Red Hat Network 5.0.0

クライアント設定ガイド

Red Hat Network

Last Updated: 2022-07-31

Red Hat Network 5.0.0 クライアント設定ガイド

Red Hat Network

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

RHN クライアント設定ガイドへようこそ。

目次

前書き

1. ドキュメント規則

本ガイドでは、いくつかの規則を使用して特定の単語やフレーズを強調表示し、特定の情報への注意を促しています。

1.1. 表記規則

特定の単語や句への注意を促すために 4 つの表記慣習を使用しています。これらの規則や、これらが適用される状況は以下のとおりです。

等幅ボールド

シェルコマンド、ファイル名、パスなど、システム入力を強調表示するために使用されます。キーとキーの組み合わせを強調表示するためにも使用されます。以下に例を示します。

> 現在の作業ディレクトリーのファイル **my_next_bestselling_novel** の内容を表示するには、シェルプロンプトで **cat my_next_bestselling_novel** コマンドを入力し、**Enter** を押してコマンドを実行します。

上記には、ファイル名、シェルコマンドおよびキーが含まれます。これはすべて等幅ボールドで表示され、コンテキストにより区別可能なものになります。

キーの組み合わせは、キーの組み合わせの各パーツをつなげるプラス記号によって個別のキーと区別できます。以下に例を示します。

> **Enter** を押してコマンドを実行します。
>
> **Ctrl**+**Alt**+**F2** を押して、仮想ターミナルに切り替えます。

最初の例では、押す特定のキーを強調表示しています。2 つ目の例は、同時に押す 3 つのキーのセットというキーの組み合わせを強調表示しています。

ソースコードの場合、段落内で記述されるクラス名、メソッド、関数、変数名、および戻り値は、上記のように **等幅ボールド** で示されます。以下に例を示します。

> ファイル関連のクラスには、ファイルシステムの **filesystem**、ファイルの **file**、ディレクトリーの **dir** が含まれます。各クラスには、独自の関連付けられたパーミッションセットがあります。

プロポーショナルボールド

これは、アプリケーション名、ダイアログボックステキスト、ラベルが付いたボタン、チェックボックスおよびラジオボタン、メニュータイトルおよびサブメニュータイトルなど、システムで発生した単語またはフレーズを示します。以下に例を示します。

> メインメニューバーから System → Preferences → Mouse を選択し、Mouse Preferences を起動します。**Buttons** タブで、**Left-handed mouse** チェックボックスを選択し、**Close** をクリックしてメインのマウスボタンを左から右に切り替えます (マウスを左手で使い易くします)。
>
> 特殊文字を gedit ファイルに挿入するには、メインメニューバーから Applications → Accessories → Character Map を選択します。次に、Character Map メニューバーから Search → Find... を選択し、**Search** フィールドに文字の名前を入力して **Next** をクリックします。目的の文字が **Character Table** で強調表示されます。この強調表示し

た文字をダブルクリックして **Text to copy** フィールドに配置し、**Copy** ボタンをクリックします。ここでドキュメントに戻り、**gedit** メニューバーから **Edit → Paste** を選択します。

上記のテキストにはアプリケーション名、システム全体のメニュー名および項目、アプリケーション固有のメニュー名、GUI インターフェース内のボタンおよびテキストなどがあります。すべては proportional bold で示され、コンテキストと区別できます。

等幅ボールドイタリック または ***プロポーショナルボールドイタリック***

等幅ボールドまたはプロポーショナルボールドのいずれでも、イタリックが追加されると、置換または変数テキストを意味します。イタリックは、状況に応じて変化するテキストや、文字を入力しないテキストを表します。以下に例を示します。

ssh を使用してリモートマシンに接続するには、シェルプロンプトで **ssh *username@domain.name*** を入力します。リモートマシンが **example.com** で、そのマシン上でのユーザー名が john の場合は、**ssh john@example.com** と入力します。

mount -o remount *file-system* コマンドにより、指定したファイルシステムが再マウントされます。たとえば、**/home** ファイルシステムを再マウントする場合、コマンドは **mount -o remount /home** となります。

現在インストールされているパッケージのバージョンを表示するには、**rpm -q *package*** コマンドを使用します。これにより、***package-version-release*** のような結果が返されます。

上記の太字のイタリック体の用語、username、domain.name、file-system、package、version、および release に注意してください。各単語はプレースホルダーで、コマンドの発行時に入力するテキストまたはシステムによって表示されるテキストのどちらかになります。

作業のタイトルを示す標準的な使用法のほかに、イタリックは新用語と重要な用語の最初の使用を示します。以下に例を示します。

Publican は *DocBook* 公開システムです。

1.2. 引用規則

端末の出力およびソースコードの一覧は、周りのテキストから視覚的に表示されます。

ターミナルに送信される出力は **等幅ローマン** に設定され、以下のように表示されます。

```
books       Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

ソースコードの一覧も **等幅ローマン** に設定されますが、以下のように構文の強調表示が追加されます。

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                 struct kvm_assigned_pci_dev *assigned_dev)
{
        int r = 0;
        struct kvm_assigned_dev_kernel *match;

        mutex_lock(&kvm->lock);

        match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
```

```
                                assigned_dev->assigned_dev_id);
        if (!match) {
                printk(KERN_INFO "%s: device hasn't been assigned before, "
                  "so cannot be deassigned\n", __func__);
                r = -EINVAL;
                goto out;
        }

        kvm_deassign_device(kvm, match);

        kvm_free_assigned_device(kvm, match);

out:
        mutex_unlock(&kvm->lock);
        return r;
}
```

1.3. 注記および警告

最後に、3 つの視覚的スタイルを使用して、見落とす可能性のある情報に注意を促します。

注記

注記とは、タスクへのヒント、ショートカット、または代替アプローチです。注意を無視しても悪い結果を招くことはありませんが、便利なヒントを見逃してしまう可能性があります。

重要

見落としやすい詳細のある重要なボックス: 現行セッションにのみ適用される設定変更や、更新を適用する前に再起動が必要なサービスなどです。「Important」というラベルが付いたボックスを無視しても、データが失われることはありませんが、スムーズな操作が行えないことがあります。

警告

警告は無視すべきではありません。警告を無視すると、データが失われる可能性があります。

第1章 はじめに

本ガイドは、RHN Satellite Server および RHN Proxy Server をお使いのお客様がクライアントシステムをより簡単に設定できるようにすることを目的としています。

デフォルトでは、すべての Red Hat Network クライアントアプリケーションは、中央の Red Hat Network Server と通信するように設定されています。クライアントを Red Hat Network Server ではなく RHN Satellite Server または RHN Proxy Server に接続する場合は、これらの設定の多くを変更する必要があります。1 つまたは 2 つのシステムのクライアント設定を変更するのは、比較的簡単かもしれません。数百または数千のシステムが含まれる大規模なエンタープライズ環境では、ここで説明する大規模再設定手順によりメリットが得られると考えられます。

これを実施するのは複雑であるため、お客様は、Satellite または Proxy サーバーにアクセスするのに必要なタスクの多くを自動化する、事前入力済みのスクリプトを利用することができます。詳細は、「5章*RHN Bootstrap の使用*」を参照してください。Red Hat は、これらの変更の意図を理解することが有用であると考えています。したがって、始めの章で再設定の手動手順について説明します。お客様の組織にとって最適なソリューションを決定する際には、最善の判断を下してください。

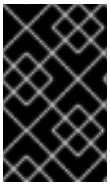
本ガイドで説明するコマンドの多くは、表示されたとおりに適用できますが、お客様によって採用されるすべてのネットワーク設定の可能性を予測することは不可能です。したがって、Red Hat は、これらのコマンドを参考として使用することをお勧めします。実際には、組織の個々の設定を考慮に入れる必要があります。

注記

Unix クライアントの設定情報は、 『RHN 4.0 Reference Guide』の『Unix Support』の章に記載されています。

第2章 クライアントアプリケーション

RHN Satellite への登録など、Red Hat Network のほとんどのエンタープライズクラスの機能を利用するには、最新のクライアントアプリケーションの設定が必要です。クライアントを Red Hat Network に登録する前にこれらのアプリケーションを入手するのは難しい場合があります。このパラドックスは、多数の古いシステムを Red Hat Network に移行する顧客にとって特に問題となります。この章では、このジレンマを解決するための手法を確認します。



重要

Red Hat では、RHN Proxy Server または RHN Satellite Server に接続しているクライアントが Red Hat Enterprise Linux の最新アップデートを実行している状態を強く推奨します。これにより、適切な接続を維持するされます。

2.1. 最新の RED HAT NETWORK クライアント RPM のデプロイ

ご使いの環境で RHN Proxy Server または RHN Satellite Server を使用する前に、それらをクライアントシステムにインストールすることが重要です。

RHN クライアントソフトウェアのこの更新を実施するために、いくつかの重要なアプローチがあります。その 1 つは、すべてのクライアントシステムがアクセスできる場所に RPM を保存し、可能な限り単純なコマンドでパッケージをデプロイすることです。ほとんどすべての場合、**up2date** および **rhn_register** (RHEL 2.1 の場合) を手動でデプロイする必要はありません。これらのクライアントツールは、RHN Satellite または Proxy 環境への接続に関して問題がないはずです。以下の説明では、「そんままの」**up2date** および **rhn_register** が最新でなく、ご使用の環境では機能しないと仮定しています。

Red Hat Enterprise Linux 2.1 を実行しているシステムだけは、RHN への登録に **Red Hat Network Registration Client** を使用しなければならないことを思い出してください。Red Hat Enterprise Linux 3 以降を実行しているシステムでは、**Red Hat Update Agent** に組み込まれている登録機能を使用できます。

このドキュメントは、お客様がネットワークに少なくとも 1 つの RHN Satellite Server または RHN Proxy Server をインストールしていることを前提としています。以下の例は、管理者が初めて **up2date** および **rhn_register** をデプロイする際の簡単な方法を示しています。ここでは、マシンにまだ動作状態の RHN がないと仮定しています。管理者は、**/var/www/html/pub/** ディレクトリーにクライアントシステムに必要な **up2date** および **rhn_register** (RHEL 2.1 システムの場合) RPM のコピーを作成し、続いて単純な **rpm -Uvh** コマンドを使用して、それらの RPM をクライアントシステムにデプロイしています。クライアントからこのコマンドを実行すると、そのクライアントに RPM がインストールされます (ドメイン名、パス、および RPM バージョンが正しいと仮定)。

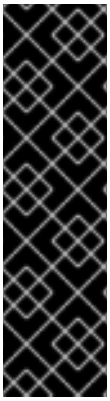
```
rpm -Uvh \
http://your_proxy_or_sat.your_domain.com/pub/rhn_register-2.9.12-1.2.1AS.i386.rpm \
http://your_proxy_or_sat.your_domain.com/pub/rhn_register-gnome-2.9.12-1.2.1AS.i386.rpm \
http://your_proxy_or_sat.your_domain.com/pub/up2date-2.9.14-1.2.1AS.i386.rpm \
http://your_proxy_or_sat.your_domain.com/pub/up2date-gnome-2.9.14-1.2.1AS.i386.rpm
```

関連する gnomeRPM が含まれていることに注意してください。実際のシステムに応じて、アーキテクチャー (ここでは i386) を変更しなければならない場合があることを忘れないでください。

2.2. クライアントアプリケーションの設定

すべてのお客様がセキュアに組織内の RHN Satellite Server または RHN Proxy Server に接続しなければならない訳ではありません。すべてのお客様がカスタムパッケージ用の GPG キーをビルドしてデプ

ロイしなければならない訳ではありません。(これらのトピックについては、共にこの後詳しく説明します。) RHN Satellite Server または RHN Proxy Server を使用するすべてのお客様は、Red Hat Network から RHN Satellite Server または RHN Proxy Server にリダイレクトするように、**Red Hat Update Agent** (**up2date**) および場合によっては **Red Hat Network Registration Client** (**rhn_register**) を再設定する必要があります。



重要

ポートの設定はできませんが、**Red Hat Update Agent** が使用するポートは、 SSL (HTTPS) の場合は 443、非 SSL (HTTP) の場合は 80 であることに注意してください。デフォルトでは、**up2date** は SSL だけを使用します。このため、ユーザーはファイアウォールがポート 443 経由の接続を許可するように設定する必要があります。SSL をバイパスするには、**/etc/sysconfig/rhn/up2date** で **serverURL** のプロトコルを **https** から **http** に変更します。同様に、Red Hat Network Monitoring Daemon を必要とする RHN の監視機能とプローブを使用するには、クライアントシステムがポート 4545 (または **sshd** を使用する場合はポート 22) 経由の接続を許可する必要があることに注意してください。

デフォルトでは、**Red Hat Network Registration Client** および **Red Hat Update Agent** は、メインの Red Hat Network Server を参照します。ユーザーは、RHN Satellite Server または RHN Proxy Server を参照するようにクライアントシステムを再設定する必要があります。

最新バージョンの **Red Hat Update Agent** は、複数の RHN Server に対応するように設定できます。そのため、プライマリーサーバーにアクセスできない場合にフェイルオーバーによる保護が可能です。この機能を有効にする手順は、「サーバーフェイルオーバーの実装」を参照してください。

次のセクションでは、RHN Satellite Server または RHN Proxy Server にアクセスするようにクライアントシステムを設定する方法について説明します。アクティベーションキーを使用する、**up2date --configure** コマンドを使用する、および設定ファイルを手動で更新する、の3 とおりです。(スクリプト化できる再設定項目については、「6章*設定の手動スクリプト化*」を参照してください。)

2.2.1. アクティベーションキーによる登録

Red Hat は、RHN Proxy Server または RHN Satellite Server にアクセスするクライアントシステムの登録および設定には、アクティベーションキーを使用することをお勧めします。アクティベーションキーを使用して、システムの登録、エンタイトルメント、およびサブスクライブをまとめて実施できます。使用方法については、『RHN Management Reference Guide』の**Red Hat Update Agent**の章の「Activation Keys」セクションを参照してください。

アクティベーションキーを使用した登録は、以下の 4 つの基本ステップで実施されます。

1. 『RHN Management Reference Guide』の**Red Hat Update Agent**の章の「Activation Keys」セクションの説明に従って、アクティベーションキーを生成します。

2. カスタム GPG キーをインポートします。

3. RHN Proxy Server または RHN Satellite Server の **/pub/** ディレクトリーから、SSL 証明書 RPM をダウンロードしてインストールします。このステップのコマンドは以下のようになります。

```
rpm -Uvh\
http://your-satellite.com/pub/rhn-org-trusted-ssl-cert-1.0-1.noarch.rpm
```

4. システムを RHN Proxy Server または RHN Satellite Server に登録します。このステップのコマンドは以下のようになります。

```
rhnreg_ks --activationkey mykey --serverUrl https://your-satellite.com/XMLRPC
```

あるいは、上記のステップのほとんどを、以下の行が含まれるシェルスクリプトに組み合わせることができます。

```
wget -0 - http://your-satellite-DQDN/pub/bootstrap.sh | bash \ && rhnreg_ks --activation-key my_key
--serverUrl \ https://your-satellite-FQDN/XMLRPC
```

インストール時に生成され、RHN Satellite Server と RHN Proxy Server の両方で使用可能なブートストラップスクリプトは、そのようなスクリプトです。スクリプトとそれを生成する RHN Bootstrap については、「5章*RHN Bootstrap の使用*」で詳しく説明します。

警告

Red Hat Enterprise Linux 2.1 および 8.0 より前のバージョンの Red Hat Linux を実行しているシステムでは、アクティベーションキーを使用して SSL 証明書設定を **rhn_register** から **up2date** に移行する際に問題が発生する場合があります。したがって、これらのシステムの SSL 証明書情報は手動で設定する必要があります。サーバー URL など、その他のすべての設定は正しく転送されます。

2.2.2. --configure オプションの使用

Red Hat Enterprise Linux に同梱されている **Red Hat Network Registration Client** と **Red Hat Update Agent** は、どちらもさまざまな設定を定義するためのインターフェイスを提供します。これらの設定の完全なリストについては、『RHN Management Reference Guide』でそれぞれのアプリケーションの章を参照してください。

それぞれのアプリケーションには、設定用のグラフィカルユーザーインターフェイス (GUI) が用意されていて、RHN Proxy Server または RHN Satellite Server に必要な設定を変更できます。GUI を使用するには、クライアントシステムが X Window System を実行している必要があります。GUI 設定インターフェイスを起動するコマンドは、以下のようになります。

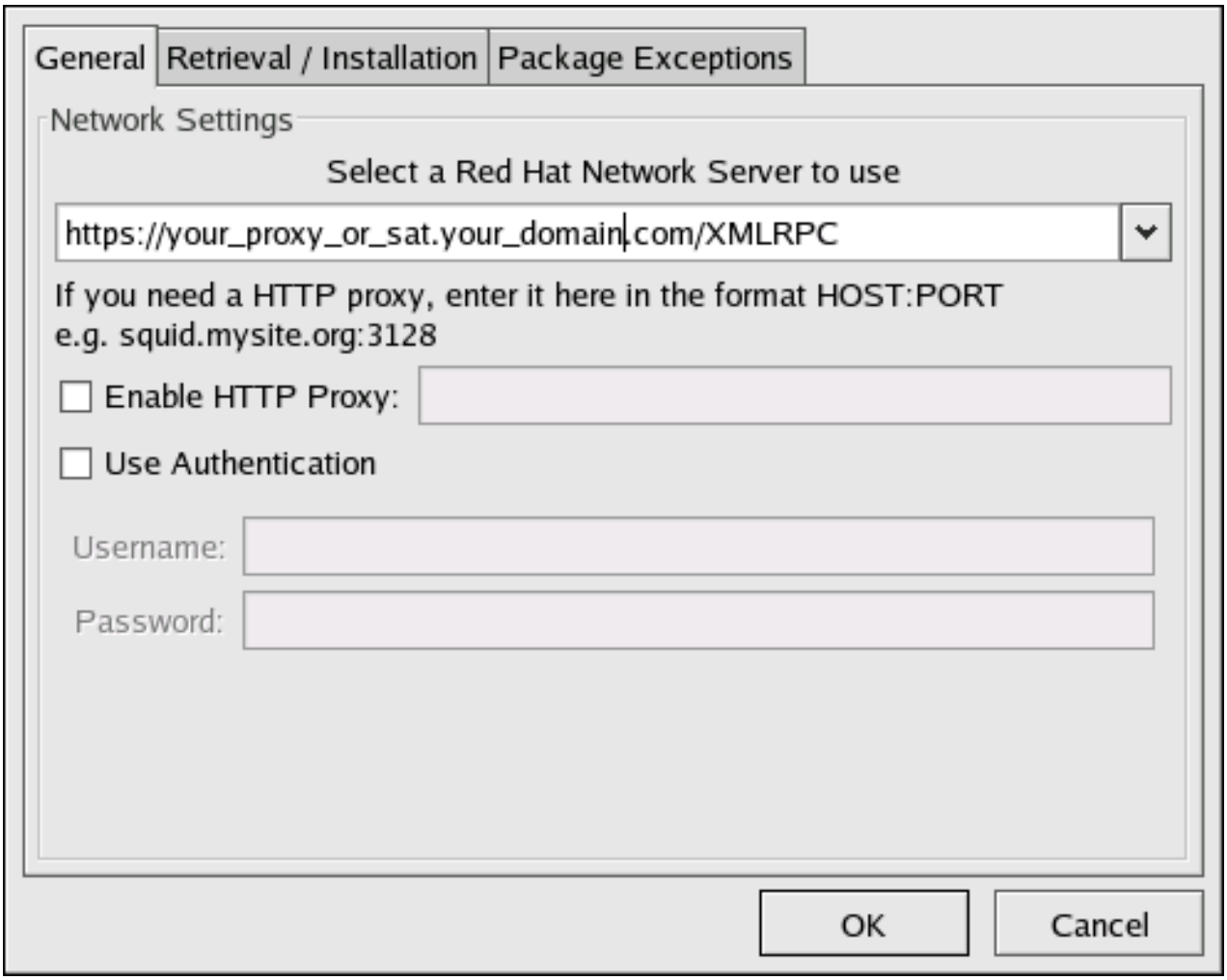
```
application_filename --configure
```

Red Hat Update Agent を再設定するには、root として以下のコマンドを実行します。

```
up2date --configure
```

再設定するさまざまな設定が含まれるダイアログボックスが表示されます。**General** タブの **Select a Red Hat Network Server to use** で、デフォルト値を RHN Satellite Server または RHN Proxy Server の完全修飾ドメイン名 (FQDN) に置き換えます (例: **https://your_proxy_or_sat.your_domain.com/XMLRPC**)。最後の **/XMLRPC** を残します。完了したら、**OK** をクリックします。

図2.1 Red Hat Update Agent GUI の設定



[D]

RHN Satellite Server または RHN Proxy Server のドメイン名を正しく入力するようにしてください。間違ったドメインを入力したり、フィールドを空白のままにしたりすると、**up2date --configure** が起動しない可能性があります。ただし、これは、**up2date** 設定ファイルの値を編集することで解決できます。正確な手順については、「設定ファイルの手動更新」を参照してください。

警告

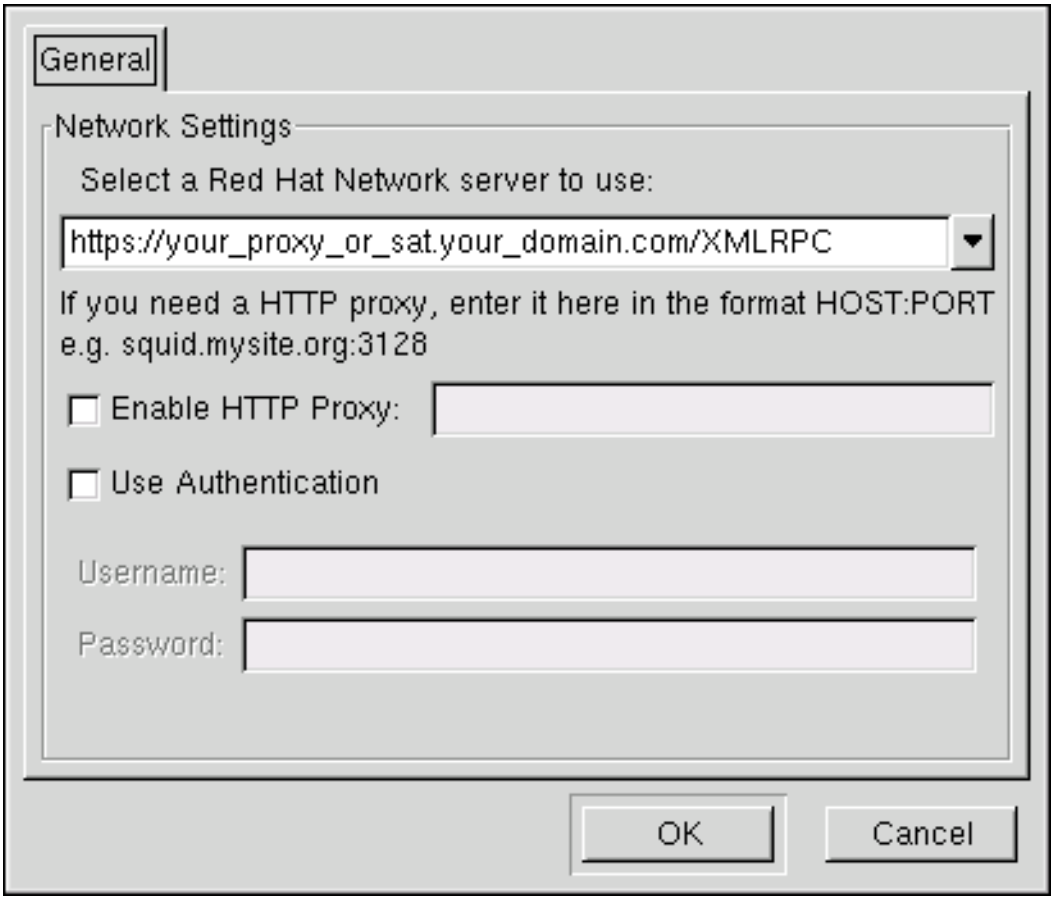
Red Hat Enterprise Linux 3 以降を実行しているシステムの場合、**Red Hat Update Agent** に登録機能が組み込まれているため、**Red Hat Network Registration Client** をインストールしないでください。Red Hat Enterprise Linux 2.1 (および 8.0 より前のバージョンの Red Hat Linux) を実行しているシステムの場合は、**Red Hat Network Registration Client** および **Red Hat Update Agent** を再設定して使用する必要があります。

Red Hat Network Registration Client を再設定するには、ほぼ同じステップのセットを実行します。root で以下のコマンドを実行します。

```
/usr/bin/rhn_register --configure
```

再設定する基本的な設定が含まれるダイアログボックスが表示されます。**Select a Red Hat Network server to use** で、デフォルト値を RHN Satellite Server または RHN Proxy Server の完全修飾ドメイン名 (FQDN) に置き換えます (例: **https://your_proxy_or_sat.your_domain.com/XMLRPC**)。最後の **/XMLRPC** を残します。完了したら、**OK** をクリックします。

図2.2 Red Hat Network Registration Client GUI の設定



[D]

お使いのバージョンの **rhn_register** ではサーバーフィールドが表示されず、それ以降のバージョンにアップグレードできない場合は、RHN Satellite Server または RHN Proxy Server のドメイン名を **rhn_register** 設定ファイルに直接入力できます。正確な手順については、「設定ファイルの手動更新」を参照してください。

2.2.3. 設定ファイルの手動更新

前のセクションで説明した GUI インターフェイスの代わりとして、ユーザーはアプリケーションの設定ファイルを編集して **Red Hat Network Registration Client** および **Red Hat Update Agent** を再設定することもできます。

RHN Proxy Server または RHN Satellite Server に接続しているクライアントシステムで **Red Hat Update Agent** を設定するには、(root として) **/etc/sysconfig/rhn/up2date** 設定ファイルの **serverURL** および **noSSLServerURL** 設定の値を編集します。デフォルトの Red Hat Network の URL を、RHN Proxy Server または RHN Satellite Server の完全修飾ドメイン名 (FQDN) に置き換えます。以下に例を示します。

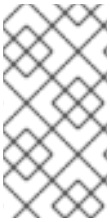
```
serverURL[comment]=Remote server URL
serverURL=https://your_primary.your_domain.com/XMLRPC
```

```
noSSLServerURL[comment]=Remote server URL without SSL
noSSLServerURL=http://your_primary.your_domain.com/XMLRPC
```

警告

/etc/sysconfig/rhn/up2date の **httpProxy** 設定は、RHN Proxy Server を **参照しません**。これは、クライアントのオプションの HTTP プロキシーを設定するのに使用されます。RHN Proxy Server が配置されている場合、**httpProxy** 設定は空白 (いずれの値も設定しない) でなければなりません。

クライアントシステムで Red Hat Enterprise Linux 3 以降を実行している場合は、このセクションをスキップしてください。



注記

クライアントシステムが新しい証明書を認識できるように、クライアントシステムではバージョン 2.7.11 以降の **rhn_register** を使用する必要があります。この RPM は、Proxy の **up2date** を実行した後、プロキシーシステムの **/var/spool/up2date** から利用できます。

RHN Proxy Server または RHN Satellite Server に接続しているクライアントシステムで **Red Hat Network Registration Client** を設定するには、(root として) **/etc/sysconfig/rhn/rhn_register** 設定ファイルの **serverURL** および **noSSLServerURL** オプションの値を編集します。デフォルトの Red Hat Network の URL を、RHN Proxy Server または RHN Satellite Server の完全修飾ドメイン名 (FQDN) に置き換えます。以下に例を示します。

```
serverURL[comment]=Remote server URL
serverURL=https://your_proxy_or_sat.your_domain.com/XMLRPC

noSSLServerURL[comment]=Remote server URL without SSL
noSSLServerURL=http://your_proxy_or_sat.your_domain.com/XMLRPC
```

2.2.4. サーバーフェイルオーバーの実装

up2date-4.2.38 以降、一連の RHN Server からアップデートを探すように **Red Hat Update Agent** を設定できます。これは、プライマリーの RHN Proxy Server または RHN Satellite Server がオフラインになる可能性がある場合に、継続的な更新を維持するのに特に役立ちます。

この機能を使用するには、まず、必要なバージョンの **up2date** を実行するようにしてください。次に、(root として) セカンダリーサーバーを **/etc/sysconfig/rhn/up2date** 設定ファイルの **serverURL** および **noSSLServerURL** 設定に手動で追加します。プライマリーサーバーの直後に、セミコロン (;) で区切って Proxy または Satellite の完全修飾ドメイン名 (FQDN) を追加します。以下に例を示します。

```
serverURL[comment]=Remote server URL
serverURL=https://your_primary.your_domain.com/XMLRPC; \
https://your_secondary.your_domain.com/XMLRPC;
```

```
noSSLServerURL[comment]=Remote server URL without SSL
noSSLServerURL=http://your_primary.your_domain.com/XMLRPC; \
https://your_secondary.your_domain.com/XMLRPC;
```

ここで指定した順序で、サーバーへの接続を試みます。必要な数のサーバーを含めることができます。中央の RHN Server もリストに含めることができます。ただし、これはクライアントシステムがインターネットにアクセスできる場合にのみ意味があります。

2.3. SATELLITE を使用した RED HAT NETWORK ALERT NOTIFICATION TOOL の設定

Red Hat デスクトップのパネルにある丸いアイコンである **Red Hat Network Alert Notification Tool** は、Red Hat Enterprise Linux 3 以降を実行しているシステムにおいて、お使いの RHN Satellite Server のカスタムチャネルから利用可能なアップデートを認識するように設定できます。RHN Satellite Server がこの機能をサポートするように設定されていることを確認する必要があります。(RHN Proxy Server は、クライアントまたはサーバーを変更せずにアプレットをサポートします。) **Red Hat Network Alert Notification Tool** を設定する手順は以下のとおりです。

1. お使いの RHN Satellite Server がバージョン 3.4 以降であること、および Satellite に **rhns-applet** パッケージがインストールされていることを確認します。このパッケージは、バージョン 3.4 以降の RHN Satellite ソフトウェアチャネルから利用可能です。

2. **up2date** を使用するか、Red Hat Network Tools ソフトウェアチャネルを通じて、 **rhn-applet-actions** パッケージを取得します。**Red Hat Network Alert Notification Tool**でカスタムアップデートの通知を受け取るすべての Red Hat Enterprise Linux 3 以降のクライアントシステムに、パッケージをインストールします。クライアントシステムには、Management または Provisioning サービスレベルのエンタイトルメントが必要です。

3. サテライトのバージョンの RHN Web サイト内で、各システムの **System Details** ページに移動し、**RHN Applet** 領域内のリンクをクリックして、**Red Hat Network Alert Notification Tool** を Satellite にリダイレクトします。

次回アプレットを起動すると新しい設定が適用され、RHN Satellite Server に接続してアップデートを探します。

第3章 SSL インフラストラクチャー

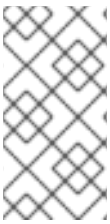
Red Hat Network のお客様にとって、セキュリティー上の懸念が最重要事項です。Red Hat Network の強みの 1 つは、すべての要求を Secure Sockets Layer (SSL) を介して処理できることです。このレベルのセキュリティーを維持するには、インフラストラクチャー内に Red Hat Network をインストールするお客様は、カスタム SSL キーと証明書を生成する必要があります。

SSL キーと証明書の手動作成とデプロイは、非常に複雑になる可能性があります。RHN Proxy Server と RHN Satellite Server では、共にインストール中に独自のプライベート認証局 (CA) に基づいて独自の SSL キーと証明書を作成できます。さらに、この目的のために、別のコマンドラインユーティリティーである **Red Hat Network SSLMaintenanceTool**が存在します。いずれにせよ、これらのキーと証明書は、管理対象インフラストラクチャー内のすべてのシステムにデプロイする必要があります。多くの場合、これらの SSL キーと証明書のデプロイは自動化されています。本章では、これらすべてのタスクを実行するための効率的な方法について説明します。

本章では SSL について詳しく説明しないことに注意してください。**Red Hat Network SSL Maintenance Tool** は、この公開鍵インフラストラクチャー (PKI) のセットアップとメンテナンスに伴う複雑な操作の多くを省くように設計されました。利用できる優れた参考資料が多数あるので、詳細については、これらのいくつかを参照してください。

3.1. SSL の概要

SSL (Secure Sockets Layer) は、クライアント/サーバーアプリケーションが情報を安全に渡すことのできるプロトコルです。SSL は、公開鍵と秘密鍵のペアによるシステムを使用して、クライアントとサーバー間で受け渡しされる通信を暗号化します。公開証明書はアクセス可能なままにしておくことができますが、秘密鍵は保護する必要があります。秘密鍵とそのペアとなる公開証明書の間の数学的な関係 (デジタル署名) により、このシステムが機能します。この関係を通じて、信頼の接続が確立されます。



注記

本ドキュメント全体を通して、SSL 秘密鍵と公開証明書について説明します。技術的には、両方をキー (公開鍵と秘密鍵) と呼ぶことができます。ただし、SSL について説明する場合、SSL キーペア (またはキーセット) のパブリック側を SSL 公開証明書と呼ぶのが慣例です。

組織の SSL インフラストラクチャーは、通常、以下の SSL キーと証明書で構成されます。

- 認証局 (CA) の SSL 秘密鍵と公開証明書: 通常は、1 組織につき 1 セットのみが生成されます。公開証明書は、その秘密鍵によりデジタル署名されます。公開証明書はすべてのシステムに配布されます。

- Web サーバーの SSL 秘密鍵と公開証明書: 1 アプリケーションサーバーにつき 1 セット。公開証明書は、その秘密鍵と CA の SSL 秘密鍵の両方によりデジタル署名されます。多くの場合、Web サーバーのキー *セット* と呼ばれます。これは、生成される中間 SSL 証明書要求があるためです。これが何に使用されるかの詳細は、ここでは重要ではありません。3 つすべてが RHN Server にデプロイされます。

シナリオは次のとおりです。1 つの RHN Satellite Server と 5 つの RHN Proxy Server がある場合に、1 つの CA SSL キーペアと 6 つの Web サーバー SSL キーセットを生成します。CA の SSL 公開証明書はすべてのシステムに配布され、すべてのクライアントがそれぞれのアップストリームサーバーへの接続を確立するのに使用されます。それぞれのサーバーには、厳密にそのサーバーのホスト名に関連付けられ、専用の SSL 秘密鍵と CA の SSL 秘密鍵を組み合わせて使用して生成された独自の SSL キーセット

があります。これにより、Web サーバーの SSL 公開証明書と CA の SSL キーペアおよびサーバーの秘密鍵との間にデジタル検証可能な関連が確立されます。Web サーバーのキーセットを他の Web サーバーと共有することはできません。

重要

このシステムの最も重要な部分は、CA の SSL キーペアです。管理者は、その秘密鍵と公開証明書から、任意の Web サーバーの SSL キーセットを再生成できます。この CA の SSL キーペアを保護する必要があります。サーバーの RHN インフラストラクチャー全体がセットアップされ動作状態になったら、このツールやインストーラーによって生成された SSL ビルドディレクトリーを別のメディアにアーカイブし、CA パスワードを書き留めて、メディアおよびパスワードを安全な場所に保管することを強くお勧めします。

3.2. RED HAT NETWORK SSL MAINTENANCE TOOL

Red Hat Network には、セキュアなインフラストラクチャーの管理を容易にするコマンドラインツール **Red Hat Network SSL Maintenance Tool**(そのコマンドの **rhn-ssl-tool** として一般に知られている) が用意されています。このツールは、**rhns-certs-tools**パッケージの一部として利用できます。このパッケージは、最新の RHN Proxy Server および RHN Satellite Server のソフトウェアチャネル (ならびに RHN Satellite Server ISO) 内にあります。**Red Hat Network SSL Maintenance Tool**を使用すると、専用の認証局 SSL キーペアおよび Web サーバー SSL キーセット (*キーペア*と呼ばれることもあります) を生成できます。

このツールはビルドツールにすぎません。必要なすべての SSL キーと証明書を生成します。また、すべてのクライアントマシンにすばやく配布およびインストールできるように、ファイルを RPM 形式でパッケージ化します。ただし、それらをデプロイすることはありません。その処理は管理者に委ねられるか、あるいは、多くの場合、RHN Satellite Server によって自動的に処理されます。

注記

rhn-ssl-tool が含まれる **rhns-certs-tools** は、最小限の要件で現在の任意の Red Hat Enterprise Linux システムにインストールして実行できます。これは、ワークステーションまたは RHN Server 以外の別のシステムから SSL インフラストラクチャーを管理する管理者にとって便利なものです。

ツールが必要な場合は以下のとおりです。

- CA 公開証明書を更新する場合: これはまれなケースです。
- 最上位のサービス (ホスト型サービス) として中央の RHN Server に接続された RHN Proxy Server バージョン 3.6 以降をインストールする場合: セキュリティー上の理由から、これを CA の SSL キーおよび証明書のリポジトリーにすることはできません。これは、組織のみがアクセス可能とすべきものです。
- 以前は使用していなかった SSL を使用するように RHN インフラストラクチャーを再設定する場合。
- 3.6 より前のバージョンの RHN RHN Proxy Server を RHN インフラストラクチャーに追加する場合。
- 複数の RHN Satellite Server を RHN インフラストラクチャーに追加する場合: これに関する手順については、Red Hat の担当者にご相談ください。

ツールが **不要な場合** は以下のとおりです。

- RHN Satellite Server のインストール中: すべての SSL 設定は、インストールプロセス中に定義されます。SSL キーおよび証明書は、自動的にビルドおよびデプロイされます。

- 最上位のサービスとして RHN Satellite Server バージョン 3.6 以降に接続された RHN Proxy Server バージョン 3.6 以降をインストールする場合: RHN Satellite Server には、RHN Proxy Server の SSL キーおよび証明書の設定、ビルド、およびデプロイに必要なすべての SSL 情報が含まれています。

RHN Satellite Server と RHN Proxy Server 両方のインストール手順により、CA の SSL 公開証明書が各サーバーの **/pub**ディレクトリーにデプロイされるようになります。この公開証明書は、RHN Server に接続するのにクライアントシステムによって使用されます。詳細は、「クライアントへの CA SSL 公開証明書のデプロイ」を参照してください。

つまり、組織の RHN インフラストラクチャーに最上位のサービスとして最新バージョンの RHN Satellite Server をデプロイしている場合、ツールを使用する必要はほとんどありません。そうでない場合は、その使用法をよく理解してください。

3.2.1. SSL の生成について

Red Hat Network SSL Maintenance Toolを使用する主なメリットは、優れたセキュリティー、柔軟性、および移植性です。セキュリティーは、RHN サーバーごとに個別の Web サーバー SSL キーおよび証明書を作成することで達成されます。これらは、すべて組織によって作成された単一の認証局 SSL キーペアによって署名されます。柔軟性は、ツールが **rhns-certs-tools** パッケージがインストールされた任意のマシンで動作できることで得られます。移植性は、どこにでも安全に保管でき、必要に応じてどこにでもインストールできるビルド構造により達成されます。

繰り返しになりますが、インフラストラクチャーの最上位の RHN Server が最新の RHN Satellite Server である場合、必要なのは、**ssl-build** ツリーをアーカイブから **/root** ディレクトリーに復元し、RHN Satellite Server の Web サイト内で提供される設定ツールを利用することです。

Red Hat Network SSL Maintenance Toolを最大限に活用するには、以下に概要を示すタスクをほぼこの順序で実行します。詳細については、残りのセクションを参照してください。

1. 組織内のシステムに **rhns-certs-tools** パッケージをインストールします。通常は RHN Satellite Server または RHN Proxy Server ですが、そうである必要はありません。

2. 組織用に 1 つの認証局 SSL キーペアを作成し、得られる RPM または公開証明書をすべてのクライアントシステムにインストールします。

3. デプロイする Proxy および Satellite ごとに Web サーバー SSL キーセットを作成し、得られる RPM を RHN Server にインストールし、後で **httpd**サービスを再起動します。

   ```
   /sbin/service httpd restart
   ```

4. プライマリービルドディレクトリーおよびすべてのサブディレクトリーならびにファイルで構成される SSL *ビルドツリー*をフロッピーディスクなどのリムーバブルメディアにアーカイブします。(ディスク容量の要件は重要ではありません。)

5. そのアーカイブを確認し、Proxy または Satellite どちらかのインストールガイドの **Additional Requirements** セクションでバックアップ用に説明するように、安全な場所に保管します。

6. 今後の使用に備えて、CA パスワードを記録して安全な場所に保管します。

7. セキュリティー確保のために、ビルドシステムからビルドツリーを削除します。ただし、RHN インフラストラクチャー全体がデプロイされ設定された後に限ります。

8. 追加の Web サーバー SSL キーセットが必要な場合は、**Red Hat Network SSL Maintenance Tool** を実行しているシステム上にビルドツリーを復元し、ステップ 3 から 7 までを繰り返します。

3.2.2. Red Hat Network SSL Maintenance Tool のオプション

Red Hat Network SSL Maintenance Toolには、認証局の SSL キーペアを生成し、サーバーの SSL 証明書およびキーを管理するための多数のコマンドラインオプションが用意されています。このツールは、基本的に **rhn-ssl-tool --help** (一般)、**rhn-ssl-tool --gen-ca --help** (認証局)、および **rhn-ssl-tool --gen-server --help** (Web サーバー) の 3 つのコマンドラインオプションヘルプリストを提供します。rhn-ssl-tool の man ページ (**man rhn-ssl-tool**) も非常に詳細で、役に立ちます。

以下の 2 つの表は、関連タスク (CA または Web サーバーの SSL キーセット生成のいずれか) ごとに、オプションの詳細をまとめています。

-gen-ca 引数に続いて、以下のオプションのセットを指定する必要があります。

表3.1 SSL 認証局 (CA) オプション (rhn-ssl-tool --gen-ca --help)

オプション	説明
--gen-ca	認証局 (CA) キーペアおよび公開 RPM を生成します。このオプションは、この表の残りのオプションのいずれかと共に使用する必要があります。
-h、--help	認証局の生成と管理に固有の基本オプションのリストが含まれるヘルプ画面を表示します。
-f、--force	新しい CA 秘密鍵や公開証明書を強制的に作成します。
-p=、--password=*PASSWORD*	CA パスワード。指定している場合は、これを求めるプロンプトが表示されます。安全な場所に記録、保管してください。
-d=、--dir=*BUILD_DIRECTORY*	**ほとんどのコマンドに必要**: 証明書および RPM がビルドされるディレクトリー。デフォルトは **./ssl-build** です。
--ca-key=*FILENAME*	CA 秘密鍵のファイル名。デフォルトは **RHN-ORG-PRIVATE-SSL-KEY** です。
--ca-cert=*FILENAME*	CA 公開証明書のファイル名。デフォルトは **RHN-ORG-TRUSTED-SSL-CERT** です。
--cert-expiration=*CA_CERT_EXPIRE*	公開 CA 証明書の有効期限。デフォルトは、エポックロールオーバー (2038 年 1 月 18 日) の 1 日前までの日数です。
--set-country=*COUNTRY_CODE*	2 文字の国コード。デフォルトは US です。

オプション	説明
--set-state=*STATE_OR_PROVINCE*	CA の州または地区。デフォルトは '' です。
--set-city=*CITY_OR_LOCALITY*	都市または地域。デフォルトは '' です。
--set-org=*ORGANIZATION*	Red Hat 等の会社または組織。デフォルトは Example Corp 株式会社 です。
--set-org-unit=*SET_ORG_UNIT*	RHN などの組織単位。デフォルトは '' です。
--set-common-name=*HOSTNAME*	**通常 CA には設定されません**: 一般名。
--set-email=*EMAIL*	**通常 CA には設定されません**: メールアドレス。
--rpm-packager=*PACKAGER*	「RHN Admin (rhn-admin@example.com)」など、生成された RPM のパッケージャー。
--rpm-vendor= *VENDOR*	「IS/IT Example Corp」など、生成された RPM のベンダー。
-v、--verbose	詳細なメッセージを表示します。「v」を追加するごとに、詳細度が増します。
--ca-cert-rpm=*CA_CERT_RPM*	**ほとんど変更されない**: CA 証明書が含まれる RPM 名 (filename-version-release.noarch.rpm ではなくベースのファイル名)。
--key-only	**ほとんど使用されない**: CA の秘密鍵だけを生成します。詳細については、**--gen-ca --key-only --help** を確認してください。
--cert-only	**ほとんど使用されない**: CA の公開証明書だけを生成します。詳細については、**--gen-ca --cert-only --help** を確認してください。
--rpm-only	**ほとんど使用されない**: デプロイメント用に RPM だけを生成します。詳細については、**--gen-ca --rpm-only --help** を確認してください。
--no-rpm	**ほとんど使用されない**: RPM 生成を除くすべての CA 関連のステップを実行します。

--gen-server 引数に続いて、以下のオプションのセットを指定する必要があります。

表3.2 SSL Web サーバーオプション (rhn-ssl-tool --gen-server --help)

オプション	説明
--gen-server	Web サーバーの SSL キーセット、RPM、および tar アーカイブを生成します。このオプションは、この表の残りのオプションのいずれかと共に使用する必要があります。
-h、--help	サーバーキーペアの生成と管理に固有の基本オプションのリストが含まれるヘルプ画面を表示します。
-p=、--password=*PASSWORD*	CA パスワード。指定している場合は、これを求めるプロンプトが表示されます。安全な場所に記録、保管してください。
-d=、--dir=*BUILD_DIRECTORY*	**ほとんどのコマンドに必要**: 証明書および RPM がビルドされるディレクトリー。デフォルトは **./ssl-build** です。
--server-key=*FILENAME*	Web サーバーの SSL 秘密鍵のファイル名。デフォルトは **server.key** です。
--server-cert-req=*FILENAME*	Web サーバーの SSL 証明書要求のファイル名。デフォルトは **server.csr** です。
--server-cert=*FILENAME*	Web サーバーの SSL 証明書のファイル名。デフォルトは **server.crt** です。
--startdate=*YYMMDDHHMMSSZ*	サンプル形式のサーバー証明書の有効期間の開始日: 年、月、日、時、分、秒 (値ごとに 2 文字)。Z は Zulu の略で、必須です。デフォルトは、生成の 1 週間前です。
--cert-expiration=*SERVER_CERT_EXPIRE*	サーバー証明書の有効期限。デフォルトは、エポックロールオーバー (2038 年 1 月 18 日) の 1 日前までの日数です。
--set-country=*COUNTRY_CODE*	2 文字の国コード。デフォルトは US です。
--set-state=*STATE_OR_PROVINCE*	州または地区。デフォルトは North Carolina です。
--set-city=*CITY_OR_LOCALITY*	都市または地域。デフォルトは Raleigh です。
--set-org=*ORGANIZATION*	Red Hat 等の会社または組織。デフォルトは Example Corp 株式会社 です。
--set-org-unit=*SET_ORG_UNIT*	RHN などの組織単位。デフォルトは単位です。

オプション	説明
--set-hostname=*HOSTNAME*	キーを受信する RHN Server のホスト名。デフォルトは、ビルドマシンのホスト名に動的に設定されます。
--set-email=*EMAIL*	証明書の連絡先のメールアドレス。デフォルトは admin@example.corp です。
--rpm-packager=*PACKAGER*	「RHN Admin (rhn-admin@example.com)」など、生成された RPM のパッケージャー。
--rpm-vendor=*VENDOR*	「IS/IT Example Corp」など、生成された RPM のベンダー。
-v、--verbose	詳細なメッセージを表示します。「v」を追加するごとに、詳細度が増します。
--key-only	**ほとんど使用されない**: サーバーの秘密鍵だけを生成します。詳細については、**--gen-server --key-only --help** を確認してください。
--cert-req-only	**ほとんど使用されない**: サーバーの証明書要求だけを生成します。詳細については、**--gen-server --cert-req-only --help** を確認してください。
--cert-only	**ほとんど使用されない**: サーバーの証明書だけを生成します。詳細については、**--gen-server --cert-only --help** を確認してください。
--rpm-only	**ほとんど使用されない**: デプロイメント用に RPM だけを生成します。詳細については、**--gen-server --rpm-only --help** を確認してください。
--no-rpm	**ほとんど使用されない**: RPM 生成を除くすべてのサーバー関連のステップを実行します。
--server-rpm=*SERVER_RPM*	**ほとんど変更されない**: Web サーバーの SSL キーセットが含まれる RPM 名 (filename-version-release.noarch.rpm ではなくベースのファイル名)。

オプション	説明
--server-tar=*SERVER_TAR*	**ほとんど変更されない**: ホストされる RHN Proxy Server のインストールルーチンによってのみ使用される、Web サーバーの SSL キーセットおよび CA 公開証明書の .tar アーカイブの名前 (filename-version-release.tar ではなくベースのファイル名)。

3.2.3. 認証局 SSL キーペアの生成

Web サーバーに必要な SSL キーセットを作成する前に、認証局 (CA) の SSL キーペアを生成する必要があります。CA の SSL 公開証明書は、Satellite または Proxy のクライアントシステムに配布されます。**Red Hat Network SSL Maintenance Tool**を使用すると、必要な場合に CA の SSL キーペアを生成し、それを以降のすべての RHN サーバーのデプロイメントに再利用できます。

ビルドプロセスにより、クライアントへの配布用にキーペアおよび公開 RPM が自動的に作成されます。すべての CA コンポーネントは、最終的にコマンドラインで指定されたビルドディレクトリー (通常は **/root/ssl-build**、古いSatellite および Proxy の場合は **/etc/sysconfig/rhn/ssl**) に配置されます。CA の SSL キーペアを生成するには、以下のようなコマンドを実行します。

```
rhn-ssl-tool --gen-ca --password=MY_CA_PASSWORD --dir="/root/ssl-build" \
--set-state="North Carolina" --set-city="Raleigh" --set-org="Example Inc." \
--set-org-unit="SSL CA Unit"
```

例の値を実際の組織に該当する値に置き換えます。コマンドの実行により、指定したビルドディレクトリーに以下の関連ファイルが作成されます。

- **RHN-ORG-PRIVATE-SSL-KEY**: CA の SSL 秘密鍵
- **RHN-ORG-TRUSTED-SSL-CERT**: CA の SSL 公開証明書
- **rhn-org-trusted-ssl-cert-*VER-REL*.noarch.rpm**: クライアントシステムへの配布用に準備された RPM。これには CA の SSL 公開証明書 (上記) が含まれ、それを **/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT** にインストールします。
- **rhn-ca-openssl.cnf**: SSL CA 設定ファイル
- **latest.txt**: 常に関連するファイルの最新バージョンのリストが含まれます。

この手順が完了すると、RPM をクライアントシステムに配布する準備が整います。「クライアントへの CA SSL 公開証明書のデプロイ」を参照してください。

3.2.4. Web サーバーの SSL キーセットの生成

CA の SSL キーペアをすでに生成している必要がありますが、特に複数の Proxy または Satellite がデプロイされている場合は、Web サーバーの SSL キーセットをより頻繁に生成する可能性があります。**--set-hostname** の値は、サーバーごとに異なることに注意してください。つまり、個別の RHN サーバーのホスト名ごとに、SSL キーと証明書の個別のセットを生成してインストールする必要があります。

サーバー証明書のビルドプロセスは、CA SSL キーペアの生成とほとんど同じように行われますが、1 つの例外があります。すべてのサーバーコンポーネントは、最終的にビルドシステムのマシン名を反映したビルドディレクトリーのサブディレクトリーに配置されます (例: **/root/ssl-**

build/MACHINE_NAME)。サーバー証明書を生成するには、以下のようなコマンドを実行します。

```
rhn-ssl-tool --gen-server --password=MY_CA_PASSWORD --dir="/root/ssl-build" \
--set-state="North Carolina" --set-city="Raleigh" --set-org="Example Inc." \
--set-org-unit="IS/IT" --set-email="admin@example.com" \
--set-hostname="rhnbox1.example.com
```

例の値を実際の組織に該当する値に置き換えます。コマンドの実行により、ビルドディレクトリーのマシン固有のサブディレクトリーに以下の関連ファイルが作成されます。

- **server.key**: Web サーバーの SSL 秘密サーバーキー
- **server.csr**: Web サーバーの SSL 証明書要求
- **server.crt**: Web サーバーの SSL 公開証明書
- **rhn-org-httpd-ssl-key-pair-*MACHINE_NAME-VER-REL*.noarch.rpm**: RHN Server への配布用に準備された RPM。関連する src.rpm ファイルも生成されます。この RPM には、上記の 3 つのファイルが含まれています。それらが以下の場所にインストールされます。
  - **/etc/httpd/conf/ssl.key/server.key**
  - **/etc/httpd/conf/ssl.csr/server.csr**
  - **/etc/httpd/conf/ssl.crt/server.crt**
- rhn-server-openssl.cnf: Web サーバーの SSL 設定ファイル
- **latest.txt**: 常に関連するファイルの最新バージョンのリストが含まれます。

この手順が完了すると、RPM をそれぞれの RHN Server に配布してインストールする準備が整います。インストール後に **httpd** サービスを再起動する必要があることに注意してください。

```
/sbin/service httpd restart
```

3.3. クライアントへの CA SSL 公開証明書のデプロイ

RHN Proxy Server と RHN Satellite Server 両方のインストールプロセスでは、CA の SSL 公開証明書および RPM を生成することにより、クライアントのデプロイを比較的容易にしています。これらのインストールプロセスにより、これらの一方または両方のコピーを RHN サーバーの **/var/www/html/pub/**ディレクトリーに配置して公開します。

任意の Web ブラウザーで参照するだけで (***http://proxy-or-sat.example.com*/pub/**)、この公開用ディレクトリーを簡単に調べることができます。

wget または **curl** を使用して、そのディレクトリー内の CA SSL 公開証明書をクライアントシステムにダウンロードできます。以下に例を示します。

```
curl -O http://proxy-or-sat.example.com/pub/RHN-ORG-TRUSTED-SSL-CERT
```

```
wget http://proxy-or-sat.example.com/pub/RHN-ORG-TRUSTED-SSL-CERT
```

あるいは、CA の SSL 公開証明書 RPM が **/pub** ディレクトリーにある場合は、クライアントシステムに直接インストールできます。

```
rpm -Uvh \
http://proxy-or-sat.example.com/pub/rhn-org-trusted-ssl-cert-VER-REL.noarch.rpm
```

これらのコマンドを実行する前に、証明書または RPM の実際の名前を確認してください。

3.4. クライアントシステムの設定

RPM または証明書そのものがクライアントシステムにデプロイされたら、そのシステムの管理者は、新しい CA の SSL 公開証明書ファイルを使用して適切な RHN Proxy Server または RHN Satellite Server に接続するように、**Red Hat Update Agent** および必要であれば **Red Hat Network Registration Client** の設定ファイルを変更する必要があります。その CA SSL 公開証明書用に一般的に受け入れられている場所は、**/usr/share/rhn** ディレクトリーにあります。

RHN Proxy Server と RHN Satellite Server の両方には、デフォルトで **RHN Bootstrap** がインストールされているため、これらの反復ステップが大幅に削減され、クライアントシステムの登録と設定のプロセスが簡素化されます。詳細は、「5章*RHN Bootstrap の使用*」を参照してください。

第4章 カスタム GPG キーのインポート

専用の RPM を安全にビルドおよび配布することを計画しているお客様には、すべてのカスタム RPM を GNU Privacy Guard (GPG) で署名することを強くお勧めします。GPG キーの生成および GPG で署名したパッケージのビルドについては、『Red Hat Network Channel Management Guide 』で説明しています。

パッケージに署名したら、これらの RPM をインポートするすべてのシステムに公開鍵をデプロイ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、クライアントが公開鍵を取得できるように鍵のメイン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、RHN クライアントアプリケーションのデプロイに推奨される Web サイトアプローチを使用して処理できます。(「最新の Red Hat Network クライアント RPM のデプロイ」 を参照してください。) そのためには、Web サーバー上に公開用ディレクトリーを作成し、その中に GPG 公開署名を配置します。

```
cp /some/path/YOUR-RPM-GPG-KEY /var/www/html/pub/
```

クライアントシステムは、**Wget** を使用してキーをダウンロードできます。

```
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信するのに対して、**-q** オプションは **Wget** をクワイエットモードで実行するように設定します。*YOUR-RPM-GPG-KEY* 変数をキーのファイル名に置き換えることを忘れないでください。

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、以下のコマンドを使用します。

```
rpm --import /path/to/YOUR-RPM-GPG-KEY
```

Red Hat Enterprise Linux 2.1 の場合は、以下のコマンドを使用します。

```
gpg $(up2date --gpg-flags) --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。

第5章 RHN BOOTSTRAP の使用

Red Hat Network には、前の章で説明した手動による再設定の多くを自動化するツール **RHN Bootstrap** が用意されています。このツールは、**RHN Satellite Server インストールプログラム** の重要な役割りを果たし、インストール中にブートストラップスクリプトを生成できるようにします。

RHN Proxy Server をお使いのお客様および Satellite 設定を更新しているお客様には、独立して使用できるブートストラップツールが必要です。コマンド **/usr/bin/rhn-bootstrap** で呼び出される **RHN Bootstrap** は、その目的に対応し、RHN Satellite Server と RHN Proxy Server の両方にデフォルトでインストールされます。

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。

- クライアントアプリケーションを RHN Proxy または Satellite にリダイレクトする
- カスタムの GPG キーをインポートする
- SSL 証明書をインストールする
- アクティベーションキーを使用して、システムを RHN および特定のシステムグループとチャネルに登録する
- パッケージの更新、再起動、RHN 設定の変更など、設定後のさまざまな作業を実行する

ただし、スクリプトを使用して設定を行うことに固有のリスクに注意する必要があります。SSL 証明書などのセキュリティーツールは、スクリプト自体によってインストールされます。したがって、それらはまだシステムに存在せず、トランザクションを処理するのに使用することはできません。これにより、誰かが Satellite になりすまして不正なデータを送信する可能性があります。事実上すべての Satellite とクライアントシステムは顧客のファイアウォールの背後で動作し、外部のトラフィックから制限されていることから、この問題の影響は軽減されます。登録は SSL により行われるため、保護されています。

ブートストラップスクリプト **bootstrap.sh** は、RHN Server の **/var/www/html/pub/bootstrap/** ディレクトリーに自動的に配置されます。そこからすべてのクライアントシステムにダウンロードして実行することができます。次のセクションで説明するように、いくつかの準備と生成後の編集が必要であることに注意してください。ツールのオプションの完全なリストについては、「**RHN Bootstrap のオプション**」を参照してください。また、スクリプトの例については、「付録A *サンプルのブートストラップスクリプト*」を参照してください。

5.1. 準備

クライアントシステムを正しく設定するのに、**RHN Bootstrap** (**rhn-bootstrap**) は Red Hat Network インフラストラクチャーの他のコンポーネントに依存します。したがって、スクリプトを生成する前に、これらのコンポーネントの準備を行う必要があります。推奨される初期手順を以下に示します。

- スクリプトによって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を使用して、Red Hat Enterprise Linux システムの登録、RHN サービスレベルへのエンタイトルメント、特定のチャネルおよびシステムグループへのサブスクライブを、すべて 1 つのアクションで実行できます。アクティベーションキーを使用するには、管理エンタイトルメントを利用できる必要があります。一方、一度に複数のアクティベーションキーを含めるには、プロビジョニングエンタイトルメントが必要です。RHN Web サイト (プロキシー用の中央の RHN Server または Satellite の完全修飾ドメイン名) の **Systems** カテゴリー内の **Activation Keys** ページからアクティベーションキーを生成します。キーの作成および使用方法については、『RHN Reference Guide』の「Red Hat Update Agent」および「RHN Website」の章を参照してください。

- Red Hat は、RPM をカスタム GNU プライバシーガード (GPG) キーで署名することをお勧めします。スクリプトから参照できるように、キーを利用可能にします。『RHN Channel Management Guide』の説明に従ってキーを生成し、「4章 *カスタム GPG キーのインポート*」に従って RHN Server の **/var/www/html/pub/** ディレクトリーにキーを配置します。

- スクリプトを使用して CA SSL 公開証明書をデプロイする場合は、証明書またはその証明書が含まれるパッケージ (RPM) をその RHN Server で使用できるようにし、スクリプトの生成時に **--ssl-cert** オプションを指定して証明書を含めます。詳細は、「3章*SSL インフラストラクチャー*」を参照してください。

- 再設定するシステムの種類に応じて、1 つまたは複数のブートストラップスクリプトを開発できるように値を設定します。**RHN Bootstrap** はすべての再設定オプションを提供するため、これを使用して、システムのタイプに応じたさまざまなブートストラップスクリプトを生成できます。たとえば、**bootstrap-web-servers.sh** を使用して Web サーバーを再設定する一方、**bootstrap-app-servers.sh** でアプリケーションサーバーを処理できます。完全なリストについては、「**RHN Bootstrap のオプション**」を参照してください。

5.2. 生成

必要なコンポーネントがすべて揃ったので、**RHNBootstrap** を使用して必要なスクリプトを生成できます。root として RHN Satellite Server または RHN Proxy Server にログインし、**rhn-bootstrap** コマンドに続いて必要なオプションと値を指定します。オプションを指定しない場合、ホスト名、SSL 証明書、SSL および GPG 設定 (存在する場合)、**client-config-overrides.txt** ファイルの呼び出しなど、サーバーから派生した重要な値が含まれる **bootstrap.sh** ファイルが **bootstrap/** サブディレクトリーに作成されます。

Red Hat では、次のように、スクリプトに少なくともアクティベーションキー、GPG キー、および高度な設定オプションを組み込むことを強く推奨します。

- 「準備」で把握したエンタイトルメント要件を考慮して、**--activation-keys** オプションを使用してキーを含めます。

- **--gpg-key** オプションを使用して、スクリプト生成時にキーのパスとファイル名を確認します。それ以外の場合は、**-no-gpg** オプションを使用して、クライアントシステムでこの検証をオフにします。Red Hat は、このセキュリティー対策を維持することをお勧めします。

- **--allow-config-actions** フラグを指定して、スクリプトがアクセスするすべてのクライアントシステムでリモート設定管理を有効にします。この機能は、複数のシステムを同時に再設定する場合に役立ちます。

- **--allow-remote-commands** フラグを指定して、すべてのクライアントシステムでリモートスクリプトを使用できるようにします。設定管理と同様に、この機能は複数のシステムを再設定するのに役立ちます。

完了すると、コマンドは以下のようになります。

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

必ず実際のキー名を指定してください。オプションの完全なリストについては、「**RHN Bootstrap のオプション**」を参照してください。

5.3. スクリプトの使用

最後に、使用するスクリプトの準備が完了したら、いつでも実行することができます。RHN Satellite Server または RHN Proxy Server にログインし、**/var/www/html/pub/bootstrap/** ディレクトリーに移動し、以下のコマンドを実行します。ホスト名とスクリプト名は、システムタイプに合わせて適宜変更してください。

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

安全性の低い代替手段は、**wget** または **curl** を使用して、すべてのクライアントシステムからスクリプトを取得して実行することです。各クライアントマシンにログインし、以下のコマンドを実行します。スクリプト名とホスト名を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
```

あるいは、**curl** を使用する場合は、以下のコマンドを実行します。

```
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 RHN Server を使用するように設定を行なう必要があります。

5.4. RHN BOOTSTRAP のオプション

RHN Bootstrap には、クライアントのブートストラップスクリプトを作成するための多くのコマンドラインオプションが用意されています。これらのオプションの説明は以下の表にありますが、コマンド **rhn-bootstrap --help** を実行するか、その man ページを表示することで、お使いの RHN Server にインストールされているツールのバージョンで確認できます。

表5.1 RHN Bootstrap のオプション

オプション	説明
-h、--help	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。
--activation-keys=*ACTIVATION_KEYS*	RHN Web サイトで定義されるアクティベーションキー。複数エントリーの場合は、スペースなしのコンマで区切ります。
--overrides=*OVERRIDES*	設定オーバーライドのファイル名。デフォルトは client-config-overrides.txt です。
--script=*SCRIPT*	ブートストラップスクリプトのファイル名。デフォルトは bootstrap.sh です。

オプション	説明
--hostname=*HOSTNAME*	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。
--ssl-cert=*SSL_CERT*	組織の公開 SSL 証明書へのパス (パッケージまたは証明書そのもののいずれか)。**--pub-tree** オプションにコピーされます。値が **""** の場合、**--pub-tree** の検索を強制します。
--gpg-key=*GPG_KEY*	組織の公開 GPG キーへのパス (使用されている場合)。**--pub-tree** オプションで指定された場所にコピーされます。
--http-proxy=*HTTP_PROXY*	クライアントシステムの HTTP プロキシー設定 (**hostname:port** の形式)。値が **""** の場合、この設定を無効にします。
--http-proxy-username=*HTTP_PROXY_USERNAME*	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。値が **""** の場合、この設定を無効にします。
--http-proxy-password=*HTTP_PROXY_PASSWORD*	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。
--allow-config-actions	ブール値。このオプションを追加すると、RHN によるすべての設定アクションを許可するようにシステムが設定されます。これには、特定の rhncfg-* パッケージのインストールが必要です (通常は、アクティベーションキーを使用)。
--allow-remote-commands	ブール値。このオプションを追加すると、RHN 経由の任意のリモートコマンドを許可するようにシステムが設定されます。これには、特定の rhncfg-* パッケージのインストールが必要です (通常は、アクティベーションキーを使用)。
--no-ssl	**推奨されません**: ブール値。このオプションを追加すると、クライアントシステムで SSL がオフになります。
--no-gpg	**推奨されません**: ブール値。このオプションを追加すると、クライアントシステムで GPG の確認がオフになります。

オプション	説明
--no-up2date	**推奨されません**: ブール値。このオプションを追加した場合、システムのブートストラップが完了すると、**up2date** が実行されなくなります。
--pub-tree=*PUB_TREE*	**変更は推奨されません**: CA SSL 証明書とパッケージが配置される公開ディレクトリーツリー (ブートストラップのディレクトリーおよびスクリプト)。デフォルトは **/var/www/html/pub/**です。
--force	**推奨されません**: ブール値。このオプションを追加すると、警告があってもブートストラップスクリプトの生成を強制します。
-v、--verbose	詳細なメッセージを表示します。累積的な設定: **-vvv** を指定すると、非常に詳細なメッセージが表示されます。

第6章 設定の手動スクリプト化

本章では、ブートストラップスクリプトを生成する際に、**RHN Bootstrap** を使用する以外の方法を説明します。以下の手順を使用すると、独自のブートストラップスクリプトをゼロから作成できるようになります。

まず始めに考えなければならないことは共通です。つまり、各クライアントで実行される単純なスクリプト可能なコマンドを使用して取得およびインストールできるように、必要なファイルを一元化された場所にデプロイすることです。本章では、これらすべての要素を組み合わせて、組織内の任意のシステムから呼び出すことができる単一のスクリプトを作成する方法について説明します。

前の章のすべてのコマンドを最も適切な順序で組み合わせると、以下のスクリプトが得られます。Red Hat Enterprise Linux 3 以前には **rhn_register** が存在しないことに注意してください。

```
# First, install the latest client RPMs to the system.

rpm -Uvh \
 http://proxy-or-sat.example.com.com/pub/rhn_register-2.8.27-1.7.3.i386.rpm \
 http://proxy-or-sat.example.com.com/pub/rhn_register-gnome-2.8.27-1.7.3.i386.rpm \
 http://proxy-or-sat.example.com.com/pub/up2date-3.0.7-1.i386.rpm \
 http://proxy-or-sat.example.com.com/pub/up2date-gnome-3.0.7-1.i386.rpm


# Second, 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


# Third, install the SSL client certificate for your company's
# RHN Satellite Server or RHN Proxy Server.

rpm -Uvh http://proxy-or-sat.example.com/pub/rhn-org-trusted-ssl-cert-*.noarch.rpm


# Fourth, 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


# Fifth, download the GPG key needed to validate custom packages.

wget -O - -q http://proxy-or-sat.example.com.com/pub/YOUR-RPM-GPG-KEY


# Sixth, import that GPG key to your GPG keyring.

rpm --import /path/to/YOUR-RPM-GPG-KEY
```

6 番目のステップは、Red Hat Linux 3 以降を実行しているシステムに関連するため、ここに記載されていることを忘れないでください。Red Hat Enterprise Linux 2.1 の場合は、以下のコマンドを使用してください。

```
gpg $(up2date --gpg-flags) --import /path/to/YOUR-RPM-GPG-KEY
```

このスクリプトはクリーンで反復可能なプロセスで構成され、RHN Proxy Server または RHN Satellite Server に登録するために準備として、あらゆる Red Hat Network クライアントを完全に設定します。RHN Server の URL、その公開ディレクトリー、実際の GPG キーなどのキーの値を、スクリプト内に記載されているプレースホルダーに挿入する必要があることに注意してください。また、環境によっては、追加の変更が必要になる場合があります。このスクリプトはほぼそのままの状態で機能する可能性がありますが、ガイドとして使用する必要があります。

コンポーネントと同様に、このスクリプトを中央のロケーションに配置できます。このスクリプトをサーバーの **/pub/** ディレクトリーに配置し、そのサーバー上で **wget -O-** を実行し、出力をシェルセッションにパイプで渡すことにより、各クライアントからの 1 つのコマンドで、ブートストラッププロセス全体を実行できます。

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

警告

Web 接続を介してパイプで渡された入力から直接シェルスクリプトを実行すると、明らかにいくつかの固有のセキュリティーリスクが発生します。したがって、この場合、ソースサーバーのセキュリティーを確保することが重要です。

その場合、この 1 行のコマンドをネットワーク上のすべてのシステムで呼び出すことができます。管理者が該当するすべてのシステムに SSH アクセスできる場合、これらのすべてのシステム上で操作を繰り返し、コマンドをリモートで実行するのは簡単な作業です。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と良いでしょう。

第7章 キックスタートの実装

システムの設定に変更を加えるのに最適なタイミングは、そのシステムが最初にビルドされる時です。すでにキックスタートを効果的に使用しているお客様の場合は、そのプロセスにブートストラップスクリプトを追加すると良いでしょう。

すべての設定の問題が解決したら、**up2date** および **rhn_register** RPM に付属の **rhnreg_ks** ユーティリティーを使用して、システムをローカルの Red Hat Network Server に登録することもできます。本章では、システムを登録する際の **rhnreg_ks** の適切な使用方法について説明します。

rhnreg_ksユーティリティーは、**アクティベーションキー** を使用して、システムの指定されたチャネルへの登録、エンタイトルメント、およびサブスクライブを素早く処理します。アクティベーションキーの詳細については、Red Hat Network の『リファレンスガイド』の「Red Hat Update Agent」および「Red Hat Network Web サイト」の章を参照してください。

Red Hat Network を使用してシステムが設定されていく全プロセスを理解する上で、以下のコメント付きキックスタートファイルは理想的な例と言えます。

```
# 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 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 Base. This is quite a minimal set of packages;
# your mileage may vary.

%packages
@ Base
@ Utilities
@ GNOME
```

```
@ Laptop Support
@ Dialup Support
@ Software Development
@ Graphics and Image Manipulation
@ Games and Entertainment
@ Sound and Multimedia Support


# Now for the interesting part.

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

# Remember that nifty one-line command for the bootstrap script that we
# went through? This is an ideal place for it. And 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 the usage of rhnreg_ks, 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 Widgetco "Laptop Software"
# channel. Note that this section applies only to Proxy 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
```

付録A サンプルのブートストラップスクリプト

RHN Satellite Server の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** スクリプトにより、クライアントシステムを再設定し、RHN Server に簡単にアクセスすることができます。この機能は、**RHN Bootstrap** ツールを介して RHN Satellite Server および RHN Proxy Server 両方のお客様が利用できます。特定の用途に合わせてスクリプトを変更した後、各クライアントマシンで実行できます。

詳細については、サンプルおよびシャープ記号 (#) で始まるそのコメントを確認してください。「5章*RHN Bootstrap の使用*」に記載の手順に従って、使用に向けてスクリプトを準備します。

#!/bin/bash echo "RHN Server Client bootstrap script v3.6" # 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 Server via ssh (i.e., from the # RHN 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 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 of the corporate public" echo " GPG key filename (residing in /var/www/html/pub) if appropriate." 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=insert_activation_key_here ORG_GPG_KEY=insert_org_gpg_pub_key_here # can be edited, but probably correct: CLIENT_OVERRIDES=client-config-overrides.txt HOSTNAME=your_rhn_server_host.example.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=0 ALLOW_REMOTE_COMMANDS=0 FULLY_UPDATE_THIS_BOX=1 # # ---------- ---------- # 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=`/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=`/usr/bin/curl -k 2>&1`

```
error=`echo $output | grep "is unknown"` if [ -z "$error" ] ; then FETCH="/usr/bin/curl -SksO" else
FETCH="/usr/bin/curl -SsO" 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 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" rm -f ${ORG_GPG_KEY} $FETCH
${HTTPS_PUB_DIRECTORY}/${ORG_GPG_KEY} # get the major version of up2date res=$(rpm -q -
-queryformat '%{version}' up2date | sed -e 's/\..*//g') if [ $res -eq 2 ] ; then gpg $(up2date --gpg-flags)
--import $ORG_GPG_KEY else rpm --import $ORG_GPG_KEY fi fi echo echo "* attempting to install
corporate public CA cert" if [ $USING_SSL -eq 1 ] ; then if [ $ORG_CA_CERT_IS_RPM_YN -eq 1 ] ;
then rpm -Uvh ${HTTP_PUB_DIRECTORY}/${ORG_CA_CERT} else rm -f ${ORG_CA_CERT}
$FETCH ${HTTP_PUB_DIRECTORY}/${ORG_CA_CERT} mv ${ORG_CA_CERT} /usr/share/rhn/ fi
fi echo echo "REGISTRATION" echo "------------" # Should have created an activation key or keys on
the RHN Server's # website and edited the value of ACTIVATION_KEYS above. # # If you require use
of several different activation keys, copy this file and # change the string as needed. # if [ -z
"$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 [ $REGISTER_THIS_BOX -eq 1 ] ; then echo "* registering" /usr/sbin/rhnreg_ks
--force --activationkey "$ACTIVATION_KEYS" echo echo "*** this system should now be registered,
please verify ***" echo else echo "* explicitely not registering" fi echo echo "OTHER ACTIONS" echo
"-------------------------------------------------------" if [ $FULLY_UPDATE_THIS_BOX -eq 1 ] ; then echo
"up2date up2date; up2date -p; up2date -uf (conditional)" else echo "up2date up2date; up2date -p" 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 up2date itself is updated" fi /usr/sbin/up2date up2date /usr/sbin/up2date -p if [
$FULLY_UPDATE_THIS_BOX -eq 1 ] ; then /usr/sbin/up2date -uf fi echo "-bootstrap complete-"
```

付録B 改訂履歴

改訂 2.0-1.400 Publican 4.0.0 用にリビルド	2013-12-18	Rüdiger Landmann
改訂 2.0-1 ソースから再構成	Fri Nov 1 2013	Zac Dover

索引

シンボル

A

B

C

G

R

S