーを渡さなければならぬので注意してください。期限は設定しないことを推奨します。期限を選択しない場合、その選択についての確認することが求められます。

Key does not expire at all Is this correct (y/n)?

6. **y** を押して決定を確認します。
7. 次にユーザー名、**Email** アドレス、オプションのコメントなどを含むユーザー ID を入力します。これらは別々に入力が求められます。終了すると入力した情報の要約が表示されます。
8. 選択を承認したらパスフレーズを入力します。



注記

アカウントのパスワードと同様、**GnuPG**でも最善のセキュリティー対策には適切なパスフレーズが欠かせません。パスフレーズは大文字、小文字、数字を混ぜて句読点も入れると良いでしょう。

9. パスフレーズの入力と確認を行うとキーが生成されます。次のようなメッセージが表示されます。

We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number generator a better chance to gain enough entropy.

+++++.+++++.+++++...+++++.+++++.+++++.+++++.+++.  
+++++.+++++.+++++.+++++.+++++.+++++.+++++.+++++.+

画面上の生成動作が完了すると、**root** のホームディレクトリー内の **.gnupg** ディレクトリーに新しいキーが配置されます。これは、**root** ユーザーでキーペアを生成した場合にキーが配置されるデフォルトの場所になります。

root のキーを表示させるには、次のコマンドを使用します。

```
gpg --list-keys
```

出力は以下のようになります。

```
gpg: key D97D1329 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   3  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 3u
gpg: next trustdb check due at 2013-08-28
pub   2048D/D97D1329 2013-08-27 [expires: 2013-08-28]
       Key fingerprint = 29C7 2D2A 5F9B 7FF7 6411  A9E7 DE3E 5D0F D97D 1329
uid           Your Name<you@example.com>
sub   2048g/0BE0820D 2013-08-27 [expires: 2013-08-28]
```

パブリックキーを検索する場合は、次のコマンドを発行します。

```
gpg --export -a 'Your Name' > public_key.txt
```

パブリックキーが **public\_key.txt** ファイル内に書き込まれます。

このパブリックキーは非常に重要となります。**yum** でカスタムのソフトウェアを受信する全クライアントシステムに配備しなければならないキーになります。このキーを組織全体に配備する手法については『Red Hat Network クライアント設定ガイド』で扱われます。

### B.2.2. パッケージの署名

パッケージに署名を行なう前に、まず **~/.rpmmacros** ファイルに次の行を含ませます。

```
%_signature gpg
%_gpg_name B7085C8A
```

**\_gpg\_name** キー ID 値の **B7085C8A** は、パッケージの署名に使用する GPG キーリングのキー ID に置き換えます。この値で **RPM** にどの署名を使用するのか指示します。

パッケージ **package-name-1.0-1.noarch.rpm** に署名を行なうには、次のコマンドを使用します。

```
rpm --resign package-name-1.0-1.noarch.rpm
```

パスフレーズを入力します。パッケージが署名されていることを確認するには次のコマンドを使用します。

```
rpm --checksig -v package-name-1.0-1.noarch.rpm
```



#### 注記

**rpm --checksig -v** コマンドを実行する前に、GPG キーをインポートします。さらに詳しくは、次のセクションの「[カスタム GPG キーのインポート](#)」を参照してください。

出力に **Good signature from "Your Name"** のフレーズが表示されるはずです。 *Your Name* は署名キーに関連付けした名前になります。

### B.2.3. カスタム GPG キーのインポート

独自の RPM を安全にビルドして配付する予定の場合、カスタムの RPM はすべて **GNU Privacy Guard (GPG)** を使用して署名することを強くお勧めします。GPG キーを生成して GPG 署名のパッケージをビルドする方法については、「[GnuPG キーペアの作成](#)」に記載されています。

パッケージに署名したら、これらの RPM をインポートする全システムにパブリックキーを配備する必要があります。この作業は 2 つのステップに分けられます。まず、各クライアントがキーを取り込めるようパブリックキーの中央となる場所を作成します。次に、各システムのローカルの GPG キーリングにキーを追加します。

最初のステップは一般的であり、**Red Hat Network** のクライアントアプリケーション導入で推奨している Web サイトを利用した方法を使って行うことができます。これを実行するには、Web サーバー上にパブリックディレクトリーを作成して GPG のパブリック署名をその中に置きます。

```
cp /some/path/YOUR-RPM-GPG-KEY /var/www/html/pub/
```

次に **Wget** を使用してクライアントシステムからキーをダウンロードします。

```
wget -O- -q http://your_proxy_or_sat.your_domain.com/pub/YOUR-RPM-GPG-KEY
```

**-O-** オプションは結果を標準出力に送るのに対し、**-q** オプションを使用すると **Wget** を出力なしの **quiet** モードで実行します。*YOUR-RPM-GPG-KEY* 変数を、使用するキーのファイル名に必ず置き換えてください。

キーがクライアントのファイルシステムで使用できるようになったら、ローカルの GPG キーリングにインポートします。インポート方法については、オペレーティングシステムによって異なる場合があります。

Red Hat Enterprise Linux 3 以降の場合は、次のコマンドを使用します。

```
rpm --import /path/to/YOUR-RPM-GPG-KEY
```

GPG キーがクライアントに正常に追加されると、該当キーを使って署名したカスタム RPM の検証が可能になるはずです。



#### 注記

カスタムの RPM とチャンネルを使用する場合は、それらのパッケージに対してカスタムの GPG キーを必ず作成してください。GPG キーの配置場所についても、キックスタートプロファイルに追加する必要があります。

カスタムの GPG キーはクライアントシステムに追加する必要があり、これが行なわれないとキックスタートインストールに失敗する可能性があります。

## 付録C 改訂履歴

<b>改訂 1.1-0.3</b> 翻訳ファイルを XML ソースバージョン 1.1-0 と同期	<b>Tue Oct 10 2017</b>	<b>Terry Chuang</b>
<b>改訂 1.1-0.2</b> 翻訳ファイルを XML ソースバージョン 1.1-0 と同期	<b>Tue Oct 10 2017</b>	<b>Terry Chuang</b>
<b>改訂 1.1-0.1</b> 翻訳ファイルを XML ソースバージョン 1.1-0 と同期	<b>Tue Sep 12 2017</b>	<b>Terry Chuang</b>
<b>改訂 1.1-0</b> Red Hat Satellite 5.8 リリース向けの初版。	<b>Wed Feb 1 2017</b>	<b>Satellite Documentation Team</b>