

# SUSE Linux Enterprise

11

[www.novell.com](http://www.novell.com)

July 16, 2009

Subscription Management Tool Guide



# Subscription Management Tool Guide

**List of Authors:** Florian Nadge

All content is copyright © 2006- 2009 Novell, Inc.

Legal Notice

This manual is protected under Novell intellectual property rights. By reproducing, duplicating or distributing this manual you explicitly agree to conform to the terms and conditions of this license agreement.

This manual may be freely reproduced, duplicated and distributed either as such or as part of a bundled package in electronic and/or printed format, provided however that the following conditions are fulfilled:

That this copyright notice and the names of authors and contributors appear clearly and distinctively on all reproduced, duplicated and distributed copies. That this manual, specifically for the printed format, is reproduced and/or distributed for noncommercial use only. The express authorization of Novell, Inc must be obtained prior to any other use of any manual or part thereof.

For Novell trademarks, see the Novell Trademark and Service Mark list <http://www.novell.com/company/legal/trademarks/tmlist.html>. \* Linux is a registered trademark of Linus Torvalds. All other third party trademarks are the property of their respective owners. A trademark symbol (®, ™ etc.) denotes a Novell trademark; an asterisk (\*) denotes a third party trademark.

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

<b>About this Guide</b>	<b>v</b>
<b>1 SMT Installation</b>	<b>1</b>
1.1 Installation During the Initial Installation Process . . . . .	1
1.2 Installation On Top of an Already Installed System . . . . .	2
1.3 SMT Configuration Wizard . . . . .	3
<b>2 SMT Server Configuration</b>	<b>5</b>
2.1 Activating and Deactivating SMT with YaST . . . . .	6
2.2 Setting NU Credentials with YaST . . . . .	7
2.3 Setting SMT Database Password with YaST . . . . .	9
2.4 Setting E-mail Addresses to Receive Reports with YaST . . . . .	10
2.5 Setting the SMT Job Schedule with YaST . . . . .	10
<b>3 Mirroring Installation and Update Repositories with SMT</b>	<b>13</b>
3.1 Getting Mirror Credentials . . . . .	13
3.2 Managing Software Repositories with SMT . . . . .	15
3.3 The /srv/www/htdocs Structure . . . . .	20
3.4 Using the Test Environment . . . . .	20
3.5 Testing and Filtering Update Repositories with Staging . . . . .	21
<b>4 Managing Repositories with SMT Management</b>	<b>25</b>
4.1 Starting SMT Management Module . . . . .	25
4.2 Viewing and Managing Repositories . . . . .	26
4.3 Staging repositories . . . . .	28
4.4 Checking the Client Status . . . . .	29

<b>5</b>	<b>Managing Client Machines With SMT</b>	<b>31</b>
5.1	Listing Registered Clients . . . . .	31
5.2	Deleting Registrations . . . . .	31
5.3	Manual Registration of Clients at Novell Customer Center . . . . .	32
5.4	Scheduling Periodic Registrations of Clients at Novell Customer Center . . . . .	32
5.5	Lowest Repeat Time . . . . .	33
<b>6</b>	<b>SMT Reports</b>	<b>35</b>
6.1	Report Schedule and Recipients . . . . .	35
6.2	Types of SMT Reports . . . . .	36
6.3	Report Output Formats and Targets . . . . .	37
<b>7</b>	<b>SMT Tools and Configuration Files</b>	<b>39</b>
7.1	Important Scripts and Tools . . . . .	39
7.2	SMT Configuration Files . . . . .	55
7.3	Server Certificates . . . . .	64
<b>8</b>	<b>Configuring Clients to Use SMT</b>	<b>69</b>
8.1	Using Kernel Parameters to Access an SMT Server . . . . .	70
8.2	Configuring Clients with AutoYaST Profile . . . . .	71
8.3	Configuring Clients with the clientSetup4SMT.sh Script . . . . .	72
8.4	Registering Clients Against SMT Test Environment . . . . .	74
	<b>Index</b>	<b>75</b>

# About this Guide

This chapter introduces the SMT installation process and the SMT Configuration Wizard. The *Subscription Management Tool Guide* (SMT) for SUSE Linux Enterprise 11 helps customers to manage their SUSE Linux Enterprise software updates while maintaining corporate firewall policy and regulatory compliance requirements. SMT is a package proxy system that is integrated with the Novell® Customer Center and provides key Novell Customer Center capabilities locally at the customer site. It provides a repository and registration target that is synchronized with the Novell Customer Center, thus maintaining all the capabilities of the Novell Customer Center while allowing a more secure centralized deployment.

## 1 Overview

This guide is divided into the following parts:

### SMT Installation and Configuration

This chapter introduces the SMT installation process and the SM Configuration Wizard. You can install the SMT add-on together with your base program during the installation process or on top of an already installed based system. The SMT Configuration Wizard guides you through the individual installation steps.

### SMT Server Configuration

This chapter introduces the YaST configuration module SMT Server. the YaST graphical interface lets you activate and configure. You can set and configure mirroring credentials, SMT database passwords, and e-mail addresses to send SMT reports, or set the SMT job schedule, and activate or deactivate the SMT service.

### Installation Mirror and Source Updates with YaST.

This chapter introduces the option to mirror the installation and update sources with YaST.

### Management for Client Machines to use SMT

This chapter introduces the option to register client machines on Novell Customer Center. The Client machines must be configured to use SMT.

## SMT Reports

This chapter introduces generated reports based on SMT and Novell Customer Center data. Generated reports contain statistics of all registered machines and products used and of all active, expiring, or missing subscriptions.

## SMT Tools and Configuration Files

This chapter introduces the most important scripts, configuration files and certificates shipped with SMT.

## Configuration for Clients to Use SMT

This chapter introduces the option to configure any client machine to register against SMT and download software updates from there instead of communicating directly with the Novell Customer Center and the NU servers.

# 2 Additional Documentation and Resources

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>.

# 3 Feedback

Several feedback channels are available:

- To report bugs for a product component or to submit enhancement requests, please use <https://bugzilla.novell.com/>. If you are new to Bugzilla, you might find the *Bug Writing FAQs* helpful, available from the Novell Bugzilla home page *Submitting Bug Reports* article—available under [http://en.opensuse.org/Submitting\\_Bug\\_Reports](http://en.opensuse.org/Submitting_Bug_Reports) helpful. Frequently asked questions on reporting bugs are available under [http://en.opensuse.org/Bug\\_Reporting\\_FAQ](http://en.opensuse.org/Bug_Reporting_FAQ).

- We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation and enter your comments there.

## 4 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 shown in uppercase 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.



# SMT Installation

SMT is distributed as an add-on product for SUSE Linux Enterprise Server 11 system. To install it, install the SUSE Linux Enterprise Server 11 base system. You can choose to install the SMT add-on together with your base system during the initial installation process, or you can install the SMT add-on on top of an already-installed base system at any later time.

## 1.1 Installation During the Initial Installation Process

To install SMT add-on together with your base system during the initial installation process, follow these steps:

- 1** Start the SUSE Linux Enterprise Server 11 installation as usual. For more information, see the SUSE Linux Enterprise Server documentation.
- 2** To include the SMT add-on product, check the *Include Add-On Products from Separate Media* option in the *Installation Mode* dialog in the *System Analysis* step, and click *Next*.
- 3** In the next dialog, click *Add* and, if you are installing SMT from a CD medium, select *CD* as the source type. If you are installing from a different source, such as NFS or HTTP, choose the appropriate source type. Click *Next*.

- 4 If you are installing from a CD, insert the SMT add-on product CD. If you are installing from a different source, provide the necessary source. Click *Continue*.
- 5 Confirm the SMT license agreement and click *Next*.
- 6 The SMT add-on product is displayed in the overview. Continue with the installation as usual. Make sure that the *SMT: Subscription Management Tool for SLE* installation pattern is selected automatically and do not remove it.
- 7 A two-step *SMT Configuration Wizard* is shown during the final steps of the installation workflow. Configure SMT as described in [Section 1.3, “SMT Configuration Wizard”](#) (page 3) and continue with the SUSE Linux Enterprise Server installation as usual.

## 1.2 Installation On Top of an Already Installed System

To install SMT on top of an already-installed base system, follow these steps:

- 1 Start YaST and select *Software > Add-On Product*.
- 2 If you are installing SMT from a CD, select *CD* as the source type. If you are installing from a different source, such as NFS or HTTP, choose the appropriate source type. Click *Next*.
- 3 If you are installing from a CD, insert the SMT add-on product CD. If you are installing from a different source, provide the necessary source. Click *Continue*.
- 4 Confirm the SMT license agreement and click *Next*.
- 5 Click *Accept* to install the *SMT: Subscription Management Tool for SLE* pattern.
- 6 The *SMT Configuration Wizard* is launched. See [Section 1.3, “SMT Configuration Wizard”](#) (page 3).

# 1.3 SMT Configuration Wizard

During the installation of SMT the two-step *SMT Configuration Wizard* is launched, regardless of the used installation method. Configure SMT using this wizard. However, you will be able to change the configuration later using the YaST SMT Configuration module.

- 1 The *Enable Subscription Management Tool service (SMT)* option is checked by default. Uncheck it only if you want to disable the installed SMT.

If the firewall is enabled, check *Open Port in Firewall* to allow access to the SMT service from remote computers.

Enter your Novell Customer Center mirroring credentials in *NU User* and *NU Password*. NU stands for Novell Update. If you do not know your Novell Customer Center credentials, refer to [Section 3.1, “Getting Mirror Credentials”](#) (page 13). Test the entered credentials by pressing the *Test* button. SMT will connect to the Customer Center server using the provided credentials and download some testing data.

Enter the e-mail address you have used for the Novell Customer Center registration into *NCC E-mail Used for Registration*.

*Your SMT Server URL* should contain the URL of the SMT server being configured. It is populated automatically.

Press *Next* to continue to the second configuration step.

- 2 For security reasons, enter `smt` as username in the database and select a *SMT Database Password* in the respective fields. The password must be entered.

Enter all e-mail addresses that SMT should send reports to using the *Add* button. You are also able to *Edit* or *Delete* any incorrect or needless addresses.

Press *Next*.

- 3 If the current MySQL root password is empty—as in any freshly installed system—you will be asked to enter a *New MySQL Root Password*.



# SMT Server Configuration

You can use the SMT Server Configuration module for the initial and overall configuration. The SMT Server configures mirroring credentials, sets database passwords and e-mail addresses for SMT reports, sets the SMT job schedule, and activates or deactivate the SMT service.

To configure SMT with the YaST module SMT Server, follow these steps:

- 1 To start SMT Server with text (ncurses) interface, run the `yast smt-server` command as root.

To start the SMT Server with a graphical interface, open YaST Control Center and select *SMT Configuration* in the *Network Services* section.

- 2 To activate SMT, check the *Enable Subscription Management Tool Service (SMT)* option in the *Customer Center Access* tab. If you want to disable SMT, uncheck this option. For more information about activating SMT with YaST, see [Section 2.1, “Activating and Deactivating SMT with YaST”](#) (page 6)
- 3 In the *Customer Center Access* tab, set and test credentials for the NU (Novell Update) service. Correct credentials are necessary to enable mirroring from NU and determine the products that should be mirrored. Also set the e-mail address used for the registration and the URL of your SMT server. For more information, see [Section 2.2, “Setting NU Credentials with YaST”](#) (page 7).
- 4 In the *Database and Reporting* tab, set the password for the SMT user in the MySQL database and enter the e-mail addresses where reports should be sent. For more information, see [Section 2.3, “Setting SMT Database Password with](#)

YaST” (page 9) and [Section 2.4, “Setting E-mail Addresses to Receive Reports with YaST”](#) (page 10).

- 5 In the *Scheduled SMT Jobs* tab, set a schedule of periodic SMT jobs, such as synchronization of updates, Novell Customer Center registration, and SMT report generation. For more information, see [Section 2.5, “Setting the SMT Job Schedule with YaST”](#) (page 10).
- 6 If satisfied with the configuration, click *Finish*. YaST adjusts the SMT configuration and starts or restarts necessary services.

If you want to abort the configuration and cancel any changes, click *Abort*.

---

#### NOTE

When the SMT Server applies configuration changes, it checks for the existence of the common server certificate. If the certificate does not exist, you will be asked if the certificate needs to be created.

---

## 2.1 Activating and Deactivating SMT with YaST

YaST provides an easy way to activate or deactivate the SMT service. To activate SMT with YaST, follow these steps:

- 1 Open the *Customer Center Access* tab of the SMT Server .
- 2 Check the *Enable Subscription Management Tool service (SMT)* option.

---

#### NOTE

If not already configured, mirroring credentials should be configured before activating SMT. For more information about how to set mirroring credentials with YaST, see [Section 2.2, “Setting NU Credentials with YaST”](#) (page 7).

---

- 3 Click *Finish* to apply the changes and leave the SMT Server.

To deactivate SMT with YaST, follow these steps:

- 1 Open the *Customer Center Access* tab of the SMT Server.
- 2 Uncheck the *Enable Subscription Management Tool service (SMT)* option.
- 3 Click *Finish* to apply the changes and leave the SMT Server .

When activating SMT, the following important operations are performed by YaST:

- The Apache configuration is changed by creating symbolic links in the `/etc/apache2/conf.d/` directory. Links to the `/etc/smt.d/nu_server.conf` and `/etc/smt.d/smt_mod_perl.conf` files are created there.
- The Apache Web server is started (or reloaded if already running).
- The MySQL server is started (or reloaded if already running). If it does not exist, `smt` user and all necessary tables in the database are created.
- The schema of the SMT database is checked. If the database schema is obsolete, the SMT database is upgraded to conform to the current schema.
- Cron is adjusted by creating a symbolic link in the `/etc/cron.d/` directory. A link to the `/etc/smt.d/novell.com-smt` file is created there.

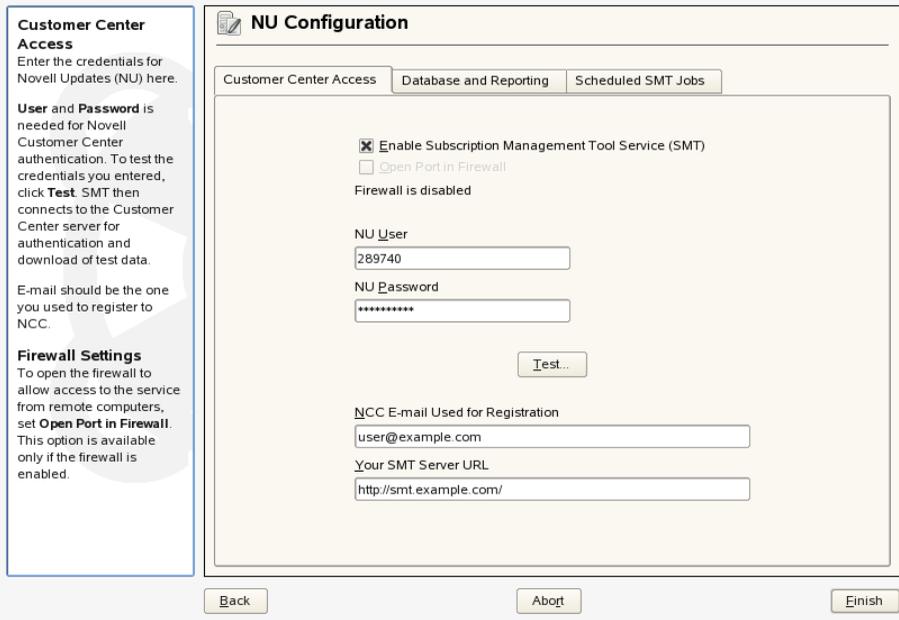
When deactivating SMT, the following important operations are performed by YaST:

- Symbolic links created upon SMT activation in the `/etc/apache2/conf.d/` and `/etc/cron.d/` directories are deleted.
- The Cron, Apache Web and MySQL servers are reloaded. Neither Apache nor MySQL are stopped, because they may be used for other purposes than the SMT service.

## 2.2 Setting NU Credentials with YaST

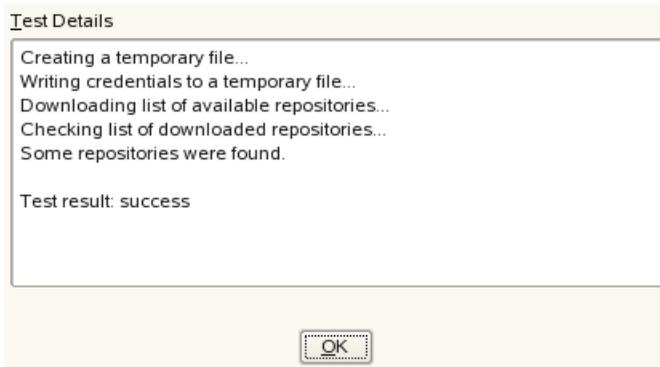
YaST provides a comfortable interface to set and test NU credentials and the URL of the NU service. To do so, follow these steps:

**Figure 2.1** *Setting NU Credentials with YaST*



- 1 Open the *Customer Center Access* tab of the SMT Server . If the credentials have been already set with YaST or the `/etc/smt.conf` configuration file, they appear in the dialog. Otherwise, the *NU User* and *NU Password* fields are blank.
- 2 If you do not have your credentials, visit Novell Customer Center to obtain them. For more details, see [Section 3.1, “Getting Mirror Credentials”](#) (page 13).
- 3 Enter your NU username in *NU User* and the corresponding password in *NU Password*.
- 4 Press *Test* to check the credentials. YaST will try to download a list of available repositories with the provided credentials. If the test succeeds, the last line of the test results will read `Test result: success`. If the test fails, check the provided credentials and try again.

**Figure 2.2** *Successful Test of NU Credentials*



- 5 Enter the *NCC E-mail Used for Registration*. This should be the address you used to register to Novell Customer Center.

Enter *Your SMT Server URL* if it has not been detected automatically.

- 6 Press *Finish* or continue with other configurations.

## 2.3 Setting SMT Database Password with YaST

For security reasons, SMT uses its own user in the database. YaST provides a comfortable interface for setting up or changing the SMT database password. To set or change the SMT database password with YaST follow these steps:

- 1 Open the *Database and Reporting* tab of the SMT Server module.
- 2 Enter the *SMT Database Password for SMTUser*. Confirm the password by reentering it and press *Finish*, or continue with other configurations.

## 2.4 Setting E-mail Addresses to Receive Reports with YaST

YaST SMT provides a comfortable interface for setting up a list of e-mail addresses to which SMT reports will be sent. To edit this list of addresses, follow these steps:

- 1 Open the *Database and Reporting* tab of the SMT Server .
- 2 The list of e-mail addresses is shown in the table. You can *Add*, *Edit*, or *Delete* addresses with the relevant buttons.
- 3 Press *Finish* or continue with other configurations.

The comma-separated list of addresses for SMT reports is written to the `reportEmail` option of the `/etc/smt.conf` configuration file.

## 2.5 Setting the SMT Job Schedule with YaST

The SMT Server Configuration module provides a comfortable interface to schedule periodical SMT jobs. YaST uses `cron` to schedule configured jobs. If needed, `cron` can be used directly. Five types of periodical jobs can be set:

### Synchronization of Updates

Synchronizes with Novell Customer Center, updates repositories, and downloads new updates.

### Report Generation

Generates and sends SMT Subscription Reports to addresses defined in [Section 2.4, “Setting E-mail Addresses to Receive Reports with YaST”](#) (page 10).

### NCC Registration

Registers all clients to Novell Customer Center that are not already registered or that changed their data since the last registration.

## Job Que Cleanup

Cleans up cued jobs. function unknown

## Uploads Support Configs

Uploads support configs, only when the SMT -Support package is installed.

**Figure 2.3** Setting SMT Job Schedule with YaST

**Editing a SMT Scheduled Job**

Frequency: Periodically      Job to Run: NCC Registration

Job Start Time

Day of the Week: Sunday      Day of the Month: 1

Every H-th Hour: 0      Every M-th Minute: 15

OK      Cancel

To configure the schedule of SMT jobs with YaST, follow these steps:

- 1 Open the *Scheduled SMT Jobs* tab of the SMT Server . The table contains a list of all scheduled jobs, their type, frequency, date, and time to run. You can add, delete or edit these scheduled events.
- 2 If you want to add a scheduled SMT job, press *Add*. The *Adding New SMT Scheduled Job* dialog opens.

Choose the synchronization job to schedule. You can choose between *Synchronization of Updates*, *Report Generation*, *NCC Registration*, *Job Queue Cleanup*, and *Uploading Support Configs*.

Choose the *Frequency* of the new scheduled SMT job. Jobs can be performed *Daily*, *Weekly*, *Monthly*, or *Periodically* (every n-th hour or every m-th minute).

Set the *Job Start Time* by entering *Hour* and *Minute*. In case of periodical frequency, enter the relevant periods. For weekly and monthly schedules, select *Day of the Week* or *Day of the Month*.

Press *Add*.

- 3** If you want to edit a scheduled SMT job (for example, change its frequency, time, or date), select the job in the table and press *Edit*. Then change the desired parameters and press *OK*.
- 4** If you want to cancel a scheduled job and delete it from the table, select the job in the table and press *Delete*.
- 5** Press *Finish* to apply the settings and quit the SMT Server, or continue with other configurations.

# Mirroring Installation and Update Repositories with SMT

# 3

With SMT you can mirror the installation and update repositories locally. This allows you to bypass per-machine downloads and the bandwidth use that goes with it.

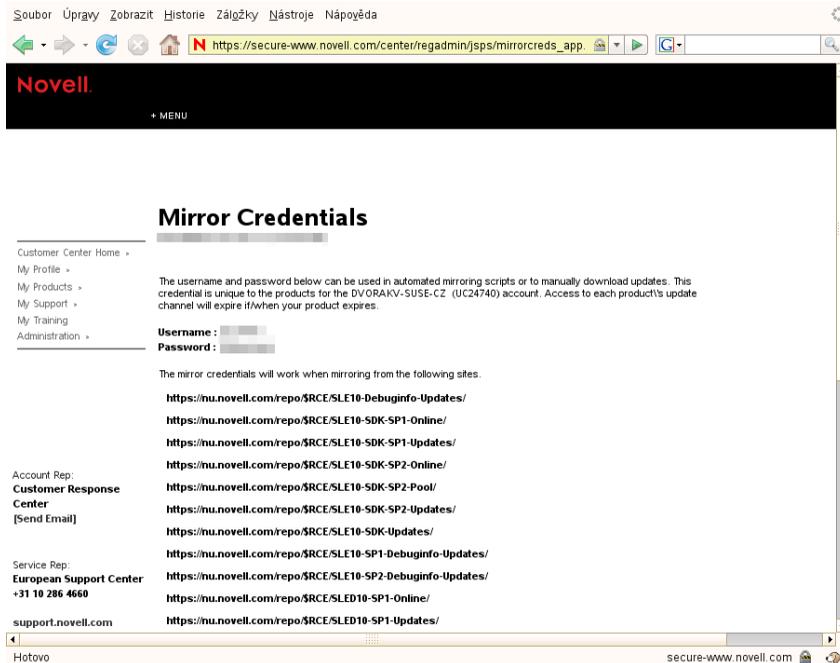
## 3.1 Getting Mirror Credentials

Before you can create local mirrors of the repositories, you need to have proper mirror credentials. You can get these credentials from the Novell Customer Center with these steps:

- 1 Visit Novell Customer Center at <http://www.novell.com/center> and log in.
- 2 Click on *My Products*. The list of product families is shown.
- 3 Expand any product family by clicking on its name. You can also expand all product families by clicking on the icon showing the arrow with two converse arrowheads (with the *Expand All Product Families* tooltip). Products in the expanded families are shown.
- 4 Double click on any specific product in the list to show detailed information about the product.
- 5 In the *Downloads* section, click on the *Mirror Credentials* link.

- 6 If necessary (for example if you are accessing the page for the first time), click on the *Generate* button.
- 7 The credentials and mirror sites will be listed. These values are the same for all users and subscriptions for a specific company.

**Figure 3.1** *NU Credentials in Novell Customer Center*



The obtained credentials should be set in the SMT Server module or manually written in the `/etc/smt.conf` file. For more information about the `/etc/smt.conf` file, see [Section 7.2.1, “/etc/smt.conf”](#) (page 55)

To request a merge, the customer or their sales rep send an email [emea\\_pic@novell.com](mailto:emea_pic@novell.com) [mailto:emea\_pic@novell.com] (for EMEA-based customers only) with the applicable customer and site IDs. The EMEA PIC team will verify the records. The contact for NALAAP is [cmf@novell.com](mailto:cmf@novell.com) [mailto:cmf@novell.com]

## 3.2 Managing Software Repositories with SMT

This section describes tools and procedures for viewing information about software repositories available through SMT, configuring these repositories and setting new custom repositories. For details on the newly designed SMT Management module, see [Chapter 4, \*Managing Repositories with SMT Management\*](#) (page 25)

### 3.2.1 Updating the local SMT database

The local SMT database needs to be updated periodically with the information downloaded from Novell Customer Center. These periodical updates can be configured with the SMT Management module, as described in [Section 2.5, “Setting the SMT Job Schedule with YaST”](#) (page 10).

To update the SMT database manually, use the `smt-ncc-sync` command. For more information about the `smt-ncc-sync` command, see [Section “smt-ncc-sync”](#) (page 47).

### 3.2.2 Enabled Repositories and Repositories that Can Be Mirrored

The database installed with SMT contains information about all software repositories available on Novell Customer Center. However, the used mirror credentials determine which repositories can really be mirrored. For more information about getting and setting mirror credentials, see [Section 3.1, “Getting Mirror Credentials”](#) (page 13).

The mirrorability of repositories is determined by retrieving <https://nu.novell.com/repo/repoindex.xml> using the provided mirror credentials. Repositories that can be mirrored have the `MIRRORABLE` flag set in the repositories table in the SMT database.

The fact that a repository can be mirrored does not mean that it has to be mirrored. Only repositories with the `DOMIRROR` flag set in the SMT database will be mirrored.

For more information about setting which repositories should be mirrored, see [Section 3.2.4, “Selecting Repositories to be Mirrored”](#) (page 17).

## 3.2.3 Getting Information About Repositories

Use the `smt-repositories` command to list available software repositories and additional information. Using this command without any options lists all available repositories, including repositories that cannot be mirrored. In the first column, the enabled repositories (repositories set to be mirrored) are marked with `Yes`. Disabled repositories are marked with `No`. The other columns show ID, type, name, target, and description of the listed repositories. The last column shows whether the repository can be mirrored.

Use the `-verbose` option, to get additional information as the source URL of the repository and the path it will be mirrored to.

The repository listing can be limited to only repositories that can be mirrored or to enabled repositories. To list only repositories that can be mirrored, use the `-m` or `--only-mirrorable` option: `smt-repositories -m`.

To list only enabled repositories, use the `-o` or `--only-enabled` option: `smt-repositories -o`.

### **Example 3.1** *Listing All Enabled Repositories*

```
tux:~ # smt-repositories --o
```

Mirror?	ID	Type	Name	Target	Description	Can be Mirrored
Yes	6	nu	SLES10-SP2-Online	sles-10-i586	SLES10-SP2-Online for sles-10-i586	Yes
Yes	7	nu	SLES10-SP2-Online	sles-10-ia64	SLES10-SP2-Online for sles-10-ia64	Yes
Yes	8	nu	SLES10-SP2-Online	sles-10-ppc	SLES10-SP2-Online for sles-10-ppc	Yes
Yes	9	nu	SLES10-SP2-Online	sles-10-s390x	SLES10-SP2-Online for sles-10-s390x	Yes
Yes	10	nu	SLES10-SP2-Online	sles-10-x86_64	SLES10-SP2-Online for sles-10-x86_64	Yes
Yes	11	nu	SLES10-SP2-Updates	sles-10-i586	SLES10-SP2-Updates for sles-10-i586	Yes
Yes	12	nu	SLES10-SP2-Updates	sles-10-ia64	SLES10-SP2-Updates for sles-10-ia64	Yes
Yes	13	nu	SLES10-SP2-Updates	sles-10-ppc	SLES10-SP2-Updates for sles-10-ppc	Yes
Yes	14	nu	SLES10-SP2-Updates	sles-10-s390x	SLES10-SP2-Updates for sles-10-s390x	Yes
Yes	15	nu	SLES10-SP2-Updates	sles-10-x86_64	SLES10-SP2-Updates for sles-10-x86_64	Yes

You can also list only repositories with a particular name or show information about a repository with a particular name and target. To list repositories with a particular name, use the `smt-repositories repository_name` command. To show information

about a repository with a particular name and target, use the `smt-repositories repository_name target` command.

## 3.2.4 Selecting Repositories to be Mirrored

Only enabled repositories can be mirrored. In the database, the enabled repositories have the `DOMIRROR` flag set. Repositories can be enabled or disabled using the `smt-repositories` script.

To enable one or more repositories, follow these steps:

- 1 If you want to enable all repositories that can be mirrored or just choose one repository from the list of all repositories, run the `smt-repos -e` command.

You are able to limit the list of repositories by using the relevant options. To limit the list to only repositories that can be mirrored, use the `-m` option: `smt-repositories -m -e`. To limit the list to only repositories with a particular name, use the `smt-repositories -e repository_name` command. To list only a repository with a particular name and target, use the command `smt-repositories -e repository_name target`.

If you want to enable all repositories belonging to a certain product, use the `--enable-by-prod` or `-p` option followed by the name of the product and, optionally, its version, architecture, and release: `smt-repositories -p product[, version[, architecture[, release]]]`. For example, to enable all repositories belonging to SUSE Linux Enterprise Server 10 SP2 for PowerPC architecture, use the `smt-repositories -p SUSE-Linux-Enterprise-Server-SP2,10,ppc` command. The list of known products can be obtained with the `smt-list-products` command.

- 2 If more than one repository is listed, choose the one you want to enable by specifying its ID listed in the repository table and pressing Enter. If you want to enable all the listed repositories, use `a` and press Enter.

To disable one or more repositories, follow these steps:

- 1 If you want to disable all enabled repositories or just choose one repository from the list of all repositories, run the `smt-repositories -d` command.

If you want to choose the repository to be disabled from a shorter list, or if you want to disable all repositories from a limited group, you can use any of the available options to limit the list of the repositories. To limit the list to only enabled repositories, use the `-o` option: `smt-repositories -o -d`. To limit the list to only repositories with a particular name, use the `smt-repositories -d repository_name` command. To list only a repository with a particular name and target, use the `smt-repositories -d repository_name target` command.

- 2 If more than one repositories is listed, choose which one you want to disable by specifying its ID listed in the repository table shown and pressing Enter. If you want to disable all the listed repositories, use `a` and press Enter.

## 3.2.5 Mirroring Custom Repositories

You can mirror repositories that are not available at the Novell Customer Center —custom repositories— using SMT. Use the `smt-setup-custom-repos` script for this purpose. Custom repositories can also be deleted.

To set up a custom repository to be available through SMT, follow these steps:

- 1 If you do not know the ID of the product the new repositories should belong to, use `smt-list-products` to get the ID. For the description of the `smt-list-products`, see [Section “smt-list-products”](#) (page 45).
- 2 Run the `smt-setup-custom-repos --productid product_id --name repository_name --exturl repository_url` command, where `product_id` is the ID of the product the repository belongs to, `repository_name` represents the name of the repository and `repository_url` is the URL the repository is available at. In case the added repository needs to be available for more than one product, specify the IDs of all products that should use the added repository.

For example, to set `My` repository available at `http://example.com/My_repository` to the products with the IDs 423, 424, and 425, use the following command:

```
smt-setup-custom-repositories --productid 423
```

```
--productid 424 --productid 425 --name 'My_repository'  
--exturl 'http://example.com/My_repository'.
```

---

### NOTE: Mirroring Unsigned Repositories

In its default configuration, SUSE Linux Enterprise 10 does not allow the use of unsigned repositories. Therefore, if you want to mirror unsigned repositories and use them on client machines, you have to allow this explicitly by executing the following command on the client machines:

```
rug set security-level checksum
```

---

To remove an already-set custom repository from the SMT database, use `smt-setup-custom-repositories --delete ID`, where `ID` represents the ID of the repository to be removed.

## 3.2.6 Mirroring SUSE Linux Enterprise Server 9 Repositories

For mirroring old style update repositories which were used for SUSE Linux Enterprise Server 9 and similar products, use a special command: `smt-mirror-sle9`. This script mirrors from the `https://you.novell.com` server.

The `smt-mirror-sle9` script does not store information about repositories to be mirrored in the SMT database. It only uses the configuration from the `/etc/smt.conf` file. The configuration of `smt-mirror-sle9` is described in [Section “smt-mirror-sle9 Sections of /etc/smt.conf”](#) (page 60).

The `smt-mirror-sle9` command uses `wget` to mirror repositories. Therefore, you can exclude anything you do not want to be mirrored by adding the `exclude_directories` option to the `/root/.wgetrc` configuration file. For more information about `wget` and `/root/.wgetrc`, see `man 1 wget`.

## 3.3 The /srv/www/htdocs Structure

The path to the directory containing the mirror is set by the `MirrorTo` option in the `/etc/smt.conf` configuration file. For more information about `/etc/smt.conf`, see [Section 7.2.1, “/etc/smt.conf”](#) (page 55). If the `MirrorTo` option is not set to the Apache `htdocs` directory `/srv/www/htdocs/`, links should be created manually like this: `/srv/www/htdocs/repo/$RCE` should point to `/MirrorTo/repo/$RCE/`, `/srv/www/htdocs/repo/RPMMD` should point to `/MirrorTo/repo/RPMMD/`, `/srv/www/htdocs/repo/testing` should point to `/MirrorTo/repo/full/` and `/srv/www/htdocs/repo/full` should point to `/MirrorTo/repo/full/`. Here, `/MirrorTo` is the path set in the `MirrorTo` option.

For example, if the `MirrorTo` is set to `/space/MIRRORDATA/`:

```
srv64:~ # 1 /srv/www/htdocs/repo/
total 16
drwxr-xr-x 2 root root 128 2008-01-18 14:00 ./
drwxr-xr-x 4 root root 128 2008-01-17 17:14 ../
lrwxrwxrwx 1 root root 27 2008-01-11 15:17 $RCE ->
/space/MIRRORDATA/repo/$RCE/
-rw-r--r-- 1 root root 14854 2008-01-21 12:36 repoindex.xml
lrwxrwxrwx 1 root root 26 2008-01-11 15:37 RPMMD ->
/space/MIRRORDATA/repo/RPMMD/
```

The links can be created using the `ln -s` commands. For example: `ln -s '/MirrorTo/repo/$RCE/' '/srv/www/htdocs/repo/$RCE'`

## 3.4 Using the Test Environment

You can mirror repositories to a test environment instead of the production environment. The test environment can be used with a limited number of client machines before the tested repositories are moved to the production environment. The test environment can be run on the main SMT server.

The testing environment uses the same structure as the production environment, but it is located in the `/srv/www/htdocs/repos/testing/` subdirectory.

To mirror a repository to the testing environment, you can use the `Staging` tab in the SMT Management module, or the command `smt-staging`.

To register a client in the testing environment, modify the `/etc/suseRegister.conf` on the client machine by setting:

```
register = command=namespace=testing
```

To move the testing environment to the production environment, manually copy or move it using the `cp -a` or `mv` command.

You can enable "staging" for a repository in the `Repositories` tab of the SMT Management module or with the `smt repos` command. The mirroring happens automatically to `repo/full/`.

If you have a SLE11-based Update repository with patches, SMT tools can help you with the management. With these tools you can select patches and create a snapshot and put it into `repo/testing/`. After tests are finished you can put the content of `repo/testing` into the production area `/repo`.

SLE10-based Update repositories are not supported by SMT tools. Not all of these repositories support selective staging. In this case you need to mirror the complete package.

Recommended work flow:

```
repo=>repo/full, repo/full => repo/testing, repo/testing => repo/
```

## 3.5 Testing and Filtering Update Repositories with Staging

SMTStaging lets you test repositories on any clients before moving them to the production environment. You can select new update repositories manually to be installed on clients.

For Staging you can either use the `smt-staging` command, or use the SMT Management module. For more details, see [Chapter 4, \*Managing Repositories with SMT Management\*](#) (page 25)

Repositories with enabled staging are mirrored to the `/MirrorTo/repos/full` subdirectory. This subdirectory is usually not used by your clients. Incoming new updates are not automatically visible to the clients before you get a chance to test them. Later you can

generate a testing environment of this repository, which goes to `/mirrorTo/repos` directory

If you have a SLE11 based Update repository with patches, SMT tools can help you with the management. With these tools you can select patches and create a snapshot and put it into `repo/testing/`. After tests are finished you can put the content of `repo/testing` into the production area `/repo`.

SLE10 based Update repositories are not supported by SMT tools. Not all of these repositories support selective staging. In this case you need to mirror the complete package.

### >Enabling Staging

To enable or disable the staging use the `smt repos` command:

```
smt repos --enable staging
```

```
smt repos -s.
```

You can enable the required repositories by entering the ID number or by entering the name and target. If you want to enable all repositories enter *a*.

### Generating Testing and Production Snapshots

To create the testing repository enter the `$ smt staging createrepo Repository_ID --testing` . Now, you can test the installation and functionality of the patches in testing clients. If no problems are discovered during testing, create the production repository by entering `$ smt staging createrepo Repository_ID --production`.

### Filtering Patches

You can allow or forbid all or selected patches with the `allow` or `forbid` commands by their ID or Category:

```
$ smt staging forbid --patch ID
```

```
$ smt staging forbid --category Categoryname
```

## Signing Changed Repositories

If you filter one or more patches from a repository, the original signature becomes invalid. The repository needs to be signed again. The `smt createrepo` command takes care of that automatically if you configure the SMT server.

In order to enable signing of changed metadata, the admin needs to generate new signing key. This can be done with `gpg` like this:

```
$ mkdir $some-dir
```

```
$ sudo mv $somedir /var/lib/smt/.gnupg
```

```
$ sudo chown smt.users -R /var/lib/smt/.gnupg
```

```
$ sudo chmod go-rwx -R /var/lib/smt/.gnupg
```

```
$ sudo chown smt.users -R /var/lib/smt/.gnupg
```

```
$ sudo chmod go-rwx -R /var/lib/smt/.gnupg
```

Then, the ID of the newly generated key must be written into `smt.conf`, option `'signingKeyID'`.

At this point the clients do not know about this new key. In order to import the new key to clients during their registration, the following can be done:

```
$ sudo -u smt gpg --homedir /var/lib/smt/.gnupg --export  
-a 48730C7A \> $MirrorTo/repo/keys/smt-signing-key.key
```

## Registering Clients in the Testing Environment

To register a client in the testing environment, modify the `/etc/suseRegister.conf` on the client machine by setting:

```
register = command=namespace=testing
```



# Managing Repositories with SMT Management

You can use the SMT Management module for day-to-day management. SMT Management enables and disables the mirroring of repositories, the staging flag for repositories, and performs the mirroring and staging in the GUI.

## 4.1 Starting SMT Management Module

SMT Management is a YaST module. To start the module, do one of the following:

- Start YaST and select *Network Services*, then *SMT Server Management*.
- Open your favorite terminal application and enter `yast2 smt` in the command line as root.

SMT Management application window opens with the *Repositories* tab active.

**Figure 4.1** List of Repositories

Name	Target	Mirroring	Staging	Mirrored	Description
All-Down-SLE10-SP2	-	✓	✓	2009-07-14 09:35:04	All-Down-SLE10-SP2
All-Down-SLE11	-	✓	-	2009-07-14 05:35:03	All-Down-SLE11
All-Downs-SPI	-	✓	-	2009-07-08 09:37:37	All-Downs
All-Downs-SPI-Online	sles-10-ppc	✓	-	2009-07-14 05:35:04	All-Downs
SLE10-SDK-SPI-Online	sles-10-ppc	-	-	-	SLE10-SDK-SPI-Online for sles-10-ppc
SLE10-SDK-SPI-Online	sles-10-ia64	-	-	-	SLE10-SDK-SPI-Online for sles-10-ia64
SLE10-SDK-SPI-Online	sles-10-686	-	-	-	SLE10-SDK-SPI-Online for sles-10-686
SLE10-SDK-SPI-Online	sles-10-686	-	-	-	SLE10-SDK-SPI-Online for sles-10-686
SLE10-SDK-SPI-Online	sles-10-686_64	✓	✓	2009-07-14 02:00:10	SLE10-SDK-SPI-Online for sles-10-686_64
SLE10-SDK-SPI-Online	sles-10-3200x	-	-	-	SLE10-SDK-SPI-Online for sles-10-3200x
SLE10-SDK-SPI-Online	sles-10-686_64	-	-	-	SLE10-SDK-SPI-Online for sles-10-686_64
SLE10-SDK-SPI-Updates	sles-10-ppc	-	-	-	SLE10-SDK-SPI-Updates for sles-10-ppc
SLE10-SDK-SPI-Updates	sles-10-ia64	-	-	-	SLE10-SDK-SPI-Updates for sles-10-ia64
SLE10-SDK-SPI-Updates	sles-10-686	-	-	-	SLE10-SDK-SPI-Updates for sles-10-686
SLE10-SDK-SPI-Updates	sles-10-686	✓	✓	2009-07-14 05:08:03	SLE10-SDK-SPI-Updates for sles-10-686
SLE10-SDK-SPI-Updates	sles-10-686_64	-	-	-	SLE10-SDK-SPI-Updates for sles-10-686_64
SLE10-SDK-SPI-Updates	sles-10-686_64	-	-	-	SLE10-SDK-SPI-Updates for sles-10-686_64
SLE10-SDK-SPI-Updates	sles-10-3200x	-	-	-	SLE10-SDK-SPI-Updates for sles-10-3200x
SLE10-SDK-SPI-Online	sles-10-ppc	-	-	-	SLE10-SDK-SPI-Online for sles-10-ppc
SLE10-SDK-SP2-Online	sles-10-ia64	-	-	-	SLE10-SDK-SP2-Online for sles-10-ia64
SLE10-SDK-SP2-Online	sles-10-686_64	-	-	-	SLE10-SDK-SP2-Online for sles-10-686_64
SLE10-SDK-SP2-Online	sles-10-686_64	-	-	-	SLE10-SDK-SP2-Online for sles-10-686_64
SLE10-SDK-SP2-Online	sles-10-686	-	-	-	SLE10-SDK-SP2-Online for sles-10-686
SLE10-SDK-SP2-Online	sles-10-686	-	-	-	SLE10-SDK-SP2-Online for sles-10-686
SLE10-SDK-SP2-Online	sles-10-ppc	-	-	-	SLE10-SDK-SP2-Online for sles-10-ppc
SLE10-SDK-SP2-Post	sles-10-686	-	-	-	SLE10-SDK-SP2-Post for sles-10-686
SLE10-SDK-SP2-Post	sles-10-686	-	-	-	SLE10-SDK-SP2-Post for sles-10-686
SLE10-SDK-SP2-Post	sles-10-3200x	-	-	-	SLE10-SDK-SP2-Post for sles-10-3200x
SLE10-SDK-SP2-Post	sles-10-686_64	-	-	-	SLE10-SDK-SP2-Post for sles-10-686_64
SLE10-SDK-SP2-Post	sles-10-ia64	-	-	-	SLE10-SDK-SP2-Post for sles-10-ia64
SLE10-SDK-SP2-Post	sles-10-686_64	-	-	-	SLE10-SDK-SP2-Post for sles-10-686_64
SLE10-SDK-SP2-Post	sles-10-ppc	-	-	-	SLE10-SDK-SP2-Post for sles-10-ppc
SLE10-SDK-SP2-Updates	sles-10-686	-	-	-	SLE10-SDK-SP2-Updates for sles-10-686
SLE10-SDK-SP2-Updates	sles-10-686_64	-	-	-	SLE10-SDK-SP2-Updates for sles-10-686_64

## 4.2 Viewing and Managing Repositories

In the *Repositories* tab, you can see the list of all available package repositories for SMT. For each repository, the list shows the repository's name, target product and architecture, mirroring and staging flag, date of last mirroring, and a short description. You can sort the list by clicking on the relevant column's header, and scroll the list items by the scroll's on the window's right side.

### 4.2.1 Filtering Repositories

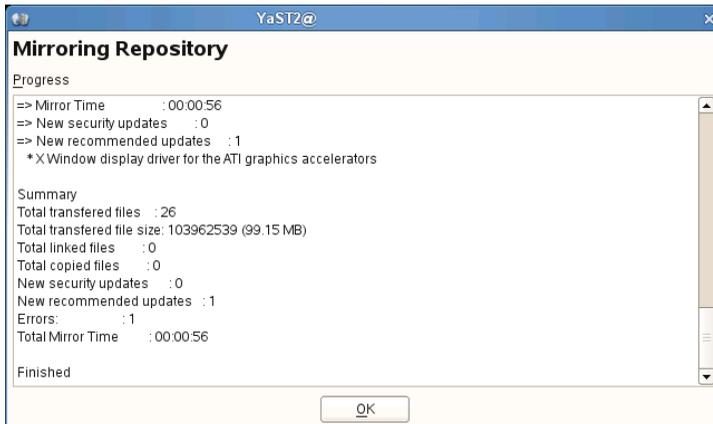
You can also filter out groups of repositories with the *Filter* drop-down list in the upper left part of the window. The filter list items are collected and assembled dynamically from the first word of the repositories' names. If you use a filter to limit the number of displayed repositories, you can always go back and view all of them by selecting *All* from the *Filter* drop-down list.

## 4.2.2 Mirroring Repositories

Before you can start to offer package repositories, you need to create a local mirror of their packages. To do this, follow the procedure:

- 1 From the list, select the line containing the name of the repository you want to mirror.
- 2 Click on the selected line highlighting it.
- 3 Click on the *Toggle Mirroring* button in the lower left part of the window. In the *Mirroring* column of the selected repository, a check mark appears. If the repository was already selected for mirroring before, the check mark will disappear, and the repository will not be mirrored anymore.
- 4 Hit the *Mirror Now* button and the repository will be mirrored immediately.
- 5 A pop-up window appears with the information about mirroring status and result.
- 6 Click OK and the original window with the list of repositories will be refreshed.

**Figure 4.2** *Status of Mirroring Process*



## 4.3 Staging repositories

After the mirroring is finished, you can stage the mirrored repositories. In SMT, *staging* is a process where you create either testing or production repositories based on the mirrored ones. The testing repository helps you examine the repository and its packages before you make them available in a production environment. To make repositories available for staging, do the following:

- 1 From the repository list, select the line containing the name of the repository you want to manage.
- 2 Click on the selected line, highlighting it.
- 3 Click on the *Toggle Staging* button in the lower left part of the window next to the *Toggle Mirroring* button. In the *Staging* column of the selected repository, a check mark appears. If the repository was already selected for staging before, the check mark will disappear, and the repository will not be available for staging.
- 4 Repeat steps 1 to 3 for all the directory whose staging flag you wish to change.

---

### **IMPORTANT: Toggle Staging Button Not Active**

You can only stage the repositories that were selected for mirroring previously. If it is not the case, the *Toggle Staging* button will not be active.

---

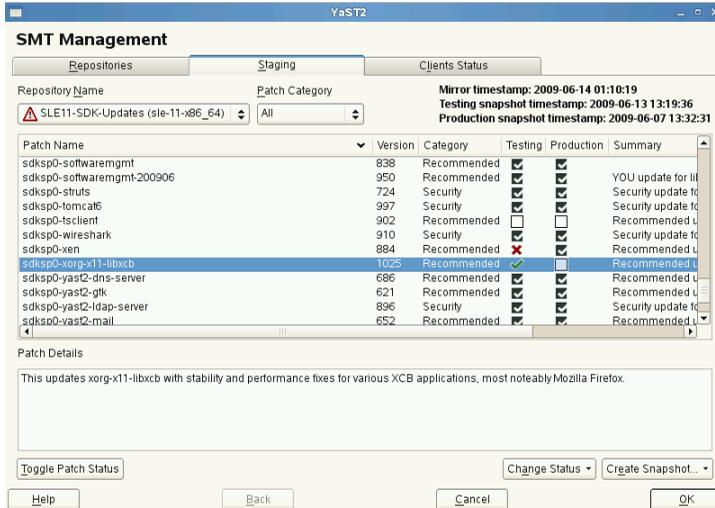
Once you mirror the repositories and make them available for staging, click on the *Staging* tab. In the upper left part of the window, there is a *Repository Name* drop-down list of all repositories which are available for staging. Select the one you want to stage and a list of packages of this repository appears below. Information about the patch name, its version and category, testing and production flags, and a short summary is available for each patch.

Next to the *Repository Name* drop-down list, there is a *Patch Category* filter. It helps you list only the patches that belong to one of the predefined categories.

If the selected repository allows for patch filtering, you can toggle the status flag for individual patches. Do so by clicking on the *Toggle Patch Status* button in the lower left part of the window.

Before creating a repository of packages that are available in the *production* environment, you need to create and test the *testing* repository. Click on the *Create Snapshot* drop-down menu and select *From Full Mirror to Testing* menu item. A small pop-up window appears informing you about the staging process. After the testing repository snapshot is created, the relevant check marks in the *Testing* column will be displayed.

**Figure 4.3** *Testing Snapshot Created*



After you have examined the newly created testing repository, you can safely create a production one. Click on the *Create Snapshot* drop-down menu and select *From Testing to Production* menu item. A small pop-up window appears informing you about the linking the testing repository to the production one. After the production snapshot is created, the relevant check marks in the *Production* column will be displayed. Also, a green check mark appears in the *Repository Name* drop-down list.

## 4.4 Checking the Client Status

The third tab called *Clients Status* contains the status information about all the clients that use the repositories on your SMT server. It is divided into two main parts: the list of the clients and the detailed information.

You can read the client's host name, the date and time of the last network contact with the SMT server, and its update status. The update status can be one of the following:

#### Up-to-date

The client packages are updated to their last version available in the production repository.

#### Updates available

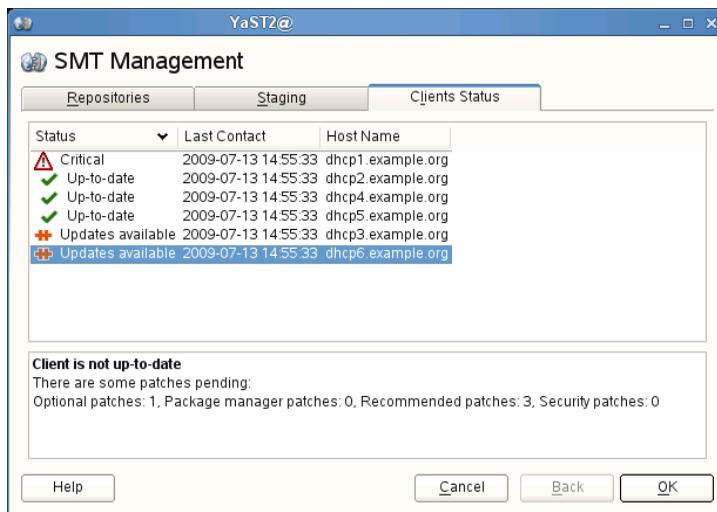
This status means that there are updates available for the client that either optional or recommended.

#### Critical

Either security patches or package manager patches are available for the client.

In the lower part of the window, more detailed information about the highlighted client is available. It usually consists of extended status information and detailed information about the number and types of available updates.

**Figure 4.4** *Clients Status*



# Managing Client Machines With SMT

SMT lets you register and manage client machines on Novell Customer Center. Client machines must be configured to use SMT. For information about configuring clients to use SMT, see [Chapter 8, \*Configuring Clients to Use SMT\*](#) (page 69).

## 5.1 Listing Registered Clients

To list SMT-registered client machines, use the `smt-list-registrations` command. The following information is listed for each client: its *Unique ID*, *Hostname*, date and time of *Last Contact* with the SMT server, and the Software *Product* the client uses.

## 5.2 Deleting Registrations

To delete a registration from SMT and Novell Customer Center, use the `smt-delete-registrations -g Client_ID` command. To delete multiple registrations, the option `-g` can be used several times.

The ID of the client machine to be deleted can be determined from the output of the `smt-list-registrations` command.

## 5.3 Manual Registration of Clients at Novell Customer Center

The `smt-register` command registers clients at Novell Customer Center. All clients that are currently not registered or whose data has changed since the last registration are registered.

To register clients whose registration has failed, use the `--reseterror` option. This option resets the NCC registration error flag and tries to submit failed registrations again.

## 5.4 Scheduling Periodic Registrations of Clients at Novell Customer Center

SMT module allows for the easy scheduling of client registrations. In the default configuration, registrations are scheduled to repeat every 15 minutes. To create or modify a new registration schedule, follow these steps:

- 1 Start YaST *SMT Configuration* module (`yast2 smt`).
- 2 Go to the *Scheduled SMT Job*.
- 3 Select any *NCC Registration* job and click *Edit* if you want to change its schedule.

To create a new registration schedule, click *Add* and select *NCC Registration as Job to Run*.

- 4 Choose the *Frequency* of the scheduled SMT job. You can perform jobs *Daily*, *Weekly*, *Monthly*, or *Periodically* (every n-th hour or every m-th minute).

Set the *Job Start Time* by entering *Hour* and *Minute*, or, in case of periodical frequency, the relevant periods. For weekly and monthly schedules, select the *Day of the Week* or the *Day of the Month* the mirroring should occur.

---

**NOTE: Lowest Registration Frequency**

Do not set the frequency lower than 10 minutes, because the maximal value of the `rndRegister` is 450 (7.5 minutes). If the frequency is lower, it may occur that the started process is still sleeping when the proceeding process starts. In this case, the second request will exit.

---

- 5 Click either *OK* or *Add* and *Finish*.

Scheduling of SMT jobs in general is covered in [Section 2.5, “Setting the SMT Job Schedule with YaST”](#) (page 10)

YaST uses `cron` to schedule Novell Customer Center registrations and other SMT jobs. If you do not want to use YaST, you can use `cron` directly.

To disable automatic registration, change the `forwardRegistration` value in the `[LOCAL]` section of the `/etc/smt.conf` configuration file to `false`.

## 5.5 Lowest Repeat Time

Do not set the repeat time lower than 10 minutes, because the maximal value of the `rndRegister` is 450 (7.5 minutes). If the repeat time is lower, it can happen that the process, started before, is still sleeping when the next process starts. In such a case, the second request will exit.



# SMT Reports

SMT allows you to register client machines on Novell Customer Center. Client machines must be configured to be able to use SMT. For information about configuring clients to be able to use SMT, SMT lets you generate reports based on SMT and Novell Customer Center data. Generated reports contain statistics of the registered machines and products used and of the active, expiring, or missing subscriptions. If the number of registered machines and products exceeds the number of available subscriptions, warnings are given.

---

**NOTE: Assignment of Reports**

If you are using more than one SMT server in your environment, generated reports may not represent all of the SMT servers or machines in your environment. For the complete statistics of all your registered machines, refer to the information in the Novell Customer Center.

---

## 6.1 Report Schedule and Recipients

Generated SMT reports can be sent to a defined list of e-mail addresses periodically. To create or edit this list, and to set the frequency of the reports, use the YaST SMT Configuration module. How to configure this list is described in [Section 2.4, “Setting E-mail Addresses to Receive Reports with YaST”](#) (page 10). Configuration of the report schedule is described in [Section 2.5, “Setting the SMT Job Schedule with YaST”](#) (page 10).

The list can also be edited manually in the `reportEmail` option of the `/etc/smt.conf` configuration file. For more information about editing the list of addresses directly, see [Section “\[REPORT\] Section of /etc/smt.conf”](#) (page 60). To set the frequency of reports manually, you can directly edit the `/usr/lib/SMT/bin/smt-gen-report` line(s) of the crontab in `/etc/cron.d/novell.com-smt`. For more information about the crontab format, see `man 5 crontab`.

Reports, including those created as a scheduled SMT job, are created by the `smt-report` command. This command has various parameters. To edit parameters used with scheduled commands, edit the `/etc/smt.d/smt-cron.conf` configuration file. For more information, see [Section 7.2.2, “/etc/smt.d/smt-cron.conf”](#) (page 63).

## 6.2 Types of SMT Reports

Two types of reports can be created with the `smt-report` command:

`- -classreport`

When the `- -classreport` option is used, the created report summarizes the number of clients running products of various classes (for example, all SUSE Linux Enterprise Server\* products belong to the class `SLES`).

`--local`

If the `--local` option is used, the created report is based solely on local SMT data.

`--ncc`

If the `--ncc` option is used, the created report is based on Novell Customer Center data.

If neither `--local` nor `--ncc` is used, the type of report is determined by `forwardRegistration` in the `/etc/smt.conf` configuration file. If the option is set to `true`, the report is based on Novell Customer Center data. If it is set to `false`, the report is based on local SMT data.

If you are creating a report based on local SMT data and you do not want local data to be synchronized with the Novell Customer Center at all, use the `--nonccsync` option together with `--local` option.

## 6.3 Report Output Formats and Targets

SMT reports can be printed to the standard output, exported to one or multiple files (in CVS format) as well as mailed to the defined list of e-mail addresses. Use the following options for the `smt-report` command:

`--quiet` or `-q`

Suppress output to STDOUT and run `smt-report` in quiet mode.

`--file` or `-F`

Export report to one or several files. By default, the report will be written to a single file, rendered as tables. Optionally, the filename or whole path may be specified after the parameter: `--file filename`. If no filename is specified, a default filename containing a timestamp is used. However, SMT will not check if the file or files already exist.

In CSV (Comma-Separated Value) mode the report will be written to multiple files, therefore the specified filename will expand to `[path/]filename-reportname.extension` for every report.

`--csv` or `-c`

The report will be exported to multiple files in CSV format. The first line of each \*.csv file consists of the column names, the data starts on line two. The `--csv` parameter should only be used together with the `--file` parameter. If the specified filename contains a `.csv` extension, the report format will be CSV (as if the `--csv` parameter was used).

`--mail` or `-m`

Activate mailing of the report to the addresses configured with the YaST SMT Configuration module and written in `/etc/smt.conf`. The report will be rendered as tables.

`--attach` or `-a`

Attach the report to the mails in CSV format. This option should only be used together with the `--mail` option.

`--pdf`

The report will be exported to multiple files in pdf format.

`--xml`

The report will be exported to multiple files in xml format.

---

### **NOTE: Disabling Sending Attachments**

If you want to disable sending CSV attachments with report mails, edit the `/etc/smt.d/smt-cron.conf` configuration file as follows: remove the `--attach` option from the `REPORT_PARAMS` value. The default line reads: `REPORT_PARAMS="--mail --attach -L /var/log/smt-report.log"`. To disable CSV attachments, change it to: `REPORT_PARAMS="--mail -L /var/log/smt-report.log"`.

If you have disabled CSV attachments but need them occasionally, you can send them manually with the `smt-report --mail --attach -L /var/log/smt-report.log` command.

---

# SMT Tools and Configuration Files

# 7

This chapter describes the most important scripts, configuration files and certificates shipped with SMT.

## 7.1 Important Scripts and Tools

There are two important groups of SMT commands: The `smt` command with its sub-commands is used for managing the mirroring of updates, registration of clients, and reporting. The `rcsmt` script is used for starting, stopping, restarting SMT services, and for checking their status.

### 7.1.1 SMT JobQueue

Since SUSE Linux Enterprise version 11, there is a new SMT service called SMT JobQueue. It is a system to delegate 'jobs' to the registered clients.

The SMT server maintains a queue of jobs for each client. To enable JobQueue for a client, the `smt-client` package needs to be installed. This will automatically configure the cron job to poll for jobs for this client (usually every 3 hours). Jobs are not pushed directly to the clients and processed immediately, but the client asks for them. Therefore, a delay of several hours may occur.

Every job can have its parent job, which sets a dependency. The child job only runs after the parent job successfully finished. Job timing is also possible: a job can have a start time and an expiration time to define its earliest execution time or the time the job

will expire. A job may also be persistent. It is run repeatedly with a delay. For example, a patch status job is a persistent job that runs once a day. For each client, a patch status job is automatically generated after it registers successfully against an SMT 11 server. The patchstatus information can be queried with the `smt client` command. For the already registered clients, you can add the patchstatus jobs manually with the `smt job` command.

You can manipulate, list, create or delete the jobs. For this reason, the command line tool `smt job` was introduced. For more details on, see [Section “smt-job”](#) (page 43)

---

**NOTE: Overriding the Automatic Creation of Patchstatus Job**

When creating a softwarepush or an update job, normally a non-persistent patchstatus job will be added automatically with the parent id set to the id of the new job. To disable this behavior, use the `--no-autopatchstatus` option.

---

**NOTE: Job Timelag Limitation**

The client will process one job at a time, report back the result, and then ask for the next job. If you create a persistent job with a timelag of only a few seconds, it will be repeated forever and block other jobs of this client. Therefore, adding jobs with a timelag smaller than one minute is not supported.

---

**SMT is not intended to be a system to directly access the clients or to immediately report the results back. It is rather a longtime maintenance and monitoring system than a live interaction tool.**

## 7.1.2 /usr/sbin/smt Commands

The main command to manage the SMT is `smt (/usr/sbin/smt)`. The `smt` command should be used together with various sub-commands described in this section. If the `smt` command is used alone, it prints out a list of all available sub-commands. To get help for individual sub-commands, use `smt subcommand --help`.

The following sub-commands are available:

- `smt-client`

- `smt-delete-registration`
- `smt-job`
- `smt-list-products`
- `smt-list-registrations`
- `smt-mirror`
- `smt-ncc-sync`
- `smt-register`
- `smt-report`
- `smt-repos`
- `smt-setup-custom-repos`
- `smt-staging`
- `smt-support`
- `smt-mirror-sle9`

There are two syntax types you can use with the `smt` command: either use `smt` followed by a sub-command or use a single command (composed of `smt`, `dash`, and the sub-command of choice). For example, it is possible to use either `smt mirror` or `smt-mirror`, as both have the same meaning.

---

### **NOTE: Conflicting Commands**

Depending on your `$PATH` environment variable, the SMT `smt` command (`/usr/sbin/smt`) may collide with the `smt` command from the `star` package (`/usr/bin/smt`). Either use the absolute path `/usr/sbin/smt`, create an alias, or set your `$PATH` accordingly.

Another solution is to always use the `smt-subcommand` syntax (connected with a minus sign) instead of `smt subcommand` (separated by a space).

---

## smt-client

The `smt-client` command shows information about registered clients. The information includes:

- `guid`
- `hostname`
- `patch status`
- `timestamps of the patch status`
- `last contact with the SMT server`

The `smt-client` understands the following options:

`--verbose` or `-v`

Show detailed information about the client. The last contact date is shown as well.

`--debug` or `-d`

Enable debugging mode.

`--logfile` or `-Lfile`

Specify the file the log will be written to.

`--hostname` or `-hname`

Only the entries whose hostname begins with *name* will be listed.

`--guid` or `-gguid`

Only the entries whose GUID is *guid* will be listed.

`--severity` or `-slevel`

Filter the result by the patch status information. The value *level* can be one of 'packagemanager', 'security', 'recommended' or 'optional'. Only those entries are listed which have patches of the respective level.

## smt-delete-registration

The `smt-delete-registration` command deletes one or more registrations from SMT and Novell Customer Center. It will deregister machines from the system. The following options are available:

`--guid ID` or `-gID`

Delete the machine with the `guid ID` from the system. You can use this option multiple times.

`--debug` or `-d`

Enable debugging mode.

## smt-job

The `smt-job` script manages jobs for individual SMT clients. You can list, create, edit, or delete jobs with it. The following options are available:

`--list` or `-l`

List all client jobs. This is the default if the operation mode switch is omitted.

`--verbose` or `-vlevel`

Show detailed information about a job or jobs in a list mode. The `level` value can be a number from 0 to 3. The bigger the value, the more verbose the command is.

`--create` or `-c`

Create a new job.

`--edit` or `-e`

Edit an existing job.

`--delete` or `-d`

Delete an existing job.

`--guid` or `-gguid`

Specify the client's `guid`. This parameter can be used multiple times to create a job for more than one client.

`--jobid` or `-jid`

Specify the job ID. You need to specify job ID and client's *guid* when editing or deleting a job, because the same job for multiple clients has the same job ID.

`--deleteall` or `-Aid`

Omit either the client's *guid* or the job ID in this delete operation. The missing parameter will match all clients respective jobs.

`--type` or `-ttype`

Specify the job type. The type can be one of 'patchstatus', 'softwarepush', 'update', 'execute', 'reboot', 'wait', 'eject'. On the client, only 'patchstatus', 'softwarepush' and 'update' are enabled by default.

`--description` *description*

Specify a job description.

`--parentid`

Specify the job ID of the parent job. Use it to describe a dependency. A job will not be processed until its parent has successfully finished.

`--name` or `-nname`

Specify a job name.

`--persistent`

Specify if a job is persistent. Non-persistent jobs are processed only once, while persistent jobs are processed again and again. Use `--timelag` to define the time that elapses until the next run.

`--finished`

Search option for finished jobs.

`--targetedtime`

Specify the earliest execution time of a job. Note that the job most likely will not run exactly at that point in time, but probably some minutes or hours after. The reason is that the client polls in a fixed interval for jobs.

`--expirestime`

Define when the job will no longer be executed anymore.

`--timelagtime`

Define the time interval for persistent jobs.

For a complete list of possible options and their explanation, see the manual page of `smt-job` command (`man smt-job`).

## Examples

To list all finished jobs, enter the following:

```
# smt-job --list --finished
```

To create a 'softwarepush' job that installs `xterm` and `bash` on client 12345 and 67890, enter the following:

```
# smt-job --create -t softwarepush -P xterm -P bash -g
12345 -g 67890
```

To change the timing for a persistent job with job ID 42 and guid 12345 to run every 6 hours, enter the following:

```
# smt-job --edit -j 42 -g 12345 --targeted 0000-00-00
--timelag 06:00:00
```

To delete all jobs with job ID 42, enter the following:

```
# smt-job --delete -jobid 42 --deleteall
```

## smt-list-products

The `smt-list-products` script lists all software products in the SMT database. The following options are available:

`--used` or `-u`

Show only used products.

`--catstat` or `-c`

Show whether all repositories needed for a product are locally mirrored.

## smt-list-registrations

The `smt-list-registrations` script lists all registrations. There are two options available for this command.

`--verbose` or `-v`

Show detailed information about the registered devices.

`--format` or `-fformat`

Format the output. Possible types of formats are *asciitable* and *csv*

## smt-mirror

The `smt-mirror` command performs the mirroring procedure and downloads repositories that are set to be mirrored.

You can run the `smt-mirror` with the following options:

`--clean` or `-c`

Remove all files no longer mentioned in the metadata from the mirror. No mirroring occurs before cleanup.

`--debug` or `-d`

Enable the debugging mode.

`--deepverify`

Turn on verifying of all package checksums.

`--hardlink` *size*

Search for duplicate files with a size greater than the size specified in kilobytes. Creates hard links for them.

`--directory` *path*

Define the directory to work on. If you use this option, the default value configured in the `smt.conf` configuration file is ignored.

`--dbreplfile` *file*

Define the path to the `*.xml` file to use as database replacement. You can create such a file with the `sync-ncc` command.

`--logfile file` or `--L file`  
Specify the path to a logfile.

## **smt-ncc-sync**

The `smt-ncc-sync` or `smt ncc-sync` command gets data from the Novell Customer Center and updates the local SMT database. It can also save Novell Customer Center data to a directory instead of the SMT database, or read Novell Customer Center data from such a directory instead of downloading it from Novell Customer Center itself.

You can run the `smt-ncc-sync` with the following options:

`--fromdir directory`  
Read Novell Customer Center data from a directory instead of downloading it from Novell Customer Center.

`--todir directory`  
Write Novell Customer Center data to the specified directory without updating the SMT database.

`--createdbreplacementfile`  
Create a database replacement file for using `smt-mirror` without database.

`--logfile file` or `--L file`  
Specify the path to a log file.

`--debug`  
Enable debugging mode.

## **smt-register**

The `smt-register` or `smt-register` command registers all currently unregistered clients at the Novell Customer Center. It also registers all clients whose data has changed since the last registration.

The following options are available:

`--logfile file` or `--L file`  
Specify the path to a log file.

`--debug`  
Enable debugging mode.

## **smt-report**

The `smt-report` or `smt report` command generates a subscription report based on local calculation or Novell Customer Center registrations.

The following options are available:

`--local`  
Force the creation of a report based on a local calculation without accessing Novell Customer Center data.

`--ncc`  
Force the creation of a report based on Novell Customer Center data.

`--nosync`  
Disable synchronizing with Novell Customer Center before creating the report.

`--mail` or `-m`  
Activate mailing the report to the addresses configured with the SMT Server and written in `/etc/smt.conf`. The report will be rendered as tables.

`--attach` or `-a`  
Append the report to the e-mails in CSV format. This option should only be used together with the `--mail` option.

`--quiet` or `-q`  
Suppress output to STDOUT and runs `smt-report` in quiet mode.

`--csv` or `-c`  
The report will be exported to multiple files in CSV format. The first line of each `*.csv` file consists of the column names, the data starts on line two. The `--csv` parameter should only be used together with the `--file` parameter. If the specified filename contains `.csv` as extension, the report format will be CSV (as if the `--csv` parameter was used).

`--pdf` or `-p`

The report will be exported in PDF format. Use it only together with the `-file` option.

`--xml`

The report will be exported in XML format. Use it only together with the `-file` option. For a detailed description of this XML format, see the manual page of the `smt-report` command.

`--file` or `-F`

Export the report to one or several files. By default, the report will be written to a single file rendered as tables. Optionally, the filename or whole path may be specified after the parameter: `--file filename`. If no filename is specified, a default filename containing a timestamp is used. However, SMT will not check if the file or files already exist.

In CSV mode the report will be written to multiple files, therefore, the specified filename will expand to `[path/]filename-reportname.extension` for every report.

`--logfile filename` or `-L filename`

Specify path to a logfile.

`--debug`

Enable debugging mode.

## smt-repos

You can use the `smt-repos` (or `smt repositories`) script can to list all available repositories and for enabling or disabling repositories. The following options are available:

`--enable-mirror` or `-e`

Enable repository mirroring.

`--enable-by-prod` or `-p`

Enable repository mirroring by giving product data in the following format:

`Product[,Version[,Architecture[,Release]]]`.

`--disable-by-prod` or `-P`  
Disable repository mirroring by giving product data in the following format:  
Product [ , Version [ , Architecture [ , Release ] ] ].

`--enable-staging` or `-s`  
Enable repository staging.

`--disable-staging` or `-S`  
Disable repository staging.

`--disable-mirror` or `-d`  
Disable repository mirroring.

`--only-mirrorable` or `-m`  
List only repositories that can be mirrored.

`--only-enabled` or `-o`  
List only enabled repositories.

`--verbose` or `-v`  
Show detailed repository information.

## **smt-setup-custom-repos**

The `smt-setup-custom-repos` or `smt setup-custom-repos` script is a tool to set up custom repositories (repositories not present in NU) to be used with SMT. You can this script to add a new repository to the SMT database or to delete a repository from the database. The script recognizes the following options:

`--productid`  
ID of a product the repository belongs to. If a repository should belong to multiple products, use this option multiple times to assign repository to all relevant products.

`--name`  
The name of the custom repository.

`--description`  
The description of the custom repository.

`--exturl`

The URL where this repository can be mirrored from. Only HTTP and HTTPS protocols are supported (no directory, file, or FTP).

`--delete`

Remove a custom repository with a given ID from the SMT database.

To set up a new repository, use the following command:

```
smt-setup-custom-repos --productid Product_ID
--name Catalog_Name --exturl URL
```

For example:

```
smt-setup-custom-repos --productid 434
--name My_Catalog --exturl http://my.domain.top/My_Catalog
```

To remove an already set repository, use the following command:

```
smt-setup-custom-repos --delete Catalog_ID
```

For example:

```
smt-setup-custom-repos --delete 1cf336d819e8e5904f4d4b05ee081971a0cc8afc
```

## smt-staging

A *patch* is an update of a package or group of packages. The term *update* and *patch* are often interchangeable. With the `smt-staging` script, you can set up patch filters for update repositories. It can also help you generate both testing repositories, or repositories for the production environment.

The first argument of `smt-staging` is always the *command*. It must be followed by a *repository*. The repository can be specified either by its ID, or by *Name* and *Target* from the table scheme returned by the `smt-repos` command. The `smt-staging` script understands the following commands:

`listupdates`

List available patches and their allowed/forbidden status.

`allow/forbid`

Allow or forbids specified patches.

`createrepo`

Generate both testing and production repository with allowed patches.

`status`

Give information about both testing and production snapshots, and patch counts.

The following options apply to any `smt-staging` command:

`--logfile` or `-Lfile path`

Write log information to the specified file. If it does not exist, it is created.

`--debug` or `-d`

Turn on the debugging output and log.

`--verbose` or `-v`

Turn more detailed output on.

The following options apply to specific `smt-staging` command:

`--patch`

Specify a patch by its ID. You can get a list of available patches with the `listupdates` command. This option can be used multiple times. Use it with the `allow`, `forbid`, and `listupdates` commands. If used with `listupdates`, the command will print detailed information about the specified patches.

`--category`

Specify the patch category. The following categories are available: 'security', 'recommended' and 'optional'. Use it with the `allow`, `forbid`, and `listupdates` commands.

`--all`

Allow or forbid all patches in the `allow` or `forbid` commands.

`--testing`

Use with the `createrepo` command to generate a repository for testing. The repository will be generated from the full unfiltered local mirror of the remote repository. It will be written into `<MirrorTo>/repo/testing` directory, where `MirrorTo` is the value taken from `smt.conf`.

`--production`

Use with the `createrepo` command to generate a repository for production use. The repository will be generated from the testing repository. It will be written into `<MirrorTo>/repo` directory, where `MirrorTo` is the value taken from `smt.conf`. If the testing repository does not exist, the production repository will be generated from the full unfiltered local mirror of the remote repository.

`--nohardlink`

During the repository creation with the `createrepo` command, avoid creating hard links instead of copying files. If not specified, hard links are created instead.

`--nodesc`

Do not print patch descriptions and summaries to save some screen space and make the output more readable.

## smt-support

The `smt-support` command manages uploaded support data usually coming from the `supportconfig` tool. You can forward the data to Novell, either selectively or all. This command understands the following options:

`--incoming` or `-idirectory`

Specify the directory where the `supportconfig` archives are uploaded. You can also set this option with the `SMT_INCOMING` environment variable. The default `SMT_INCOMING` directory is `/var/spool/smt-support`

`--list` or `-l`

List the uploaded `supportconfig` archives in the `> incoming` directory

`--remove` or `-rarchive`

Delete the specified archive.

`--empty` or `-R`

Delete all archives in the `incoming` directory.

`--upload` or `-uarchive`

Upload the specified archive to Novell. If you specify `-s`, `-n`, `-c`, `-p`, and `-e` options, the archive is repackaged with contact information.

`--uploadall` or `-U`  
Upload all archives in the incoming directory to Novell.

`--srnum` or `-sSR number`  
Accept the Novell Service Request 11 digit number.

`--name` or `-nname`  
Enter the first and last name of the contact, in quotes.

`--company` or `-ccompany`  
Enter the company name.

`--storeid` or `-did`  
Enter the store ID, if applicable.

`--terminalid` or `-tid`  
Enter the terminal ID, if applicable.

`--phone` or `-pphone`  
Enter the phone number of the contact.

`--email` or `-email`  
Enter the email address of the contact.

## 7.1.3 rcsmt Init Script

The `rcsmt` script starts, restarts, or stops SMT services. If used without any sub-commands, it returns a help text. The `rcsmt` script can be used with the following sub-commands:

```
rcsmt start  
Start the SMT services.
```

```
rcsmt stop  
Stop the SMT services.
```

```
rcsmt status  
Check the status of the SMT services. Checks whether httpd, MySQL, and cron  
are running.
```

```
racsmt restart
```

Restart the SMT services.

```
racsmt try-restart
```

Check whether the SMT is enabled and if so, restarts the SMT services.

You can enable and disable SMT with the SMT Server.

## 7.2 SMT Configuration Files

SMT has a main configuration file: `/etc/smt.conf`. Most of the options you can set with the SMT Server module. Another important configuration file is `/etc/smt.d/smt-cron.conf`, which contains parameters for commands launched as SMT scheduled jobs.

### 7.2.1 `/etc/smt.conf`

The `/etc/smt.conf` file has several sections. The `[NU]` section contains the NU credentials and URL. The `[DB]` section contains the configuration of the MySQL database SMT uses. The `[LOCAL]` section includes other configuration data. The `[REPORT]` section contains the configuration of SMT reports. You can find the configuration for the `smt-mirror-sle9` command in the `YOU9-*` sections.

---

#### WARNING

The `/etc/smt.conf` contains passwords in clear text and its default permissions (640, root, wwwrun) make its content easily accessible with scripts running on the Apache server. Be careful with running other software on the SMT Apache server. The best policy is to use this server only for SMT.

---

### [NU] Section of `/etc/smt.conf`

The following options are available in the `[NU]` section:

`NUUrl`

URL of the NU service. In most cases, it should contain the `https://nu.novell.com/` URL.

`NURegUrl`

URL of the NU registration service. It is used by `smt-ncc-sync`. If this option is missing, the URL from `/etc/suseRegister.conf` is used as a fallback.

`NUUser`

`NUUser` should contain the username for NU service. For information about getting mirroring credentials, see [Section 3.1, “Getting Mirror Credentials”](#) (page 13). You can set this value with the SMT Server.

`NUPass`

`NUPass` is the password for the user defined in `NUUser`. For information about getting mirroring credentials, see [Section 3.1, “Getting Mirror Credentials”](#) (page 13). You can set this value with the SMT Server.

## **[DB] Section of /etc/smt.conf**

The three options defined in the [DB] section are used for configuring the database SMT uses. Currently, only MySQL is supported by SMT.

`config`

The first parameter of the DBI->connect Perl method used for connection to the MySQL database. The value should be in the form

```
dbi:mysql:database=smt;host=localhost
```

where *smt* is the name of the database and *localhost* the hostname of the database server.

`user`

The user for the database. The default value is `smt`.

`pass`

The password for the database user. You can set the password with the SMT Server.

## [LOCAL] Section of /etc/smt.conf

The following options are available in the [LOCAL] section:

### url

The base URL of the SMT server which is used to construct URLs of the repositories available on the server. This value should be set by YaST automatically during installation. The format of this option should be:

```
https://server.domain.tld/.
```

You can change the URL change manually. For example, the administrator may choose to use the `http://` scheme instead of `https://` for performance reasons. Another reason may be using an alias (configured with CNAME in DNS) instead of the hostname of the server, for example `http://smt.domain.tld/` instead of `http://server1.domain.tld/`.

### nccEmail

E-mail address used for registration at the Novell Customer Center. The SMT administrator can set this value with the SMT Server.

### MirrorTo

Determines the path to mirror to.

### MirrorAll

If the `MirrorAll` option is set to `true`, the `smt-ncc-sync` script will set all repositories that can be mirrored to be mirrored (DOMIRROR flag).

### MirrorSRC

If the `MirrorSRC` option is set to `false`, no source RPM packages are mirrored.

### forwardRegistration

Determine if the clients registered at SMT should be registered at Novell Customer Center, too. If the `forwardRegistration` option is set to `true`, client registrations will be forwarded to Novell Customer Center. If the `forwardRegistration` option is set to `false`, no client registrations will be sent to Novell Customer Center.

### `rndRegister`

Specify a delay in seconds before registration of clients at Novell Customer Center. The value is a random number between 0 and 450, generated by the SMT Server. The purpose of this random delay is to prevent a high load on the Novell Customer Center server that would occur if all `smt-register` cron jobs connected at the same time.

### `HTTPProxy`

If you do not want to use global proxy settings, specify the proxy to be used for HTTP connection here. Use the following form:

```
http://proxy.example.com:3128.
```

If the proxy settings are not configured in `/etc/smt.conf`, the global proxy settings configured in `/etc/syconfig/proxy` are used. You can configure the global proxy settings with the YaST Proxy module.

The `HTTPProxy` also applies to the `smt-mirror-sle9` script.

### `HTTPSProxy`

If you do not want to use global proxy settings, specify the proxy to be used for HTTPS connection here. Use the form: `http://proxy.example.com:3128`.

If the proxy settings are not configured in `/etc/smt.conf`, the global proxy settings configured in `/etc/syconfig/proxy` are used. You can configure the global proxy settings with the YaST Proxy module.

The `HTTPSProxy` also applies to the `smt-mirror-sle9` script.

### `ProxyUser`

If your proxy requires authentication, specify a username and password here, using the `username:password` format.

If the proxy settings are not configured in `/etc/smt.conf`, the global proxy settings configured in `/etc/syconfig/proxy` are used. You can configure the global proxy settings with the YaST Proxy module.

Neither the `ProxyUser` value nor the global proxy authentication settings apply to the `smt-mirror-sle9` script. For user authentication in `smt-mirror-sle9`

write the following in the `$HOME/.wgetrc` file: `proxy_user=username`  
`proxy_password=password`.

`requiredAuthType`

Specify an authentication type to access the repository. There are three possible types:

- `none` - no authentication is required. This is the default value
- `lazy` - only username and password are checked. A valid user can access all repositories.
- `strict` - checks also if the user has access to the repository.

`smtUser`

Specify a username of a Unix user under which all `smt` commands will run.

`signingKeyID`

Specify ID of the GPG key to sign modified repositories. The user specified under `smtUser` needs to have access to the key. If this option is not set, the modified repositories will be unsigned.

## [REST] Section of `/etc/smt.conf`

The following options are available in the `[REST]` section:

`enableRESTAdminAccess`

If set to `1`, turns administrative access to the SMT `RESTService` on. Default value is `0`.

`RESTAdminUser`

Specify the username which the `REST-Admin` uses to login. Default value is `RESTroot`.

`RESTAdminPassword`

Specify the password for the `REST-Admin` user. The option has no default value. Empty password is invalid.

## **[JOBQUEUE] Section of /etc/smt.conf**

The following options are available in the [JOBQUEUE] section:

`maxFinishedJobAge`

Specify the maximum age of finished non-persistent jobs in days. Default value is 8.

## **[REPORT] Section of /etc/smt.conf**

The following options are available in the [REPORT] section:

`reportEmail`

A comma separated list of e-mail addresses to send SMT status reports to. You can set this list with the SMT Server.

`reportEmailFrom`

From field of report e-mails. If not set, the default `root@hostname.domainname` will be used.

`mailServer`

Relay mail server. If empty, e-mails are sent directly.

`mailServerPort`

Port of the relay mail server set in `mailServer`.

`mailServerUser`

User name for authentication to the mail server set in `mailServer`.

`mailServerPassword`

Password for authentication to the mail server set in `mailServer`.

## **smt-mirror-sle9 Sections of /etc/smt.conf**

Each product to be mirrored by the `smt-mirror-sle9` command has a separate predefined `YOU9-*` section in the `/etc/smt.conf` configuration file. `/etc/smt.conf` is not configured via the SMT database like the `smt-mirror` command, all configuration is contained in `/etc/smt.conf`.

`mirror_prod`

A product to be mirrored. For example `Novell-Linux-Desktop`.

`mirror_archs`

Comma-separated list of architectures to be mirrored. For example `i386,x86_64`.

Remove any architectures that do not need to be mirrored.

`mirror_version`

The version of the product to be mirrored, for example `9`.

`mirror`

If you want to mirror this product, set `mirror` to `true`.

`credentials`

If you want to mirror this product, provide credentials in the `user:password` format.

## Example `/etc/smt.conf`

### Example 7.1 `config/smt.conf`

```
[NU]
NUUrl = https://nu.novell.com/
NURegUrl = https://secure-www.novell.com/center/regsvc/
NUUser = exampleuser
NUPass = examplepassword

[DB]
config = dbi:mysql:database=smt;host=localhost
user = smt
pass = smt

[LOCAL]
# Default should be http://server.domain.top/
url = http://smt.example.com/
# This email address is used for registration at NCC
nccEmail = exampleuser@example.com
MirrorTo = /srv/www/htdocs
MirrorAll = false
MirrorSRC = true
forwardRegistration = true
rndRegister = 410
# specify proxy settings here, if you do not want to use the global proxy
settings
# If you leave these options empty the global options are used.
```

```

#
# specify which proxy you want to use for HTTP connection
# in the form http://proxy.example.com:3128
HTTPProxy =
# specify which proxy you want to use for HTTPS connection
# in the form http://proxy.example.com:3128
HTTPSProxy =
# specify username and password if your proxy requires authentication
# in the form username:password
ProxyUser =
#
# require authentication to access the repository?
# Three possible authtypes can be configured here
# 1) none : no authentication required (default)
# 2) lazy : check only username and password. A valid user has access to
all repositories
# 3) strict : check also if this user has access to the repository.
#
requiredAuthType = none
#
# the smt commands should run with this unix user
#
smtUser = smt
#
# ID of the GPG key to be used to sign modified (filtered) repositories.
# The key must be accessible by the user who runs SMT, i.e. the user specified
# in the 'smtUser' configuration option.
#
# If empty, the modified repositories will be unsigned.
#
signingKeyID =

[REPORT]
# comma separated list of eMail addresses where the status reports will be
sent to
reportEmail = exampleuser@example.com
# from field of report mails - if empty it defaults to
"root@<hostname>.<domainname>"
reportEmailFrom =
# relay mail server - leave empty if mail should be sent directly
mailServer =
mailServerPort =
# mail server authentication - leave empty if not required
mailServerUser =
mailServerPassword =

[YOU9-SLES-SDK]
mirror_prod = SLES-SDK
mirror_archs = i386, ia64, ppc, s390, s390x, x86_64
mirror_version = 9
mirror = false
credentials =

```

```
[YOU9-SUSE-CORE]
mirror_prod = SUSE-CORE
mirror_archs = i386,ia64,ppc,s390,s390x,x86_64
mirror_version = 9
mirror = false
credentials =
```

```
[YOU9-SUSE-SLES]
mirror_prod = SUSE-SLES
mirror_archs = i386,ia64,ppc,s390,s390x,x86_64
mirror_version = 9
mirror = false
credentials =
```

## 7.2.2 /etc/smt.d/smt-cron.conf

The `/etc/smt.d/smt-cron.conf` configuration file contains options of the SMT commands launched as SMT scheduled jobs set with YaST (see [Section 2.5, “Setting the SMT Job Schedule with YaST”](#) (page 10)). Cron is used to launch these scheduled jobs. The cron table is located in the `/etc/cron.d/novell.com-smt` file.

### NCC\_SYNC\_PARAMS

Contains parameters of the `smt ncc-sync` command, if called as a part of an SMT scheduled job via cron. The default value is `"-L /var/log/smt/smt-ncc-sync.log --mail"`.

### MIRROR\_PARAMS

Contains parameters of the `smt mirror` command, if called as a part of an SMT scheduled job via cron. The default value is `"-L /var/log/smt/smt-mirror.log --mail"` .

### REGISTER\_PARAMS

Contains parameters of the `smt register` command, if called as a part of an SMT scheduled job via cron. The default value is `"-r -L /var/log/smt/smt-register.log --mail"` .

### REPORT\_PARAMS

Contains parameters of the `smt report` command, if called as a part of an SMT scheduled job via cron. The default value is `"--mail --attach -L /var/log/smt/smt-report.log"` .

JOBQUEUECLEANUP\_PARAMS

Contains parameters for smt jobqueue cleanup, if called as a part of an SMT scheduled job via cron. The default value is "--mail -L /var/log/smt/smt-jobqueuecleanup.log".

## 7.3 Server Certificates

For communication between the SMT server and client machines, the encrypted HTTPS protocol is used, requiring a server certificate. If the certificate is not available, or if clients are not configured to use the certificate, the communication between server and clients will fail.

Every client must be able to verify the server certificate by trusting the CA (certificate authority) certificate which signed the server certificate. Therefore, the SMT server provides a copy of the CA at `/srv/www/htdocs/smt.crt`. This CA can be downloaded from every client via the URL `http://FQDN/smt.crt`. The copy is created when YaST writes the SMT configuration. Whenever SMT is started with the `rcsmt` init script, it checks the certificate. If a new CA certificate exists, it is copied again. Therefore, whenever the CA certificate is changed, restart SMT using the `rcsmt restart` command.

When the SMT Server module applies configuration changes, it checks for the existence of the common server certificate. If the certificate does not exist, YaST asks whether the certificate should be created. If the user confirms, the YaST CA Management module is started.

### 7.3.1 Certificate Expiration

The common server certificate SMT uses is valid for one year. After that time, a new certificate is needed. Either generate a new certificate using YaST CA Management module or import a new certificate using the YaST Common Server Certificate module. Both options are described in the following sections.

As long as the same CA certificate is used, there is no need to update certificates at the client machines. The generated CA certificate is valid for 10 years.

## 7.3.2 Creating a New Common Server Certificate

To create a new common server certificate with YaST, proceed as follows:

- 1 Start YaST and select *Security and Users > CA Management*. Conversely, start the YaST CA Management module from a command line by entering `yast2 ca_mgm as root`.
- 2 Select the required CA and click *Enter CA*.
- 3 Enter the password if entering a CA for the first time. YaST displays the CA key information in the *Description* tab.
- 4 Click the *Certificates* tab (see [Figure 7.1, “Certificates of a CA”](#) (page 65)) and select *Add > Add Server Certificate*.

**Figure 7.1** *Certificates of a CA*



- 5 Enter the fully qualified domain name of the server as *Common Name*. Add a valid e-mail address of the server administrator. Other fields, as *Organization*, *Organizational Unit*, *Locality*, and *State* are optional. Click *Next* to proceed.

---

## IMPORTANT: Hostname in Server Certificate

The server certificate must contain the correct hostname. If the client requests server `https://some.hostname/`, then `some.hostname` must be part of the certificate. The hostname must either be used as the *Common Name*, see [Step 5](#) (page 65), or as the *Subject Alternative Name*, see [Step 7](#) (page 66): `DNS:some.hostname` and/or `IP:<ipaddress>`.

---

- 6 Enter a *Password* for the private key of the certificate and reenter it in the next field to verify it.
- 7 If you want to define a *Subject Alternative Name*, click *Advanced Options*, select *Subject Alternative Name* from the list and click *Add*.
- 8 If you want to keep the default values for the other options, like *Key Length* and *Valid Period*, click *Next*. An overview of the certificate to be created is shown.
- 9 Click *Create* to generate the certificate.
- 10 To export the new certificate as the common server certificate, select it in the *Certificates* tab and select *Export > Export as Common Server Certificate*.
- 11 After having created a new certificate, restart SMT using the `rcsmt restart` command. Restarting SMT ensures that the new certificate is copied from `/etc/ssl/certs/YaST-CA.pem` to `/srv/www/htdocs/smt.crt`, the copy SMT uses. Restarting SMT also restarts the Web server.

For detailed information about managing certification and further usage of the YaST CA Management module and the Common Sever Certificate module, refer to the *Administration Guide*.

## 7.3.3 Importing a Common Server Certificate

You can import an own common server certificate from a file. The certificate to be imported has to be in the PKCS12 format with CA chain. Common server certificates can be imported with the YaST Common Server Certificate module.

To import an own certificate with YaST, proceed as follows:

- 1 Start YaST and select *Security and Users > Common Server Certificate*. Conversely, start the YaST Common Server Certificate module from the command line by entering `yast2 common_cert` as root.

The description of the currently used common server certificate is shown in the dialog that opens.

- 2 Click *Import* and select the file containing the certificate to be imported. Specify the certificate password in the *Password* field.
- 3 Press *Next*. If the certificate is successfully imported, close YaST with *Finish*.
- 4 After having created a new certificate, restart SMT using the `rcsmt restart` command. Restarting SMT ensures that the new certificate is copied from `/etc/ssl/certs/YaST-CA.pem` to `/srv/www/htdocs/smt.crt`, the copy SMT uses. Restarting SMT also restarts the Web server.

## 7.3.4 Synchronizing Time Between SMT Server and Clients

There is no need for a precise synchronization of time between the SMT server and clients. However, each server certificate has a validity period. If the client happens to be set to a time outside of this period, the certificate validation on the client side fails.

Therefore, it is advisable to keep the time on the server and clients synchronized. You can easily synchronize time using NTP (network time protocol). Use `yast2 ntp-client` to configure an NTP client. You will find detailed information about NTP in *Administration Guide*.



# Configuring Clients to Use SMT

# 8

Any machine running SUSE Linux Enterprise SP2 or later can be configured to register against SMT and download software updates from there, instead of communicating directly with the Novell Customer Center and the NU servers.

If your network includes an SMT server to provide a local update source, you need to equip the client with the server's URL. As client and server communicate via the HTTPS protocol during registration, you also need to make sure the client trusts the server's certificate. In case you set up your SMT server to use the default server certificate, the CA certificate will be available on the SMT server at `http://FQDN/smt.crt`. In this case you do not have to manually set the certificate: The registration process will automatically download the CA certificate from there, unless configured otherwise. You have to enter a path to the server's CA certificate if the certificate was issued by an external certificate authority.

---

**NOTE: Registering Against \*.novell.com Subdomain**

If you try to register against any \*.novell.com subdomain, the certificate will not be downloaded during registration for security reasons, and certificate handling will not be done. In such a case, use a different domain name or a plain IP address.

---

There are several ways to provide this information and to configure the client machine to use SMT. The first way is to provide the needed information via kernel parameters at boot time. The second way is to configure clients using an AutoYaST profile. There is also the `clientSetup4SMT.sh` script. This script can be run on a client to make it register against a specified SMT server. In SLE11 you can now set the SMT server

URL with the YaST2 registration module during installation. These methods are described in the following sections:

## 8.1 Using Kernel Parameters to Access an SMT Server

Any client can be configured to use SMT by providing the following kernel parameters during machine boot: `regurl` and `regcert`. The first parameter is mandatory, the latter is optional.

### `regurl`

URL of the SMT server. The URL needs to be in the following format:

`https://FQDN/center/regsvc/` with *FQDN* being the fully qualified hostname of the SMT server. It must be identical to the FQDN of the server certificate used on the SMT server. Example:

```
regurl=https://smt.example.com/center/regsvc/
```

### `regcert`

Location of the SMT server's CA certificate. Specify one of the following locations:

#### URL

Remote location (http, https or ftp) from which the certificate can be downloaded. Example:

```
regcert=http://smt.example.com/smt.crt
```

#### Floppy

Specifies a location on a floppy. The floppy has to be inserted at boot time—you will not be prompted to insert it if it is missing. The value has to start with the string `floppy`, followed by the path to the certificate. Example:

```
regcert=floppy/smt/smt-ca.crt
```

#### Local Path

Absolute path to the certificate on the local machine. Example:

```
regcert=/data/inst/smt/smt-ca.cert
```

### Interactive

Use `ask` to open a pop-up menu during installation where you can specify the path to the certificate. Do not use this option with AutoYaST. Example:

```
regcert=ask
```

### Deactivate Certificate Installation

Use `done` if either the certificate will be installed by an add-on product, or if you are using a certificate issued by an official certificate authority. Example:

```
regcert=done
```

---

## **WARNING: Beware of Typing Errors**

Make sure the values you enter are correct. If `regurl` has not been specified correctly, the registration of the update source will fail.

If a wrong value for `regcert` has been entered, you will be prompted for a local path to the certificate. In case `regcert` is not specified at all, it will default to `http://FQDN/smt.crt` with `FQDN` being the name of the SMT server.

---

## **WARNING: Change of SMT Server Certificate**

If the SMT server gets a new certificate from a new and untrusted CA, the clients need to retrieve the new CA certificate file. This is done automatically with the registration process but only if a URL was used at installation time to retrieve the certificate, or if the `regcert` parameter was omitted and thus, the default URL is used. If the certificate was loaded using any other method, such as floppy or local path, the CA certificate will not be updated.

---

## **8.2 Configuring Clients with AutoYaST Profile**

Clients can be configured to register with SMT server via AutoYaST profile. For general information about creating AutoYaST profiles and preparing automatic installation, refer to the *Administration Guide*. In this section, only SMT specific configuration is described.

To configure SMT specific data using AutoYaST, follow these steps:

- 1 As `root`, start YaST and select *Miscellaneous > Autoinstallation* to start the graphical AutoYaST front-end.

From a command line, you can start the graphical AutoYaST front-end with the `yast2 autoyast` command.

- 2 Open an existing profile using *File > Open*, create a profile based on the current system's configuration using *Tools > Create Reference Profile*, or just work with an empty profile.
- 3 Select *Software > Novell Customer Center Configuration*. An overview of the current configuration is shown.
- 4 Click *Configure*.
- 5 Set the URL of the *SMT Server* and, optionally, the location of the *SMT Certificate*. The possible values are the same as for the kernel parameters `regurl` and `regcert` (see [Section 8.1, “Using Kernel Parameters to Access an SMT Server”](#) (page 70)). The only exception is that the `ask` value for `regcert` does not work in AutoYaST, because it requires user interaction. If using it, the registration process will be skipped.
- 6 Perform all other configuration needed for the systems to be deployed.
- 7 Select *File > Save As* and enter a filename for the profile, such as `autoinst.xml`.

## 8.3 Configuring Clients with the `clientSetup4SMT.sh` Script

The `/usr/share/doc/packages/smt/clientSetup4SMT.sh` script is provided with SMT. This script allows you to configure a client machine to use a SMT server or to reconfigure it to use a different SMT server.

To configure a client machine to use SMT with the `clientSetup4SMT.sh` script, follow these steps:

- 1 Copy the `/usr/share/doc/packages/smt/clientSetup4SMT.sh` script at your SMT server to the client machine.
- 2 As `root`, execute the script on the client machine. The script can be executed in two ways. In the first case, the script name is followed by the registration URL:  
`./clientSetup4SMT.sh registration_URL` For example,  
`./clientSetup4SMT.sh https://smt.example.com/center/regsvc`. In the second case, the script name is followed by the `--host` option followed by hostname of the SMT server: `./clientSetup4SMT.sh --host server_hostname`. For example, `./clientSetup4SMT.sh --host smt.example.com`.
- 3 The script downloads the server's CA certificate. Accept it by pressing `y`.
- 4 The script performs all necessary modifications on the client. However, the registration itself is not performed by the script.
- 5 Downloads and asks to accept additional GPG keys to sign repositories with.
- 6 Perform a registration by executing `suse_register` or running `yast2 inst_suse_register` module on the client.

The `clientSetup4SMT.sh` script works with SUSE Linux Enterprise 10 SP1, 10 SP2 and SLE11 systems.

This script is also provided for download. You can get it by calling:

```
$> wget http://FQDN/repo/tools/clientSetup4SMT.sh
```

## 8.3.1 Problems Downloading GPG Keys from the Server

The `apache2-example-pages` package includes a `robots.txt` file. The file is installed into the Apache2 document root directory, and controls how clients can access

files from the web server. If this package is installed on the server, `clientSetup4SMT .sh` fails to download the keys stored under `/repo/keys`.

You can solve this problem by either editing `robots.txt`, or uninstalling the `apache2-example-pages` package.

If you choose to edit the `robots.txt` file, open it in your favorite text editor and add the `Allow: /repo/keys` line before the `Disallow: /` line. Then save the changes and quit the editor.

## 8.4 Registering Clients Against SMT Test Environment

To configure a client to register against the test environment instead the production environment, modify `/etc/suseRegister.conf` on the client machine by setting:

```
register = command=register&testenv=1
```

For more information about using SMT with a test environment, see [Section 3.4, “Using the Test Environment”](#) (page 20).

# Index

