SUSE Manager

1.7

www.suse.com

August 14, 2012

Client Configuration Guide

Client Configuration Guide

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither Novell, Inc., SUSE LINUX Products GmbH, the authors, nor the translators shall be held liable for possible errors or the consequences thereof.

Contents

About This Guide

SUSE® Manager lets you efficiently manage a set of Linux systems and keep them up-to-date. It provides automated and cost-effective software management, asset management, system provisioning, and monitoring capabilities. SUSE Manager is compatible with Red Hat Network Satellite Server and offers seamless management of both SUSE® Linux Enterprise and Red Hat Enterprise Linux client systems.

This best practices guide is intended to help customers of SUSE Manager configure their client systems more easily.

Many chapters in this manual contain links to additional documentation resources. These include additional documentation that is available on the system as well as documentation available on the Internet.

For an overview of the documentation available for your product and the latest documentation updates, refer to `http://www.novell.com/documentation/suse_manager/` or to the following section.

HTML versions of the manuals are also available from the *Help* tab of the SUSE Manager Web interface.

NOTE: Obtaining the Release Notes

Although this manual reflects the most current information possible, read the *SUSE Manager Release Notes* for information that may not have been available prior to the finalization of the documentation. The notes can be found at `http://www.novell.com/documentation/suse_manager/`.

1 Available Documentation

The following manuals are available on this product:

Quick Start (↑Quick Start)
: Guides you step by step through the installation, setup and basic configuration of SUSE Manager.

Proxy Quick Start (↑Proxy Quick Start)
: Gives an overview of the installation and setup of SUSE Manager Proxy.

Installation Guide (↑Installation Guide)
: Lists installation scenarios and example topologies for different SUSE Manager setups. Also contains detailed information about SUSE Manager maintenance and troubleshooting.

Client Configuration Guide (page i)
: Describes best practices for setting up clients to connect to a SUSE Manager server or SUSE Manager Proxy.

Reference Guide (↑Reference Guide)
: Reference documentation that covers administration topics like registering and updating client systems, configuring the SUSE Manager daemon, using the Web interface, monitoring client systems, and more. Also contains a glossary with key terms used in the SUSE Manager context.

HTML versions of the product manuals can be found in the installed system under `/usr/share/doc/manual`. Find the latest documentation updates at http://www.novell.com/documentation where you can download PDF or HTML versions of the manuals for your product.

2 Feedback

Several feedback channels are available:

Bugs and Enhancement Requests
: For services and support options available for your product, refer to http://www.suse.com/support/.

: To report bugs for a product component, log into the Novell Customer Center from http://www.suse.com/support/ and select *My Support* > *Service Request*.

User Comments
: We want to hear your comments about and suggestions for this manual and the other documentation included with this product. Use the User Comments feature at the bottom of each page in the online documentation or go to http://www

.suse.com/documentation/feedback.html and enter your comments there.

Mail
For feedback on the documentation of this product, you can also send a mail to doc-team@suse.de. Make sure to include the document title, the product version and the publication date of the documentation. To report errors or suggest enhancements, provide a concise description of the problem and refer to the respective section number and page (or URL).

3 Documentation Conventions

The following typographical conventions are used in this manual:

- `/etc/passwd`: directory names and filenames.
- *`placeholder`*: replace *`placeholder`* with the actual value.
- `PATH`: the environment variable PATH.
- `ls`, `--help`: commands, options, and parameters.
- `user`: users or groups.
- Alt, Alt + F1: a key to press or a key combination; keys are displayed with uppercase letters as on a keyboard.
- *File*, *File* > *Save As*: menu items, buttons.
- ► **amd64 em64t:** This paragraph is only relevant for the specified architectures. The arrows mark the beginning and the end of the text block. ◄
- *Dancing Penguins* (Chapter *Penguins*, ↑Another Manual): This is a reference to a chapter in another manual.

1 Introduction

This best practices guide is intended to help customers of SUSE Manager to configure their client systems more easily.

By default, all SUSE Manager client applications are configured to communicate with central network services. When connecting clients to SUSE Manager Server or SUSE Manager Proxy Server instead, many of these settings have to be modified. Altering client settings for a system or two may be relatively simple. A large enterprise environment, containing hundreds or thousands of systems, will likely benefit from the mass reconfiguration steps described here.

Due to the complexity of this undertaking, customers may utilize a pre-populated script that automates many of the tasks necessary to access their SUSE Manager Server or SUSE Manager Proxy server. Refer to Chapter 5, *Using Bootstrap* (page 27) for details. Understanding the implications of these changes is helpful and therefore describes the manual steps for reconfiguration in the opening chapters.

Although many of the commands provided within this guide can be applied as they appear, it is impossible to predict all potential network configurations adopted by customers. Therefore, we encourage you to use these commands as references that must take into account your organization's individual settings.

Red Hat Linux Client Applications

2

In order to utilize most enterprise-class features of SUSE Manager, configuration of the latest client applications is required. Obtaining these applications before the client has registered with SUSE Manager can be difficult. This paradox is especially problematic for customers migrating large numbers of older systems to SUSE Manager. This chapter identifies techniques to resolve this dilemma.

IMPORTANT: Latest Updates and Firewalls on Client Systems

It is strongly recommended to install the latest system updates on any client system connected to SUSE Manager or SUSE Manager Proxy to ensure proper connectivity.

Additionally, make sure to open ports `80` and `443` on the client firewalls for proper functionality with SUSE Manager.

2.1 Deploying the Latest Client RPMs

The Package Updater, `yum` (`up2date` on earlier RHEL versions) is a prerequisite for using much of SUSE Manager enterprise functionality. It is crucial to install it on client systems before attempting to use SUSE Manager Proxy Server or SUSE Manager Server in your environment.

There are several approaches to accomplish this update of the SUSE Manager client software. One of which involves storing the RPMs in a location that is accessible by all client systems and deploying the packages with the simplest command possible. In nearly all cases, a manual deployment of `yum` and `pup` (`up2date` for earlier version of Red Hat Enterprise Linux) does not have to be performed. Those client tools should have no issues connecting to your SUSE Manager or Proxy environment. The discussion below assumes that the "out of box" `yum` and `pup` (or `up2date`) are not the latest and do not work for your environment.

NOTE: Removing the Obsolete `rhn_register` Tool

Client system to be migrated to SUSE Manager must no longer use the `rhn_register` tool. During the migration, `spacewalk-client-tools` removes `rhn_register`. If it fails, remove `rhn_register` manually.

Remember, only systems running Red Hat Enterprise Linux 5 or 6 systems must have registered with SUSE Manager in `firstboot` after installation or use the rhn_register . Systems running Red Hat Enterprise Linux 4 can use the registration functionality built into the Red Hat Update Agent.

This document presumes that the customer has installed at least one SUSE Manager Server or SUSE Manager Proxy Server on their network. The example below demonstrates a simple approach of deploying `yum` and `pup` (or `up2date`) for the first time by an administrator assuming the machines do not already have a working Novell Customer Center setup. The administrator has populated the `/srv/www/htdocs/pub/` directory with a copy of the `yum` and `pup` (or `up2date`) RPMs that the client systems need, and then has simply deployed those RPMs onto the client systems with a simple `rpm -Uvh` command. Run from a client, this command installs the RPMs to that client, assuming the domain name, paths, and RPM versions are correct (replace *proxy_or_sat.domain* with your addresses):

```
rpm -Uvh \
http://proxy_or_sat.domain.com/pub/rhn-setup-0.4.17-8.el5.i386.rpm \
http://proxy_or_sat.domain.com/pub/yum-3.2.8-9.el5.i386.rpm \
http://proxy_or_sat.domain.com/pub/pirut-1.3.28-13.3l5.noarch.rpm
```

Keep in mind that the architecture (in this case, `i386`) may need to be altered depending on the systems to be served.

2.2 Configuring the Client Applications

Not every customer has to connect securely to a SUSE Manager Server or SUSE Manager Proxy Server within their organization and not every customer needs to build and deploy a GPG key for custom packages. These topics are explained in detail later. Every customer who uses SUSE Manager Server or SUSE Manager Proxy Server must reconfigure the Red Hat Update Agent (`up2date`) and the registration tool to redirect it to their SUSE Manager Server or SUSE Manager Proxy Server.

IMPORTANT

Although this is not configurable, note that the port used by the up2date is 80 for HTTP and 443 for secure HTTP (HTTPS). By default, `yum` on Red Hat Enterprise Linux 5 uses SSL only. For this reason, users should ensure that their firewalls allow connections over port 443. To bypass SSL, change the protocol for `serverURL` from `https` to `http` in `/etc/sysconfig/rhn/up2date`. If using SUSE Manager's Monitoring feature and probes requiring the Red Hat Network Monitoring `rhnmd` Daemon, client systems must allow connections on port 4545 (or port 22, if using `sshd` instead).

By default, client systems refer to the main SUSE Manager Servers. Users must reconfigure client systems to refer to their SUSE Manager Server or SUSE Manager Proxy Server.

The latest versions of the Red Hat Update Agent can be configured to accommodate several SUSE Manager Servers, thereby providing failover protection in case the primary server is inaccessible. Refer to Section 2.2.4, “Implementing Server Failover” (page 9) for instructions on enabling this feature.

The next sections describe three methods of configuring the client systems to access your SUSE Manager Server or SUSE Manager Proxy Server: using an Activation Key, `up2date --configure`, and manually updating the configuration files. To see how virtually all reconfiguration can be scripted, refer to Chapter 6, *Manually Scripting the Configuration* (page 35).

2.2.1 Registering with Activation Keys

SUSE recommends using activation keys for registering and configuring client systems that access SUSE Manager Proxy Server or SUSE Manager Server. Activation keys can be used to register, entitle, and subscribe systems in a batch. Refer to Section “Activation Keys — [Mgmt]” (Chapter 4, *SUSE Manager Web Interface*, ↑Reference Guide) for more information on activation keys.

Registering with an activation key has four basic steps:

1. Generate an activation key.

2. Import custom GPG keys.

3. Download and install the SSL Certificate RPM from the `/pub/` directory of the SUSE Manager Proxy Server or SUSE Manager Server. The command for this step looks something like this:

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

4. Register the system with your SUSE Manager Proxy Server or SUSE Manager Server. The command for this step looks something like:

   ```
   rhnreg_ks --activationkey mykey \
   --serverUrl https://your-suse_manager-FQDN/XMLRPC
   ```

Alternatively, most of the above steps can be combined in a shell script that includes the following lines. This command has been split into multiple lines for print and PDF purposes but should be typed as one line at a shell prompt:

```
wget -O - http://your-suse_manager-FQDN/pub/bootstrap.sh | bash
```

The bootstrap script, generated at installation and available for both SUSE Manager Server and SUSE Manager Proxy Server, is such a script. The script and the `mgr-bootstrap` that generates it are discussed in detail in Chapter 5, *Using Bootstrap* (page 27).

2.2.2 The up2date --configure Option

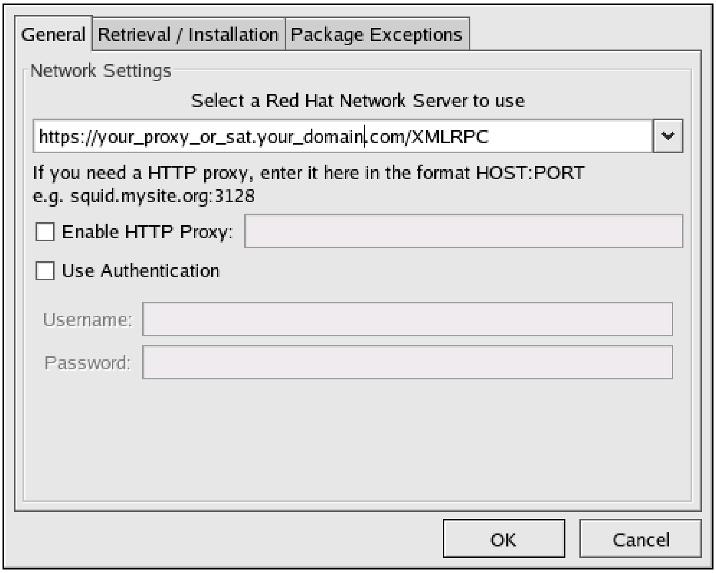
The Red Hat Update Agent in Red Hat Enterprise Linux 4 provides an interface for configuring various settings. For full listings of these settings, refer to the `up2date` manual page (`man up2date` at a command line).

To reconfigure the Red Hat Update Agent, issue the following command as root:

```
up2date --configure
```

You are presented with a dialog box offering various settings that may be reconfigured. In the *General* tab, under `Select a SUSE Manager Server to use` replace the default value with the fully qualified domain name (FQDN) of the SUSE Manager Server or SUSE Manager Proxy Server, such as `https://your_proxy_or_susemgr.your_domain.com/XMLRPC`. Retain the `/XMLRPC` at the end. When finished, click *OK*.

Figure 2.1 *Red Hat Update Agent GUI Configuration*



Select a Red Hat Network Server to use

https://your_proxy_or_sat.your_domain.com/XMLRPC

If you need a HTTP proxy, enter it here in the format HOST:PORT
e.g. squid.mysite.org:3128

□ Enable HTTP Proxy:

□ Use Authentication

Username:

Password:

Make sure you enter the domain name of your SUSE Manager Server or SUSE Manager Proxy Server correctly. Entering an incorrect domain or leaving the field blank may prevent `up2date --configure` from launching. This may be resolved, however, by editing the value in the up2date configuration file. Refer to Section 2.2.3, "Updating the Configuration Files Manually" (page 8) for precise instructions.

WARNING

Systems running Red Hat Enterprise Linux 4 have registration functionality built into the Red Hat Update Agent and therefore do not install the Red Hat Network Registration Client. Systems on Red Hat Enterprise Linux 5 or 6 do not use up2date, and need rhnreg_ks to register their systems to SUSE Manager and yum and pup to update their packages.

2.2.3 Updating the Configuration Files Manually

As an alternative to the GUI interface described in the previous section, users may also reconfigure the Red Hat Update Agent by editing the application's configuration file.

To configure Red Hat Update Agent on the client systems connecting to the SUSE Manager Proxy Server or SUSE Manager Server, edit the values of the serverURL and noSSLServerURL settings in the /etc/sysconfig/rhn/up2date configuration file (as root). Replace the default URL with the fully qualified domain name (FQDN) for the SUSE Manager Proxy Server or SUSE Manager Server. For example:

```
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
```

WARNING

The httpProxy setting in /etc/sysconfig/rhn/up2date does *not* refer to the SUSE Manager Proxy Server. It is used to configure an optional HTTP proxy for the client. With an SUSE Manager Proxy Server in place, leave the httpProxy blank.

2.2.4 Implementing Server Failover

Beginning with up2date-4.2.38, the Red Hat Update Agent can be configured to seek updates from a series of SUSE Manager Servers. This can be especially helpful in sustaining constant updates if your primary SUSE Manager Proxy Server or SUSE Manager Server is taken offline.

To use this feature, first ensure that you are running the required version of up2date. Then manually add the secondary servers to the serverURL and noSSLServerURL settings in the /etc/sysconfig/rhn/up2date configuration file (as root). Add the fully qualified domain names (FQDN) for the Proxy or SUSE Manager immediately after the primary server, separated by a semicolon (;). For example:

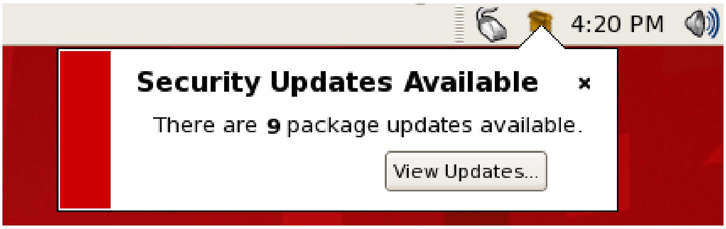
```
serverURL[comment]=Remote server URL
serverURL=https://your_primary.your_domain.com/XMLRPC;
https://your_secondary.your_domain.com/XMLRPC;

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

Connection to the servers is attempted in the order provided here. You can include as many servers as you wish. You may list the central SUSE Manager Servers, as well. This makes sense, however, only if the client systems can reach the Internet.

2.3 The Package Updater Applet

Red Hat Enterprise Linux 5 and 6 features a running program on the graphical desktop panel that periodically checks for updates from SUSE Manager server and will alert users when a new update is available.

***Figure 2.2** Package Updater Applet*



Security Updates Available

There are 9 package updates available.

The Package Updater Applet stays in the notification tray of the desktop panel and checks for new updates periodically. The applet also allows you to perform a few package maintenance tasks from the applet by clicking the notification icon and choosing from the following actions:

- Refresh — Check SUSE Manager for new updates
- View Updates — Launch the Package Updater application so you can see any available updates in more detail and configure the updates according to your needs
- Apply Updates — Download and install all updated packages.
- Quit — Close the applet

3 SSL Infrastructure

For SUSE Manager customers, security concerns are of the utmost importance. One of the strengths of SUSE Manager is its ability to process every single request over Secure Sockets Layer (SSL). To maintain this level of security, customers installing SUSE Manager within their infrastructures must generate custom SSL keys and certificates.

Manual creation and deployment of SSL keys and certificates can be quite involved. Both the SUSE Manager Proxy Server and the SUSE Manager Server allow you to build your own SSL keys and certificates based on your own private Certificate Authority (CA) during installation. In addition, a separate command line utility, the SUSE Manager SSL Maintenance Tool, exists for this purpose. Regardless, these keys and certificates must then be deployed to all systems within your managed infrastructure. In many cases, deployment of these SSL keys and certificates is automated for you. This chapter describes efficient methods for conducting all these tasks.

Please note that this chapter does not explain SSL in depth. The SUSE Manager SSL Maintenance Tool was designed to hide much of the complexity involved in setting up and maintaining this public key infrastructure (PKI). For more information, please consult some of the many good references available at your nearest bookstore.

3.1 A Brief Introduction To SSL

SSL, or Secure Sockets Layer, is a protocol that enables client-server applications to pass information securely. SSL uses a system of public and private key pairs to encrypt communication passed between clients and servers. Public certificates can be left accessible, while private keys must be secured. The mathematical relationship (a digital signature) between a private key and its paired public certificate makes this system work. Through this relationship, a connection of trust is established.

NOTE

When discussing SSL, it is referred to the public half of an SSL key pair (or key set) as the SSL public certificate.

An organization's SSL infrastructure is generally made up of these SSL keys and certificates:

- Certificate Authority (CA) SSL private key and public certificate — generally only one set per organization generated. The public certificate is digitally signed by its private key. The public certificate is distributed to every system.

- Web server SSL private key and public certificate — one set per application server. The public certificate is digitally signed by both its private key and the CA SSL private key. We often refer to a Web server's key set; this is because there is an intermediary SSL certificate request that is generated. All three are deployed to a SUSE Manager Server.

Scenario: If you have one SUSE Manager Server and five SUSE Manager Proxy Servers, you will generate one CA SSL key pair and six Web server SSL key sets. The CA SSL public certificate is distributed to all systems and used by all clients to establish a connection to their respective upstream servers. Each server has its own SSL key set that is specifically tied to that server's hostname and generated using its own SSL private key and the CA SSL private key in combination. This establishes a digitally verifiable association between the Web server's SSL public certificate and the CA SSL key pair and server's private key. The Web server's key set cannot be shared with other Web servers.

IMPORTANT

The most critical portion of this system is the CA SSL key pair. From that private key and public certificate an administrator can regenerate any Web server's SSL key set. This CA SSL key pair must be secured. It is highly recommended that once the entire SUSE Manager infrastructure of servers is set up and running, you archive the SSL build directory generated by this tool and/or the installers onto separate media, write down the CA password, and secure the media and password in a safe place.

3.2 The SUSE Manager SSL Maintenance Tool

SUSE Manager provides a command line tool to ease management of your secure infrastructure: the SSL Maintenance Tool, commonly known by its command `mgr-ssl-tool` . This tool is available as part of the `spacewalk-certs-tools` package. This package can be found within the software channels for the latest SUSE Manager Proxy Server and SUSE Manager Server (as well as the SUSE Manager Server ISO). SSL Maintenance Tool enables you to generate your own Certificate Authority SSL key pair, as well as Web server SSL key sets (sometimes called key pairs).

This tool is only a build tool. It generates all the required SSL keys and certificates. It also packages the files in RPM format for quick distribution and installation on all client machines. It does not deploy them, however. That is either left to the administrator, or is automated by the SUSE Manager Server.

NOTE

The `spacewalk-certs-tools`, which contain `mgr-ssl-tool`, can be installed and run on any current SUSE Linux Enterprise Server system with minimal requirements. This is offered as a convenience for administrators who wish to manage their SSL infrastructure from their workstation or from a system other than their SUSE Linux Enterprise Server.

The tool is required for the following cases:

- When updating your CA public certificate - rare.
- When installing a SUSE Manager Proxy Server version 1.2 or later that connects to the central SUSE Manager Servers as its top-level service - the hosted service, for security reasons, cannot be a repository for your CA SSL key and certificate, which is private to your organization.
- When reconfiguring your SUSE Manager infrastructure to use SSL where it previously did not.
- When adding SUSE Manager Proxy Servers of versions prior to 1.2 into your SUSE Manager infrastructure.

- When adding multiple SUSE Manager Servers to your SUSE Manager infrastructure.

The tool is *not* required for the following cases:

- During installation of a SUSE Manager Server - all SSL settings are configured during the installation process. The SSL keys and certificate are built and deployed automatically.

- During installation of an SUSE Manager Proxy Server version 1.2 or later if connected to an SUSE Manager Server version 1.2 or later as its top-level service - the SUSE Manager Server contains all the required SSL information to configure, build and deploy the SUSE Manager Proxy Server's SSL keys and certificates.

The installation procedures of both the SUSE Manager Server and the SUSE Manager Proxy Server ensure the CA SSL public certificate is deployed to the `/pub` directory of each server. This public certificate is used by the client systems to connect to the SUSE Manager Server. Refer to Section 3.3, “Deploying the CA SSL Public Certificate to Clients” (page 24) for more information.

If your organization's SUSE Manager infrastructure deploys the latest version of SUSE Manager Server as its top-level service, there will be no need to use the tool.

3.2.1 SSL Generation Explained

The primary benefits of using the SSL Maintenance Tool are security, flexibility, and portability. Security is achieved through the creation of distinct Web server SSL keys and certificates for each SUSE Manager server, all signed by a single Certificate Authority SSL key pair created by your organization. Flexibility is supplied by the tool's ability to work on any machine that has the `spacewalk-certs-tools` package installed. Portability exists in a build structure that can be safely stored anywhere and then installed whenever needed.

Again, if your infrastructure's top-level server is the most current SUSE Manager Server, you only have to restore your `ssl-build` tree from an archive to the `/root` directory and utilize the configuration tools provided within the SUSE Manager Server's Web site.

To make the best use of the SSL Maintenance Tool, complete the following high-level tasks in roughly this order. Refer to the remaining sections for the required details:

1. Install the `spacewalk-certs-tools` package on a system within your organization, for example, the SUSE Manager Server or SUSE Manager Proxy Server.

2. Create a single Certificate Authority SSL key pair for your organization and install the resulting RPM or public certificate on all client systems.

3. Create a Web server SSL key set for each Proxies and Servers to be deployed and install the resulting RPMs on the SUSE Manager Servers, restarting the `httpd` service afterwards:

   ```
   rcapache2 restart
   ```

4. Archive the SSL build tree - consisting of the primary build directory and all subdirectories and files - to a removable media, such as a floppy disk. (Disk space requirements are insignificant)

5. Verify and then store that archive in a safe location, such as the one described for backups in the *Additional Requirements* sections of either the Proxy or SUSE Manager installation guide.

6. Record and secure the CA password for future use.

7. Delete the build tree from the build system for security purposes, once the entire SUSE Manager infrastructure is in place and configured.

8. When additional Web server SSL key sets are needed, restore the build tree on a system running the SSL Maintenance Tool and repeat steps 3 through 7.

3.2.2 SSL Maintenance Tool Options

The SSL Maintenance Tool offers several command line options for generating your Certificate Authority SSL key pair and managing your server SSL certificates and keys. The tool offers essentially three command line option help listings: `mgr-ssl-tool --help` (general), `mgr-ssl-tool --gen-ca --help` (Certificate Authority), and `mgr-ssl-tool --gen-server --help` (Web server). The manual page for mgr-ssl-tool is also quite detailed and available to assist: `man mgr-ssl-tool`.

The two tables below break down the options by their related task, either CA or Web server SSL key set generation.

This set of options must be preceded by the `--gen-ca` argument:

Table 3.1 *SSL Certificate Authority (CA) Options (mgr-ssl-tool --gen-ca --help)*

Option	Description
`--gen-ca`	Generates a Certificate Authority (CA) key pair and public RPM. This must be issued with any of the remaining options in this table.
`-h`, `--help`	Displays the help screen with a list of base options specific to generating and managing a Certificate Authority.
`-f`, `--force`	Forcibly creates a new CA private key and/or public certificate.
`-p=`, `--password=`*`PASSWORD`*	The CA password. You will be prompted for this if it is missing. Record it in a safe manner.
`-d=`, `--dir=`*`BUILD_DIRECTORY`*	*Required for most commands* - The directory where certificates and RPMs are built. The default is `./ssl-build`.
`--ca-key=`*`FILENAME`*	The CA private key filename. The default is `RHN-ORG-PRIVATE-SSL-KEY`.
`--ca-cert=`*`FILENAME`*	The CA public certificate filename. The default is `RHN-ORG-TRUSTED-SSL-CERT`.
`--cert-expiration=`*`CA_CERT_EXPIRE`*	The expiration date of the public CA certificate. The default is the number of days until one day prior to epoch rollover (or 01-18-2038).

Option	Description
`--set-country=`*`COUNTRY_CODE`*	The two-letter country code. The default is US.
`--set-state=`*`STATE_OR_PROVINCE`*	The state or province of the CA. The default is ''.
`--set-city=`*`CITY_OR_LOCALITY`*	The city or locality. The default is ''.
`--set-org=`*`ORGANIZATION`*	The company or organization, such as SUSE. The default is Example Corp. Inc.
`--set-org-unit=`*`SET_ORG_UNIT`*	The organizational unit. The default is ''.
`--set-common-name=`*`HOSTNAME`*	*Not typically set for the CA.* - The common name.
`--set-email=`*`EMAIL`*	*Not typically set for the CA.* - The e-mail address.
`--rpm-packager=`*`PACKAGER`*	Packager of the generated RPM, such as "SUSE Admin (suse-admin@example.com)."
`--rpm-vendor=`*`VENDOR`*	Vendor of the generated RPM, such as "IS/IT Example Corp."
`-v`, `--verbose`	Displays verbose messaging. Accumulative - added "v"s result in increasing detail.
`--ca-cert-rpm=`*`CA_CERT_RPM`*	*Rarely changed* - RPM name that houses the CA certificate (the base filename, not filename-version-release.noarch.rpm).

Option	Description
`--key-only`	*Rarely used* - Generates only a CA private key. Review `--gen-ca --key-only --help` for more information.
`--cert-only`	*Rarely used* - Generates only a CA public certificate. Review `--gen-ca --cert-only --help` for more information.
`--rpm-only`	*Rarely used* - Generates only an RPM for deployment. Review `--gen-ca --rpm-only --help` for more information.
`--no-rpm`	*Rarely used* - Conducts all CA-related steps except RPM generation.

The following set of options must be preceded by the `--gen-server` argument:

Table 3.2 *SSL Web Server Options (mgr-ssl-tool --gen-server --help)*

Option	Description
`--gen-server`	Generate the Web server's SSL key set, RPM and tar archive. This must be issued with any of the remaining options in this table.
`-h`, `--help`	Displays the help screen with a list of base options specific to generating and managing a server key-pair.
`-p=`, `--password=`*`PASSWORD`*	The CA password. You will be prompted for this if it is missing. Record it in a safe manner.

Option	Description
`-d=`, `--dir=`*BUILD_DIRECTORY*	*Required for most commands* - The directory where certificates and RPMs are built. The default is `./ssl -build`.
`--server-key=`*FILENAME*	The Web server's SSL private key filename. The default is `server.key`.
`--server-cert-req=`*FILENAME*	The Web server's SSL certificate request filename. The default is `server .csr`.
`--server-cert=`*FILENAME*	The Web server's SSL certificate filename. The default is `server.crt`.
`--startdate=`*YYMMDDHHMMSSZ*	The start date for the server certificate validity in the format: year, month, date, hour, minute, second (two characters per value). Z stands for Zulu and is required. The default is one week before generation.
`--cert-expiration=`*SERVER_CERT_EXPIRE*	The expiration date of the server certificate. The default is the number of days until one day prior to epoch rollover (or 01-18-2038).
`--set-country=`*COUNTRY_CODE*	The two-letter country code. The default is US.
`--set-state=`*STATE_OR_PROVINCE*	The state or province. The default is North Carolina.
`--set-city=`*CITY_OR_LOCALITY*	The city or locality. .

Option	Description
`--set-org=`*`ORGANIZATION`*	The company or organization, such as SUSE. The default is Example Corp. Inc.
`--set-org-unit=`*`SET_ORG_UNIT`*	The organizational unit. The default is unit.
`--set-hostname=`*`HOSTNAME`*	The hostname of the SUSE Manager Server to receive the key. The default is dynamically set to the build machine's hostname.
`--set-email=`*`EMAIL`*	The e-mail address of the certificate contact. The default is admin@example.corp.
`--rpm-packager=`*`PACKAGER`*	Packager of the generated RPM, such as "SUSE Admin (suse-admin@example.com)."
`--rpm-vendor=`*`VENDOR`*	Vendor of the generated RPM, such as "IS/IT Example Corp."
`-v`, `--verbose`	Displays verbose messaging. Accumulative - added "v"s result in increasing detail.
`--key-only`	*Rarely used* - Generates only a server private key. Review `--gen-server --key-only --help` for more information.
`--cert-req-only`	*Rarely used* - Generates only a server certificate request. Review `--gen-server`

Option	Description
	`--cert-req-only --help` for more information.
`--cert-only`	*Rarely used* - Generates only a server certificate. Review `--gen-server --cert-only --help` for more information.
`--rpm-only`	*Rarely used* - Generates only an RPM for deployment. Review `--gen-server --rpm-only --help` for more information.
`--no-rpm`	*Rarely used* - Conducts all server-related steps except RPM generation.
`--server-rpm=`*`SERVER_RPM`*	*Rarely changed* - RPM name that houses the Web server's SSL key set (the base filename, not filename-version-release.noarch.rpm).
`--server-tar=`*`SERVER_TAR`*	*Rarely changed* - Name of .tar archive of the Web server's SSL key set and CA public certificate that is used solely by the hosted SUSE Manager Proxy Server installation routines (the base filename, not filename-version-release.tar).

3.2.3 Generating the Certificate Authority SSL Key Pair

Before creating the SSL key set required by the Web server, you must generate a Certificate Authority (CA) SSL key pair. A CA SSL public certificate is distributed to client systems of the SUSE Manager or SUSE Manager Proxy. The SSL Maintenance

Tool allows you to generate a CA SSL key pair if needed and re-use it for all subsequent SUSE Manager Server deployments.

The build process automatically creates the key pair and public RPM for distribution to clients. All CA components end up in the build directory specified in the command line, typically `/root/ssl-build` (or `/etc/sysconfig/rhn/ssl` for older SUSE Manager and Proxies). To generate a CA SSL key pair, issue a command like this:

```
mgr-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"
```

Replace the example values with those appropriate for your organization. This will result in the following relevant files in the specified build directory:

- `RHN-ORG-PRIVATE-SSL-KEY` — the CA SSL private key

- `RHN-ORG-TRUSTED-SSL-CERT` — the CA SSL public certificate

- `rhn-org-trusted-ssl-cert-`*`VER-REL`*`.noarch.rpm` — the RPM prepared for distribution to client systems. It contains the CA SSL public certificate (above) and installs it in this location: `/usr/share/rhn/ RHN-ORG-TRUSTED-SSL-CERT`

- `rhn-ca-openssl.cnf` — the SSL CA configuration file

- `latest.txt` — always lists the latest versions of the relevant files.

Once finished, you are ready to distribute the RPM to client systems. Refer to Section 3.3, “Deploying the CA SSL Public Certificate to Clients” (page 24).

3.2.4 Generating Web Server SSL Key Sets

Although you must have a CA SSL key pair already generated, you will likely generate Web server SSL key sets more frequently, especially if more than one Proxy or SUSE Manager is deployed. Note that the value for `--set-hostname` is different for each server. In other words, a distinct set of SSL keys and certificates must be generated and installed for every distinct SUSE Manager server hostname.

The server certificate build process works much like CA SSL key pair generation with the exception that all server components end up in subdirectories of the build directory that reflect the build system's machine name, such as `/root/ssl-build/MACHINE_NAME`. To generate server certificates, issue a command like this:

```
mgr-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="mgrbox1.example.com
```

Replace the example values with those appropriate for your organization. This will result in the following relevant files in a machine-specific subdirectory of the build directory:

- `server.key` — the Web server's SSL private server key
- `server.csr` — the Web server's SSL certificate request
- `server.crt` — the web server's SSL public certificate
- `rhn-org-httpd-ssl-key-pair-`*`MACHINE_NAME-VER-REL`*`.noarch.rpm` — the RPM prepared for distribution to SUSE Manager Servers. Its associated src.rpm file is also generated. This RPM contains the three files above. It will install them in these locations:
    - `/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 — the Web server's SSL configuration file
- `latest.txt` — always lists the latest versions of the relevant files.

Once finished, you are ready to distribute and install the RPM on its respective SUSE Manager Server. Note that the `httpd` service must be restarted after installation:

```
rcapache2 restart
```

3.3 Deploying the CA SSL Public Certificate to Clients

Both the SUSE Manager Proxy Server and SUSE Manager Server installation processes make client deployment relatively easy by generating a CA SSL public certificate and RPM. These installation processes make them publicly available by placing a copy of one or both into the `/srv/www/htdocs/pub/` directory of the SUSE Manager Server.

This public directory can be inspected easily by simply browsing to it via any Web browser: http://proxy-or-susemgr.example.com/pub/.

The CA SSL public certificate in that directory can be downloaded to a client system using `wget` or `curl`. For example:

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

Alternatively, if the CA SSL public certificate RPM resides in the `/pub` directory, it can be installed on a client system directly:

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

Confirm the actual name of the certificate or RPM before running these commands.

3.4 Configuring Client Systems

Once the RPM or raw certificate has been deployed to a client system, the administrator of that system must then alter the configuration files of the Update Agent / Online Updater to use the new CA SSL public certificate file and connect to the appropriate SUSE Manager Proxy Server or SUSE Manager Server. The generally accepted location for that CA SSL public certificate is in the `/usr/share/rhn` directory.

The SUSE Manager Proxy Server and SUSE Manager Server both have `mgr-bootstrap` installed by default, which can greatly reduce these repetitive steps and simplify the process of registering and configuring client systems. Refer Chapter 5, *Using Bootstrap* (page 27) for details.

4 Importing Custom GPG Keys

For customers who plan to build and distribute their own RPMs securely, it is strongly recommended that all custom RPMs are signed using GNU Privacy Guard (GPG). Generating GPG keys and building GPG-signed packages are covered in the *Channel Management Guide*.

Once the packages are signed, the public key must be deployed on all systems importing these RPMs creating a central location for the public key for clients to retrieve it and then adding the key to the local GPG keyring for each system. Follow these steps:

1 Use the Web site approach recommended for deploying SUSE Manager client applications to create a central location for the public keys. (Refer to Section 2.1, “Deploying the Latest Client RPMs” (page 3).) Create a public directory on the Web server and place the GPG public signature in it:

```
cp /some/path/YOUR-RPM-GPG-KEY /srv/www/htdocs/pub/
```

The key can then be downloaded by client systems using Wget:

```
wget -O- -q http://your_susemgr.your_domain.com/pub/YOUR-RPM-GPG-KEY
```

The `-O-` option sends results to standard output while the `-q` option sets Wget to run in quiet mode. Remember to replace the *`YOUR-RPM-GPG-KEY`* variable with the filename of your key.

2 Once the key is available on the client file system, import it into the local GPG keyring using the following command:

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

Once the GPG key has been successfully added to the client, the system should be able to validate custom RPMs signed with the corresponding key.

5 Using Bootstrap

SUSE Manager provides SUSE Manager Bootstrap that automates much of the manual reconfiguration described in previous chapters. This tool plays an integral role in the SUSE Manager server installation program, enabling generation of the bootstrap script during installation.

SUSE Manager Bootstrap, invoked with the command `mgr-bootstrap`, is installed by default on both SUSE Manager Server and SUSE Manager Proxy Server. The Bootstrap tool generates a script that can be run from any client system to conduct the following tasks:

- Redirect client applications to the SUSE Manager Proxy or SUSE Manager
- Import custom GPG keys
- Install SSL certificates
- Register the system to SUSE Manager and particular system groups and channels with the help of activation keys
- Perform miscellaneous post-configuration activities, including updating packages, performing reboots, and altering SUSE Manager configuration

WARNING: Security Risks

The inherent risks of using a script for configuration: Security tools such as SSL certificates are installed by the script itself and therefore, do not yet exist on the systems and cannot be used to process transactions. This allows for the possibility of someone impersonating the SUSE Manager server and transmitting bad data. Virtually all SUSE Manager servers and client systems operate behind customer firewalls and are restricted from outside traffic which mitigates the risk. Registration is conducted via SSL and is therefore protected.

The bootstrap script `bootstrap.sh` is automatically placed in the `/srv/www/htdocs/pub/bootstrap/` directory on the SUSE Manager server. From there it can be downloaded and run on all client systems. Some preparation and post-generation editing is required, as identified in the following sections. Refer to Section 5.4, “Bootstrap Options” (page 31) for the tool's complete list of options, and to the Appendix A, *Sample Bootstrap Script* (page 43) for an example script.

5.1 Preparation

Bootstrap (`mgr-bootstrap`) depends on other components of the SUSE Manager infrastructure to properly configure client systems. These components must be prepared before generating the script. The following list identifies suggested initial measures:

- Generate activation keys to be called by the script. Activation keys can be used to register client systems, entitle them to a SUSE Manager service level, and subscribe them to specific channels and system groups, all in one action. You must have Management entitlements available to use an activation key, while Provisioning entitlements is required if using multiple activation keys at once. Generate activation keys through the *Activation Keys* page within the *Systems* category of the SUSE Manager Web site. Refer to Section “Activation Keys — [Mgmt]” (Chapter 4, *SUSE Manager Web Interface*, ↑Reference Guide) for instructions on creation and use.

- We recommend to sign your RPMs by a custom GPG key. Make the key available so you are able to refer to it from the script. Place the key in the `/srv/www/htdocs/pub/` directory of the SUSE Manager server, per Chapter 4, *Importing Custom GPG Keys* (page 25).

NOTE: Importing Red Hat RPM GPG Key

If the Red Hat release key is not installed it must be provided in the server's `/srv/www/htdocs/pub` directory, and its name must be added to the bootstrap scripts ORG_GPG_KEY list. This way the key gets installed when the bootstrap script is run.

- If you wish to use the script to deploy your CA SSL public certificate, have the certificate or the package (RPM) containing that certificate available on the SUSE Manager server and include it during script generation with the --ssl-cert option. Refer to Chapter 3, *SSL Infrastructure* (page 11) for details.

- Have the values ready to develop one or many bootstrap scripts, depending on the variety of systems to be reconfigured. Since mgr-bootstrap provides a full set of reconfiguration options, use it to generate different bootstrap scripts to accommodate each type of system. For instance, bootstrap-web-servers.sh might be used to reconfigure your Web servers, while bootstrap-app-servers.sh can handle the application servers. See Section 5.4, “Bootstrap Options” (page 31) for the complete options list.

5.2 Generation

You now can use mgr-bootstrap to generate the required scripts. Log in to your SUSE Manager server or SUSE Manager Proxy server as root and issue the mgr-bootstrap command followed by the desired options and values. If no options are included, a bootstrap.sh file is created in the bootstrap subdirectory that contains the essential values derived from the server, including hostname, the SSL certificate, if it exists, SSL and GPG settings, and a call for the client-config-overrides.txt file.

We strongly recommend to at least accommodate activation keys, GPG keys, and advanced configuration options in your scripts in the following manner:

- Use the --activation-keys option to include keys, taking into account the Entitlement requirements identified in Section 5.1, “Preparation” (page 28).

- Use the --gpg-key option to identify the key path and filename during script generation. Otherwise, use the --no-gpg option to turn off this verification on client systems. We recommend retaining this security measure.

- Include the `--allow-config-actions` flag to enable remote configuration management on all client systems touched by the script. This feature is useful in reconfiguring multiple systems simultaneously.

- Include the `--allow-remote-commands` flag to enable remote script use on all client systems. Like configuration management, this feature helps reconfiguring multiple systems.

When finished, your command looks something like this:

```
mgr-bootstrap --activation-keys KEY1,KEY2 \
              --gpg-key /srv/www/htdocs/pub/MY_CORPORATE_PUBLIC_KEY \
              --allow-config-actions \
              --allow-remote-commands
```

Include the actual key names. Refer to Section 5.4, “Bootstrap Options” (page 31) for the complete list of options.

5.3 Script Use

When your script is prepared, log in to the SUSE Manager server or SUSE Manager Proxy server, navigate to the `/srv/www/htdocs/pub/bootstrap/` directory and run the following command, altering the hostname and name of the script as needed to suit the system type:

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

A less secure alternative is to use either `wget` or `curl` to retrieve and run the script from every client system. Log in to each client machine and issue the following command, altering script and hostname accordingly:

`wget`:

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

`curl`:

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

After running this script on each client system, the configuration is finished and the SUSE Manager can be used.

5.4 Bootstrap Options

The SUSE Manager Bootstrap offers many command line options for creating client boostrap scripts. Although descriptions of these options can be found in the following table, ensure that they are available in the version of the tool installed on your SUSE Manager server by using the command `mgr-bootstrap --help` or reviewing its man page.

Bootstrap Options

`-h`, `--help`
: Display the help screen with a list of options specific to generating the bootstrap script.

`--activation-keys=`*`ACTIVATION_KEYS`*
: Activation key(s) as defined in the SUSE Manager Web site with multiple entries separated by a comma and no space

`--overrides=`*`OVERRIDES`*
: Configuration overrides filename. The default is `client-config-overrides.txt`.

`--script=`*`SCRIPT`*
: The bootstrap script filename. The default is `bootstrap.sh`.

`--hostname=`*`HOSTNAME`*
: The fully qualified domain name (FQDN) of the server to which client systems will connect.

`--ssl-cert=`*`SSL_CERT`*
: The path to your organization's public SSL certificate, either a package or a raw certificate. It will be copied to the `--pub-tree` option. A value of `""` will force a search of `--pub-tree`.

`--gpg-key=`*`GPG_KEY`*

The path to your organization's public GPG key, if used. It will be copied to the location specified by the `--pub-tree` option.

`--http-proxy=`*`HTTP_PROXY`*

The HTTP proxy setting for the client systems in the form `hostname:port`. A value of `""` disables this setting.

`--http-proxy-username=`*`HTTP_PROXY_USERNAME`*

If using an authenticating HTTP proxy, specify a username. A value of `""` disables this setting.

`--http-proxy-password=`*`HTTP_PROXY_PASSWORD`*

If using an authenticating HTTP proxy, specify a password.

`--allow-config-actions`

Boolean; including this option sets the system to allow all configuration actions via Novell Customer Center. This requires installing certain rhncfg-* packages, possibly through an activation key.

`--allow-remote-commands`

Boolean; including this option sets the system to allow arbitrary remote commands via Novell Customer Center. This requires installing certain rhncfg-* packages, possibly through an activation key.

`--no-ssl`

Not recommended - Boolean; including this option turns SSL off on the client system.

`--no-gpg`

Not recommended - Boolean; including this option turns GPG checking off on the client system.

`--no-up2date`

Not recommended - Boolean; including this option ensures `zypper up` will not run once the system has been bootstrapped.

`--pub-tree=`*`PUB_TREE`*

Change not recommended - The public directory tree where the CA SSL certificate and package will land; the bootstrap directory and scripts. The default is `/srv/www/htdocs/pub/`.

`--force`

Not recommended - Boolean; including this option forces bootstrap script generation despite warnings.

`-v`, `--verbose`

Display verbose messaging. Accumulative; `-vvv` causes extremely verbose messaging.

Manually Scripting the Configuration

6

This chapter provides an alternative to using Bootstrap to generate the bootstrap script. With these instructions, you should be able to create your own bootstrap script from scratch.

All of the initial techniques have shared a common theme: the deployment of necessary files in a centralized location to be retrieved and installed using simple, scriptable commands run on each client. This chapter explains how to create a single script that can be invoked by any system in your organization.

Combining all the commands from the previous chapters on a SLES 11 SP1 system, results in the following script:

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

zypper ar \
  http://susemgr.example.com/pub/repositories/susemanager-client-setup \
  susemanager-client-setup
zypper in spacewalk-check spacewalk-client-setup spacewalk-client-tools \
  rhncfg-actions rhncfg-client rhncfg-management zypp-plugin-spacewalk

# Second, reconfigure the clients to talk to the correct server.

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

# Third, install the SSL client certificate for your company's
# SUSE Manager Server or Proxy Server.
rpm -Uvh http://susemgr.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, install the SSL certificate in /etc/ssl/certs:
-->
test -e "/etc/ssl/certs/RHN-ORG-TRUSTED-SSL-CERT.pem" || {
  ln -s "/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT" \
        "/etc/ssl/certs/RHN-ORG-TRUSTED-SSL-CERT.pem"
  /usr/bin/c_rehash /etc/ssl/certs/
}


# Sixth, download the GPG key needed to validate custom packages.
wget -O - -q http://susemgr.example.com/pub/YOUR-RPM-GPG-KEY


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

Remember, the sixth step is documented here as it pertains to systems running SLES 11 SP1 or newer.

This script comprises a clean and repeatable process that should fully configure any potential SUSE Manager client in preparation for registration to a SUSE Manager Server. Remember, key values, such as the URL of your SUSE Manager Server, its public directory, and your actual GPG key must be inserted into the placeholders listed within the script. Depending on your environment, additional modifications may be required. Although this script may work nearly verbatim, it should be used as a guide.

Like its components, this script should be centrally located. By placing this script in the `/pub/` directory of the server, running `wget -O-` on it, and piping the output to a shell session, you are able to run the entire bootstrap process with a single command from each client:

```
wget -O - http://susemgr.example.com/pub/bootstrap_script | bash
```

WARNING

Running a shell script directly from input piped in over a Web connection has some inherent security risks. Therefore, it is vital to ensure the security of the source server in this instance.

This one-line command may then be invoked across all systems on a network. If the administrator has SSH access to all systems in question, it would be a simple task to iterate over a list of these systems and run the command remotely on all of them. This script would also be a perfect addition to the `%post` section of an existing AutoYaST script.

Implementing and Tracking Autoinstallation 7

Once all the configuration issues have been resolved, a system should register with the local SUSE Manager using the `rhnreg_ks` utility that comes with the `spacewalk-client-setup` package. This chapter discusses the use of `rhnreg_ks` to register systems and tracking autoinstallations.

7.1 The Use of rhnreg_ks

The `rhnreg_ks` utility uses *activation keys* to register, entitle, and subscribe systems to specified channels in one swift motion. For more information about activation keys, refer to Section “Activation Keys — [Mgmt]” (Chapter 4, *SUSE Manager Web Interface*, ↑Reference Guide).

The following commented Kickstart file is an example of how a Red Hat Enterprise Linux system can be configured using Kickstart.

```
# 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.

%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
( # 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 a SUSE Manager server.

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 SUSE Manager
# 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
```

7.2 Tracking Autoinstallations

To track an autoinstallation, SUSE Manager is creating a unique registration key for each autoinstallation and if a client uses that key to register, SUSE Manager Manager knows that the installation has been finished successfully.

If you upload a profile, and you want to be able to track the installation, you need to use that tracking-key when you register to SUSE Manager. That key is stored in the `$redhat_management_key` variable of the profile. Use it like this (replace *`my_server_url`* with the address of your server):

```
rhnreg_ks --serverUrl=my_server_url \
  --sslCACert=$mycert \
  --activationkey=1-my-sles11-sp1-key-i586,$redhat_management_key
```

The best way to achieve it, is using the `spacewalk/sles_register` snippet like in following example:

```
...
<scripts>
  <init-scripts config:type="list">
    <script>
      <source><![CDATA[
$SNIPPET('spacewalk/sles_register')
]]>
      </source>
      <filename>ay_suse_manager_register.sh</filename>
      <debug config:type="boolean">true</debug>
    </script>
    ...
  </init-scripts>
  ...
</scripts>
...
```

Integrate this init-script into the XML file you want to upload. Then, the registration will be done with the correct tracking key automatically; for an example XML file, see Appendix B, *Sample Autoinstallation Script with Tracking* (page 51).. If you want to register the client with your own registration key in addition to the tracking key using this snippet, you must put the name of key into the `registration_key` variable.

In the Web interface (Section “Autoinstallation Details > Variables” (Chapter 4, *SUSE Manager Web Interface*, ↑Reference Guide)), click on *Variables* and, for exmple, enter:

```
registration_key=1-my-sles11-sp1-key-i586
```

This way, the snippet will do the registration with your key in addition to the tracking key.

To check the snippet click *Systems* > *Autoinstallation* > *Kickstart Snippets*. There you can find the `sles_register` snippet that will be included with the `$SNIPPET('spacewalk/sles_register')` statement as shown above in the XML fragment.

If you use an uploaded profile without a tracking key, SUSE Manager will never actually notice when the installation is finished. The machine will appear as a registered client but after a while, SUSE Manager will mark the installation as failed, because the client did not registered with the tracking key.

Sample Bootstrap Script

A

The /srv/www/htdocs/pub/bootstrap/bootstrap.sh script generated by the SUSE Manager Server installation program provides the ability to reconfigure client systems to access your SUSE Manager easily. It is available through the mgr-bootstrap command. After modifying the script for your particular use, it can be run on each client machine.

Review the sample and its comments, beginning with a hash mark (#), for additional details. Follow the steps in Chapter 5, *Using Bootstrap* (page 27) to prepare the script for use.

```
#!/bin/bash
echo "SUSE Manager 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 SUSE Manager Server via ssh (i.e., from the
#    SUSE Manager Server):
#    cd /srv/www/htdocs/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 with your SUSE Manager 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
#   SUSE Manager Server 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 a SUSE Manager Server, the ACTIVATION_KEYS, and ORG_GPG_KEY values"
echo "will probably *not* be set (see below). If this is the case, please do"
echo "the 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)"
echo "    to 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 /srv/www/htdocs/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_filename_of_corporate_public_gpg_key

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

ORG_CA_CERT=rhn-org-trusted-ssl-cert-1.0-1.noarch.rpm
ORG_CA_CERT_IS_RPM_YN=1

USING_SSL=1
USING_GPG=1

REGISTER_THIS_BOX=1

ALLOW_CONFIG_ACTIONS=0
ALLOW_REMOTE_COMMANDS=0

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 clinet 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 -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

INSTALLER=up2date
if [ -x /usr/bin/zypper ] ; then
    INSTALLER=zypper
elif [ -x /usr/bin/yum ] ; then
    INSTALLER=yum
fi
if [ "$INSTALLER" == zypper ]; then
  echo
  echo "CHECKING THE REGISTRATION STACK"
  echo "-------------------------------------------------"
  echo "* check for necessary packages being installed:"
  Z_NEEDED="spacewalk-check spacewalk-client-setup spacewalk-client-tools -
rhncfg-actions rhncfg-client rhncfg-management zypp-plugin-spacewalk"
  Z_MISSING=""
  for P in $Z_NEEDED; do
    rpm -q "$P" || Z_MISSING="$Z_MISSING $P"
  done
  if [ -z "$Z_MISSING" ]; then
    echo "  no packages missing."
  else
    echo "* going to install missing packages:"
    Z_CLIENT_REPO_NAME="susemanager-client-setup"
    Z_CLIENT_REPO_FILE="/etc/zypp/repos.d/${Z_CLIENT_REPO_NAME}.repo"
    if [ ! -f "$Z_CLIENT_REPO_FILE" ]; then
      echo "  adding client software repository $Z_CLIENT_REPO_NAME"
      cat <<EOF >"$Z_CLIENT_REPO_FILE"
[$Z_CLIENT_REPO_NAME]
name=$Z_CLIENT_REPO_NAME
baseurl=http://${HOSTNAME}/pub/repositories/${Z_CLIENT_REPO_NAME}
enabled=1
autorefresh=1
keeppackages=0
gpgcheck=0
EOF
      zypper --non-interactive --gpg-auto-import-keys refresh -
"$Z_CLIENT_REPO_NAME" || exit 1
    fi
    zypper --non-interactive in $Z_MISSING || exit 1
  fi
```

```
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
    # 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 $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
    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
fi

echo
echo "REGISTRATION"
echo "------------"
# Should have created an activation key or keys on the SUSE Manager 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 SUSE Manager Server clients, an"
    echo "           activation key or keys must be created in the SUSE -
Manager Server"
    echo "           web user interface, and the corresponding key or keys -
string"
    echo "           (XKEY,YKEY,...) must be mapped to the ACTIVATION_KEYS -
variable"
    echo "           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 $i")
        done
        [ -f /etc/sysconfig/rhn/allowed-actions/configfiles/all ] || -
files="$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" -
$profilename_opt
    [ -n "$files" ] && rm -f $files
    [ -n "$directories" ] && rmdir $(echo $directories | rev)
    echo
    echo "*** this system should now be registered, please verify ***"
    echo
else
    echo "* explicitely not registering"
fi

echo
echo "OTHER ACTIONS"
echo "------------------------------------------------------"
if [ $FULLY_UPDATE_THIS_BOX -eq 1 ] ; then
    if [ "$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 SUSE Manager Server."
        echo "    Please ensure that rhn-profile-sync in 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 SUSE Manager Server."
        echo "    Please ensure that rhn-profile-sync in 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-"
```

Sample Autoinstallation Script with Tracking

B

```
<?xml version="1.0"?>
<!DOCTYPE profile>
<profile xmlns="http://www.suse.com/1.0/yast2ns" -
xmlns:config="http://www.suse.com/1.0/configns">
  <networking>
    <keep_install_network config:type="boolean">true</keep_install_network>
  </networking>
  <add-on>
    <add_on_products config:type="list">
      <listentry>
        <ask_on_error config:type="boolean">true</ask_on_error>
        <media_url
>http://$redhat_management_server/ks/dist/child/sm-tools-i586/sles11-sp1-i586-ks2</media_url>
        <name>susemanager tools</name>
        <product>SM Tools</product>
        <product_dir>/</product_dir>
      </listentry>
      <listentry>
        <ask_on_error config:type="boolean">true</ask_on_error>
        <media_url
>http://$redhat_management_server/ks/dist/child/clone-sles11-sp1-updates-i586/sles11-sp1-i586-ks2</media_url>
        <name>SLES11 updates</name>
        <product>sles11 updates</product>
        <product_dir>/</product_dir>
      </listentry>
    </add_on_products>
  </add-on>
<scripts>
  <init-scripts config:type="list">
    <script>
      <source><![CDATA[
$SNIPPET('spacewalk/sles_register')
]]>
      </source>
      <filename>ay_suse_manager_register.sh</filename>
      <debug config:type="boolean">true</debug>
    </script>
  </init-scripts>
</scripts>
<software>
  <packages config:type="list">
    <package>spacewalk-client-tools</package>
    <package>spacewalk-client-setup</package>
  </packages>
</software>
<general>
  $SNIPPET('spacewalk/sles_no_signature_checks')
  <mode>
    <confirm config:type="boolean">true</confirm>
  </mode>
</general>
<users config:type="list">
    <user>
      <encrypted config:type="boolean">false</encrypted>
      <fullname>root</fullname>
      <gid>0</gid>
      <home>/root</home>
      <password_settings>
        <expire></expire>
        <flag></flag>
        <inact></inact>
        <max></max>
        <min></min>
        <warn></warn>
      </password_settings>
      <shell>/bin/bash</shell>
      <uid>0</uid>
      <user_password>linux</user_password>
      <username>root</username>
    </user>
</users>
</profile>
```