



# Micro Focus Subscription Management Tool 2.0 Guide

November 2022

## **Legal Notice**

**Copyright © 2022 Micro Focus. All Rights Reserved.**

The only warranties for products and services of Micro Focus and its affiliates and licensors (“Micro Focus”) are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

---

# Contents

|   |           |
|---|-----------|
| <b>About This Guide</b>   | <b>5</b>  |
| <b>1 Overview of Micro Focus Subscription Management Tool</b>   | <b>7</b>  |
| <b>2 Micro Focus SMT Installation</b>   | <b>9</b>  |
| 2.1 Installation During the Initial Installation Process  | 9         |
| 2.2 Installation on Top of an Already Installed System  | 10        |
| 2.3 SMT Configuration Wizard  | 10        |
| 2.4 Upgrade from Micro Focus SMT 1.1 to Micro Focus SMT 2.0   | 11        |
| 2.4.1 Supported Upgrade Paths   | 11        |
| 2.4.2 Upgrading Micro Focus SMT 1.1 on OES 2018 SP2 or SP3 and SLES 12 SP4 or SP5 to<br>Micro Focus SMT 2.0 | 11        |
| <b>3 Micro Focus SMT Server Configuration</b>   | <b>15</b> |
| 3.1 Activating and Deactivating SMT with YaST   | 15        |
| 3.2 Setting NU Credentials with YaST  | 16        |
| 3.3 Setting SMT Database Password with YaST   | 18        |
| 3.4 Setting E-mail Addresses to Receive Reports with YaST   | 18        |
| 3.5 Setting the SMT Job Schedule with YaST  | 18        |
| <b>4 Mirroring Repositories on the Micro Focus SMT Server</b>   | <b>21</b> |
| 4.1 Mirroring Credentials   | 21        |
| 4.2 Managing Software Repositories with SMT Command Line Tools  | 21        |
| 4.2.1 Updating the local SMT database   | 21        |
| 4.2.2 Enabled Repositories and Repositories that Can Be Mirrored  | 21        |
| 4.2.3 Getting Information About Repositories  | 22        |
| 4.2.4 Selecting Repositories to be Mirrored   | 22        |
| 4.2.5 Deleting Mirrored Repositories  | 23        |
| 4.2.6 Mirroring Custom Repositories   | 24        |
| 4.3 The /srv/www/htdocs Structure   | 24        |
| 4.4 Using the Test Environment  | 25        |
| 4.5 Testing and Filtering Update Repositories with Staging  | 26        |
| <b>5 Managing Repositories with YaST SMT Server Management</b>  | <b>29</b> |
| 5.1 Starting SMT Management Module  | 29        |
| 5.2 Viewing and Managing Repositories   | 29        |
| 5.2.1 Filtering Repositories  | 30        |
| 5.2.2 Mirroring Repositories  | 30        |
| 5.3 Staging Repositories  | 31        |
| 5.4 Checking the Client Status  | 32        |

|           |   |           |
|-----------|---|-----------|
| <b>6</b>  | <b>Managing Client Machines with Micro Focus SMT</b>                              | <b>35</b> |
| 6.1       | Listing Registered Clients .....  | 35        |
| 6.2       | Deleting Registrations .....  | 35        |
| 6.3       | Manual Registration of Clients at Micro Focus Customer Center .....               | 35        |
| 6.4       | Scheduling Periodic Registrations of Clients at Micro Focus Customer Center ..... | 35        |
| <b>7</b>  | <b>SMT Reports</b>  | <b>37</b> |
| 7.1       | Report Schedule and Recipients .....  | 37        |
| 7.2       | Types of SMT Reports .....  | 37        |
| 7.3       | Report Output Formats and Targets .....   | 38        |
| <b>8</b>  | <b>SMT Tools and Configuration Files</b>  | <b>41</b> |
| 8.1       | Important Scripts and Tools .....   | 41        |
| 8.1.1     | SMT JobQueue .....  | 41        |
| 8.1.2     | /usr/sbin/smt Commands .....  | 42        |
| 8.1.3     | SMT systemd Commands .....  | 53        |
| 8.2       | SMT Configuration Files .....   | 53        |
| 8.2.1     | /etc/smt.conf .....   | 53        |
| 8.2.2     | /etc/smt.d/smt-cron.conf .....  | 60        |
| 8.3       | Server Certificates .....   | 61        |
| 8.3.1     | Certificate Expiration .....  | 61        |
| 8.3.2     | Synchronizing Time Between SMT Server and Clients .....                           | 62        |
| <b>9</b>  | <b>Configuring Clients to Use Micro Focus SMT</b>                                 | <b>63</b> |
| 9.1       | Using Kernel Parameters to Access an SMT Server .....                             | 63        |
| 9.2       | Configuring Clients with AutoYaST Profile .....                                   | 65        |
| 9.3       | Configuring Clients with the clientSetup4SMT.sh Script .....                      | 65        |
| 9.3.1     | Problems Downloading GPG Keys from the Server .....                               | 66        |
| 9.4       | Configuring Clients with YaST .....   | 66        |
| 9.4.1     | Configuring Clients with YaST .....   | 67        |
| 9.5       | Registering Clients Against SMT Test Environment .....                            | 67        |
| 9.6       | Listing Accessible Repositories .....   | 67        |
| <b>10</b> | <b>SMT server Certificate Authority and Server Certificate</b>                    | <b>69</b> |
| 10.1      | Create a SMT server CA and server certificate .....                               | 69        |

# About This Guide

This guide describes how to install, configure, and update the Micro Focus Subscription Management Tool 2.0 (Micro Focus SMT 2.0) on OES 2023 and SLES 15 SP4.

- ♦ Chapter 1, “Overview of Micro Focus Subscription Management Tool,” on page 7
- ♦ Chapter 2, “Micro Focus SMT Installation,” on page 9
- ♦ Chapter 3, “Micro Focus SMT Server Configuration,” on page 15
- ♦ Chapter 4, “Mirroring Repositories on the Micro Focus SMT Server,” on page 21
- ♦ Chapter 5, “Managing Repositories with YaST SMT Server Management,” on page 29
- ♦ Chapter 6, “Managing Client Machines with Micro Focus SMT,” on page 35
- ♦ Chapter 7, “SMT Reports,” on page 37
- ♦ Chapter 8, “SMT Tools and Configuration Files,” on page 41
- ♦ Chapter 9, “Configuring Clients to Use Micro Focus SMT,” on page 63
- ♦ Chapter 10, “SMT server Certificate Authority and Server Certificate,” on page 69

## Audience

This guide is intended for system administrators.

## Feedback

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.

## Documentation Updates

For the most recent version of the guide, visit the [Documentation Web site \(https://www.microfocus.com/documentation/subscription-management-tool/smt-2.0/\)](https://www.microfocus.com/documentation/subscription-management-tool/smt-2.0/).



# 1 Overview of Micro Focus Subscription Management Tool

Micro Focus Subscription Management Tool (Micro Focus SMT or MFSMT) on Open Enterprise Server OES 2023, SUSE Linux Enterprise SLES 15 SP4 helps customers to manage their software updates while maintaining corporate firewall policy and regulatory compliance requirements. Micro Focus SMT is a package proxy system that is integrated with the Micro Focus Customer Center and provides key Micro Focus Customer Center capabilities locally at the customer site. It provides a repository and registration target that is synchronized with the customer center, thus maintaining all the capabilities of the Micro Focus Customer Center while allowing a more secure centralized deployment.





# 2 Micro Focus SMT Installation

Micro Focus SMT is an add-on product for Open Enterprise Server 2023 and SUSE Linux Enterprise Server 15 SP4 systems. You can either install the Micro Focus SMT add-on together with your base system during the initial installation process, or on top of an already-installed base system at any later time. Micro Focus SMT can be obtained free of charge from [Micro Focus Downloads](#).

---

**NOTE:** Installation Scope

SMT is a LAMP (Linux, Apache, MariaDB, PHP) based application. During the installation, Apache and MySQL will automatically be installed if missing.

---

---

**IMPORTANT**

- ♦ The Repository Mirroring Tool (RMT) released by SUSE and the Micro Focus Subscription Management Tool (Micro Focus SMT) released by Micro Focus cannot co-exist on the same server.
  - ♦ During installation of Micro Focus SMT, ensure that the base media or pool channel is available as the dependent packages will be pulled from them. For example, SUSE Linux Enterprise Server 15 SP4 base media or SLE-Product-SLES15-SP4-Pool on SLES 15 SP4.
  - ♦ Ensure that the server is configured with eDirectory before configuring Micro Focus SMT if you are planning to configure SMT on OES Server. The SMT configuration uses an existing eDirectory certificate.
- 

## 2.1 Installation During the Initial Installation Process

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

---

**NOTE:** The installation steps are same across OES 2023 and SLES 15 SP4, unless called out specifically.

---

- 1 Start OES or SLES installation. For more information, see the OES or SLES documentation.
- 2 To include the Micro Focus SMT add-on product, check the **I would like to install an additional Add On product** option during the installation and, if you are installing from a local ISO image, select **Local ISO Image** as the media type (repository). If you are installing from a different source, such as CD, NFS, or HTTP, choose the appropriate type. Then click **Next**.
- 3 If you are installing from a CD, insert the add-on product CD. If you are installing from a different source, provide the necessary repository information.
- 4 (Conditional - SLES) Click **Trust** to import the key found in repository.
- 5 By default, in the **Installation Settings** page, if the **Micro Focus Subscription Management Tool** is not selected, then select the pattern by clicking the **Software**.

- 6 Confirm the Packages notifications and click **OK**.
- 7 Continue with the OES or SLES installation.
- 8 On successfully completing OES or SLES installation, perform an initial SMT configuration using YaST > **Network Services** > **Micro Focus SMT Configuration Wizard** or using the command `yast2 smt-wizard` as described in [“SMT Configuration Wizard” on page 10](#).

## 2.2 Installation on Top of an Already Installed System

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

- 1 Start YaST and select **Software** > **Add-On Product**. Then click **Add**.
- 2 If you are installing Micro Focus SMT from a local ISO image, select **Local ISO Image** as the media type (repository). If you are installing from a different source, such as CD, NFS, or HTTP, choose the appropriate type. Then 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 repository information.
- 4 (Conditional - SLES) Click **Trust** to import the key found in repository.
- 5 By default, in the **Software Management** page, if the **Micro Focus Subscription Management Tool** is not selected, then select the pattern.
- 6 Click **Accept** to install the **Micro Focus Subscription Management Tool** pattern.  
Depending on the scope of already installed packages, the software manager will add more packages to resolve all dependencies. Confirm these **Automatic Changes** to perform the installation.
- 7 Confirm the Packages notifications and click **OK**.
- 8 To complete the installation, click **Finish**.
- 9 Perform an initial SMT configuration using YaST > **Network Services** > **Micro Focus SMT Configuration Wizard** or using the command `yast2 smt-wizard` as described in [“SMT Configuration Wizard” on page 10](#).

## 2.3 SMT Configuration Wizard

The two-step **SMT Configuration Wizard** helps you configure SMT, regardless of the installation method used. Launch this wizard using YaST > **Network Services** > **Micro Focus SMT Configuration Wizard** or using the command `yast2 smt-wizard`. You can change the configuration later using the YaST SMT Server Configuration module— see [“Micro Focus SMT Server Configuration” on page 15](#).

- 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 Micro Focus Customer Center credentials in **User** and **Password**. If you do not know your Micro Focus Customer Center credentials, refer to [“Mirroring Credentials” on page 21](#). Test the entered credentials by clicking **Test**. 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 Micro Focus 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.

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

- 2 For security reasons, SMT requires a separate user to connect to the database. With the **Database Password for smt User** widget set the database password for that user. Confirm it in the following field.

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 obsolete addresses.

Then click **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**.
- 4 By default, SMT is set to communicate with the client hosts via a secure protocol. For this, the server needs to have a server SSL certificate.

On the SLES server, the wizard displays a warning to create a certificate if the server does not have the SSL certificate, you must create the SSL certificate manually. For more information, see [Chapter 10, “SMT server Certificate Authority and Server Certificate,” on page 69](#)

## 2.4 Upgrade from Micro Focus SMT 1.1 to Micro Focus SMT 2.0

Micro Focus SMT provides the option of upgrading an existing system to the new version without completely reinstalling it. No new installation is needed.

### 2.4.1 Supported Upgrade Paths

The following table outlines the supported paths for upgrading to Micro Focus SMT.

| Source Version                       | Destination Version                 |
|--------------------------------------|-------------------------------------|
| ◆ OES 2018 SP2 + Micro Focus SMT 1.1 | ◆ OES 2023 + Micro Focus SMT 2.0    |
| ◆ OES 2018 SP3 + Micro Focus SMT 1.1 |                                     |
| ◆ SLES 12 SP4 + Micro Focus SMT 1.1  | ◆ SLES 15 SP4 + Micro Focus SMT 2.0 |
| ◆ SLES 12 SP5 + Micro Focus SMT 1.1  |                                     |

### 2.4.2 Upgrading Micro Focus SMT 1.1 on OES 2018 SP2 or SP3 and SLES 12 SP4 or SP5 to Micro Focus SMT 2.0

- ◆ [“Preparing the Server You are Upgrading” on page 12](#)
- ◆ [“Adding Micro Focus SMT as an Add-on Product” on page 12](#)
- ◆ [“Upgrading to Micro Focus SMT” on page 12](#)

## Preparing the Server You are Upgrading

- ◆ Ensure that the server is updated to the latest available patch before upgrading:
- ◆ It is highly recommended to back up your database and configurations before upgrading. To perform the database backup, perform the following:

- ◆ Log in to your SMT server
- ◆ Create a dump file and save it:

```
mysqldump -u smt -p smt> smt_backup.sql
```

## Adding Micro Focus SMT as an Add-on Product

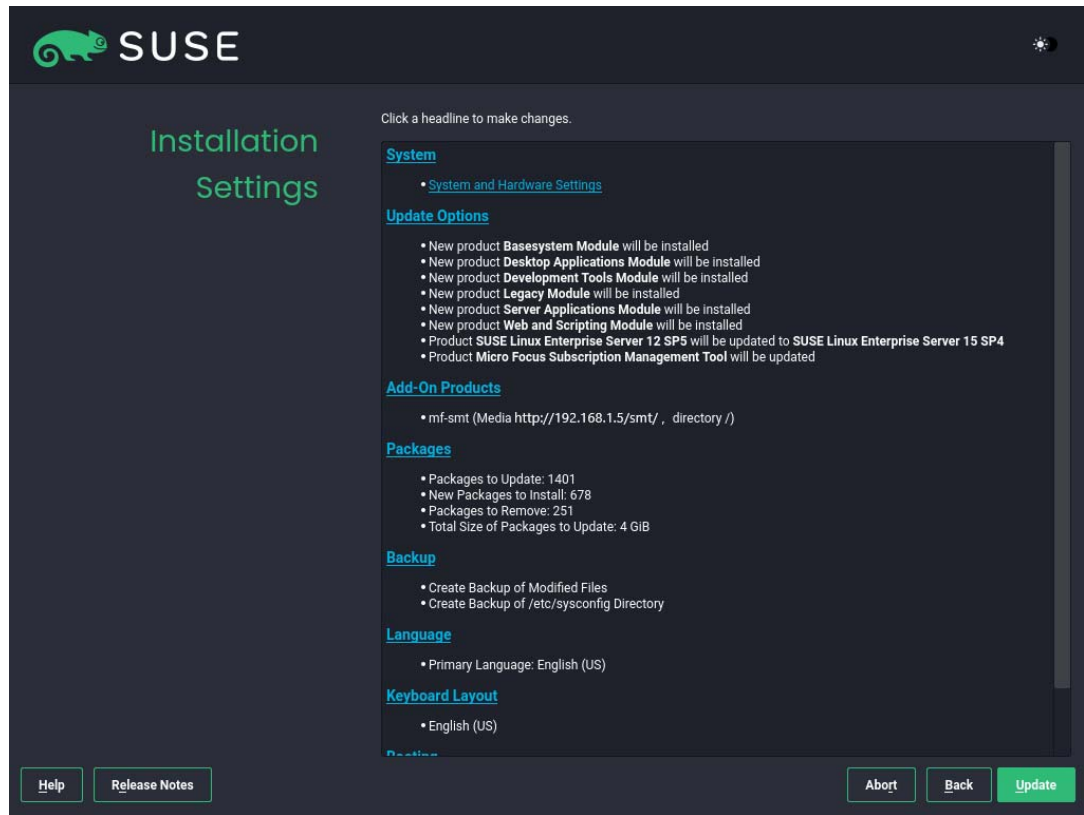
Perform a down server media upgrade of SLES/ OES and add Micro Focus Subscription Management Tool as an add-on product during the upgrade.

## Upgrading to Micro Focus SMT

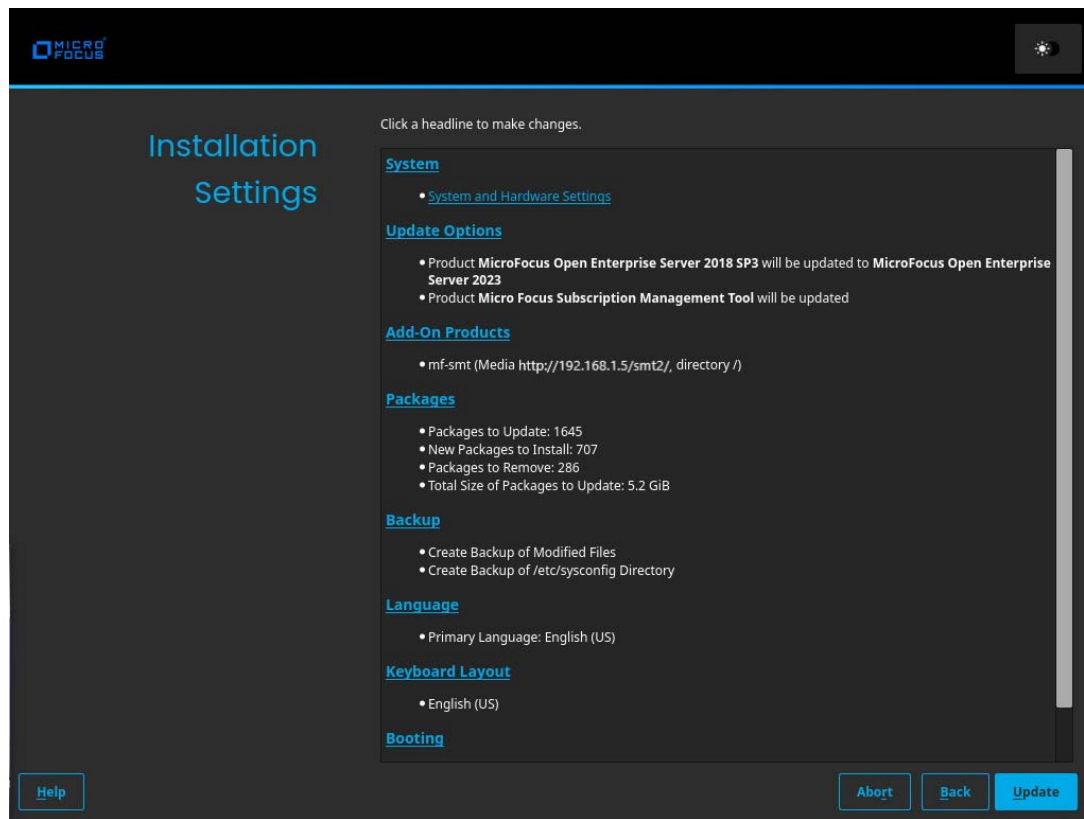
- 1 Log in to your SMT Server.
- 2 Shutdown the SMT Server, insert the OES/ SLES installation media and boot your machine.
- 3 Select the **Upgrade** entry on the boot screen.  
YaST checks your partitions for already installed systems.
- 4 On the Welcome screen, choose **Language and Keyboard** and accept the license agreement. Proceed with **Next**.  
YaST checks your partitions for already installed systems.
- 5 On the Select for Upgrade screen, select the partition to upgrade and click **Next**.  
YaST mounts the selected partition and displays all repositories that have been found on the partition that you want to upgrade.
- 6 On the Previously Used Repositories screen, adjust the status of the repositories: enable those you want to include in the upgrade process and disable any repositories that are no longer needed. Proceed with **Next**.
- 7 On the Registration screen, select whether to register the upgraded system now (by entering your registration data and clicking **Next**) or if to **Skip Registration**. For details on registering your system, see [Registering Your System](#) in SLES or [Specifying Customer Center Configuration Settings](#) in OES Installation guide.
- 8 On the Add On Product screen, select **I would like to install an Add -On product** to add Micro Focus SMT.
- 9 Specify a media type, a local path, or a network resource hosting the repository of MFSMT and continue with the upgrade process.
- 10 (Conditional - SLES) Click **Trust** to import the key found in repository.

## 11 Review the Installation Settings for the upgrade, especially the **Update Options**.

- ◆ For example on SLES 15 SP4:



- ◆ For example on OES 2023:



12 If all settings are as per your requirement, start the installation and removal procedure by clicking **Update**.

13 On completion of the upgrade, check if the `smt.target` service is enabled to start at the system boot:

```
systemctl is-enabled smt.target
```

If the `smt.target` service is not enabled, enable and start the service:

```
systemctl enable --now smt.target
```

# 3 Micro Focus SMT Server Configuration

This chapter introduces the YaST configuration module for the Micro Focus SMT server. You can set and configure mirroring credentials, SMT database passwords, and e-mail addresses to send reports to. It also lets you set the SMT job schedule, and activate or deactivate the SMT service.

To configure Micro Focus SMT with SMT Server Configuration, follow these steps:

- 1 Start the YaST module **SMT Server Configuration** from the YaST control center or by running `yast smt-server` on the command line.
- 2 To activate Micro Focus SMT, check the **Enable Subscription Management Tool Service (SMT)** option in the **Customer Center Access** tab. If you want to disable Micro Focus SMT, uncheck this option. For more information about activating Micro Focus SMT with YaST, see [“Activating and Deactivating SMT with YaST” on page 15](#).
- 3 If the firewall is enabled, check **Open Port in Firewall**.
- 4 In the **Customer Center Configuration** section of 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 [“Setting NU Credentials with YaST” on page 16](#).
- 5 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 [“Setting SMT Database Password with YaST” on page 18](#) and [“Setting E-mail Addresses to Receive Reports with YaST” on page 18](#).
- 6 In the **Scheduled SMT Jobs** tab, set a schedule of periodic SMT jobs, such as synchronization of updates, Micro Focus Customer Center registration, and SMT report generation. For more information, see [“Setting the SMT Job Schedule with YaST” on page 18](#).
- 7 If satisfied with the configuration, click **OK**. YaST adjusts the configuration and starts or restarts necessary services.

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

---

**NOTE:** When the SMT Configuration applies configuration changes, it checks for the existence of the common server certificate. If the certificate does not exist, you may have to generate a new certificate. For more information, see [Chapter 10, “SMT server Certificate Authority and Server Certificate,” on page 69](#).

---

## 3.1 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 Configuration.
- 2 Check the **Enable Subscription Management Tool service (SMT)** option.

---

**NOTE:** If not already configured, organization credentials should be configured before activating SMT. For more information about how to set organization credentials with YaST, see [“Setting NU Credentials with YaST” on page 16](#).

---

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

To deactivate SMT with YaST, follow these steps:

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

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

- ♦ The Apache Web server is started (or reloaded if already running).
- ♦ The MySQL server is started (or reloaded if already running). The SMT user and all necessary tables in the database are created as needed.
- ♦ The schema of the SMT database is checked. If the database schema is outdated, the SMT database is upgraded to conform to the current schema.

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

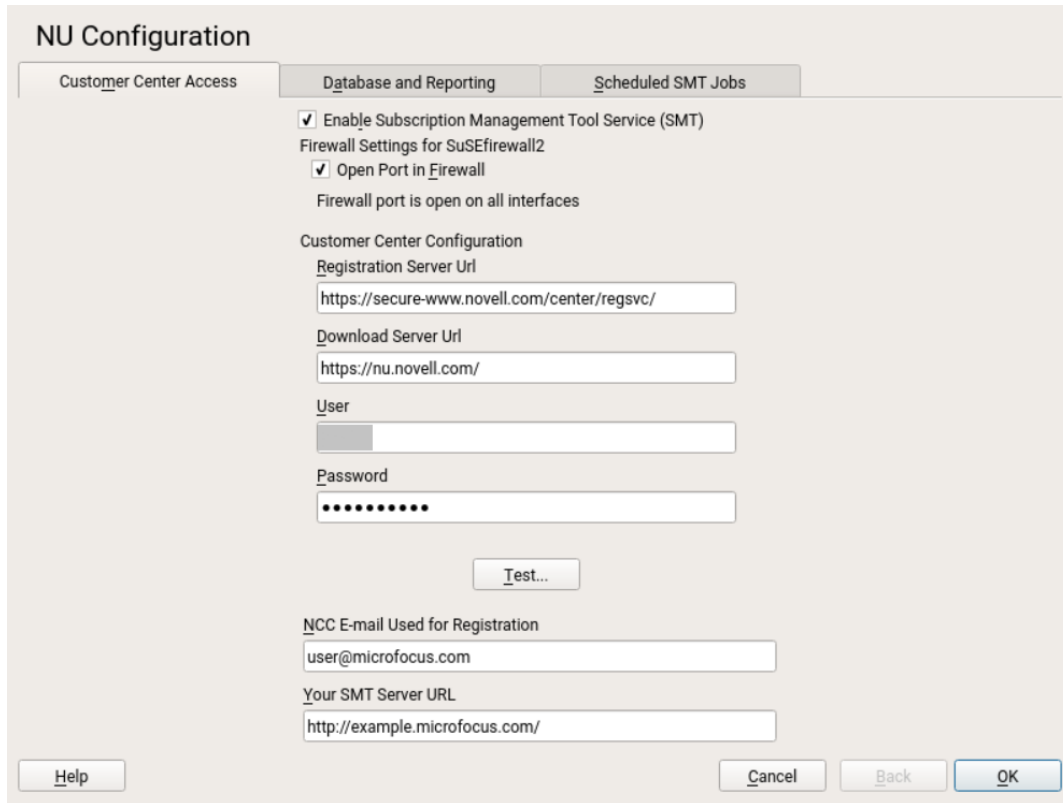
- ♦ The Cron daemon, the Apache server and the MySQL database daemon are reloaded. Neither Apache nor MySQL are stopped, because they may be used for other purposes than the SMT service.

## 3.2 Setting NU Credentials with YaST

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

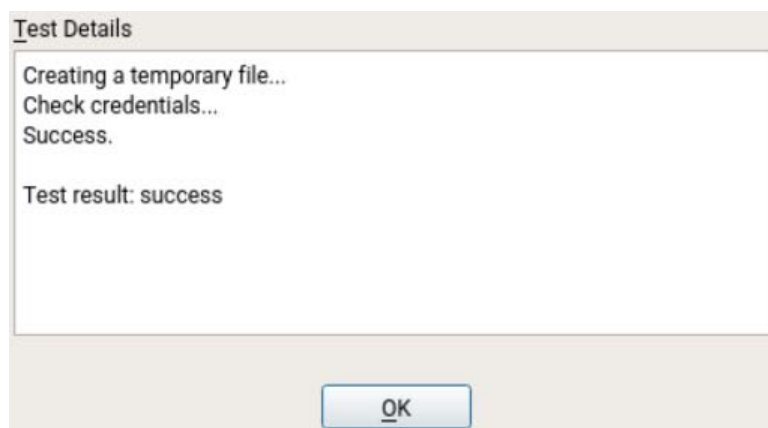


**Figure 3-1** Setting NU Credentials with YaST



- 1 Open the **Customer Center Access** tab of the SMT Configuration (NU Configuration). If the credentials have been already set with YaST or the `/etc/smt.conf` configuration file, they appear in the dialog. Otherwise, the **User** and **Password** fields are blank.
- 2 If you do not have your credentials, visit Micro Focus Customer Center to obtain them. For more details, see [“Mirroring Credentials” on page 21](#).
- 3 Enter your NU username in **User** and the corresponding password in **Password**.
- 4 Click **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 3-2** Successful Test of NU Credentials



- 5 Enter the **NCC E-mail Used for Registration**. This should be the address you used to register to Micro Focus Customer Center.  
Enter **Your SMT Server URL** if it has not been detected automatically.
- 6 Click **OK** or continue with other configurations.

### 3.3 Setting SMT Database Password with YaST

For security reasons, SMT uses its own user in the database. YaST provides an 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 Configuration module.
- 2 Enter the SMT **Database Password for SMT User**. Confirm the password by reentering it and click **OK**, or continue with other configurations.

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

YaST SMT provides an 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 Configuration.
- 2 The list of e-mail addresses is shown in the table. You can **Add**, **Edit**, or **Delete** addresses with the relevant buttons.
- 3 Click **OK** 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.

### 3.5 Setting the SMT Job Schedule with YaST

The SMT Configuration module provides an 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 Micro Focus Customer Center, updates repositories, and downloads new updates.

#### **Generation of Reports**

Generates and sends SMT Subscription Reports to addresses defined in [“Setting E-mail Addresses to Receive Reports with YaST” on page 18](#).

#### **NCC Registration**

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

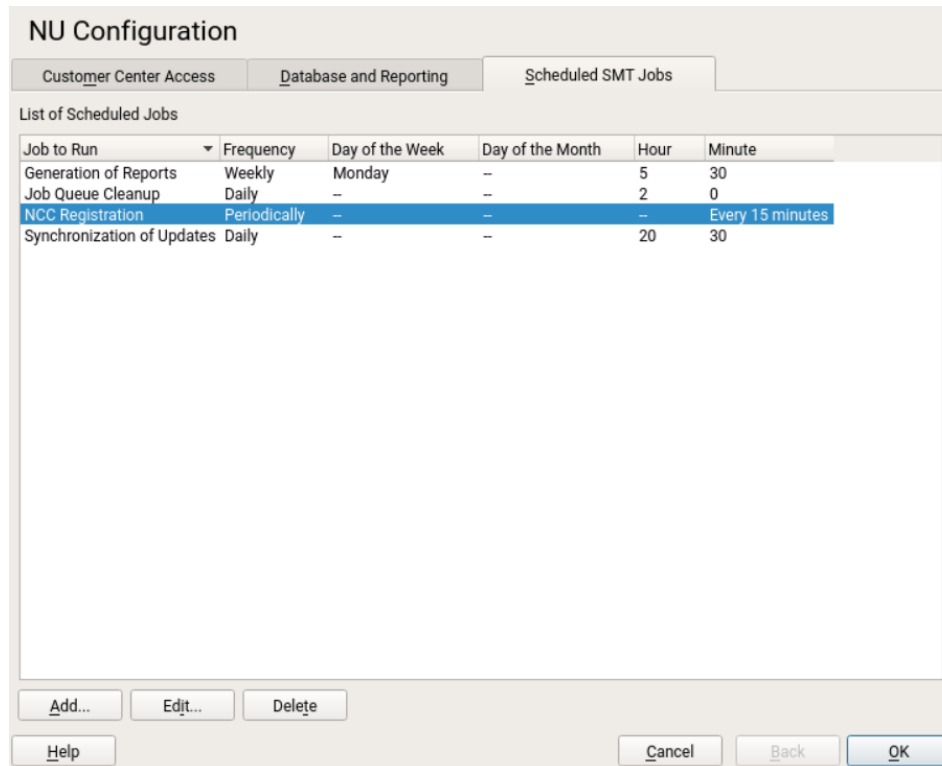
## Job Queue Cleanup

Cleans up queued jobs. It will remove finished or failed jobs from the job queue that are older than 8 days and remove job artifacts that remained in the database as result of an error.

## Uploading Support Configs

Uploading support configs is only available, when the SMT-support package is installed and manually added. To add it, click **Add**, then select **Uploading Support Configs** from the **Job to Run** list.

**Figure 3-3** SMT Job Schedule Configuration



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

- 1 Open the **Scheduled SMT Jobs** tab of the SMT Configuration. 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, click **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**.

click **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 click **Edit**. Then change the desired parameters and click **OK**.

**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

- 4 If you want to cancel a scheduled job and delete it from the table, select the job in the table and click **Delete**.
- 5 Click **OK** to apply the settings and quit the SMT Configuration, or continue with other configurations.

# 4 Mirroring Repositories on the Micro Focus SMT Server

On the Micro Focus SMT server 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.

## 4.1 Mirroring Credentials

Before you create a local mirror of the repositories, you need appropriate organization credentials. To get the credentials, log in to [My Access or Registration Service](#) and find the “Mirror / SMT Login” credentials.

The obtained credentials should be set with the YaST SMT Server Configuration module or manually written to the `/etc/smt.conf` file. For more information about the `/etc/smt.conf` file, see “[/etc/smt.conf](#)” on page 53.

## 4.2 Managing Software Repositories with SMT Command Line Tools

This section describes tools and procedures for viewing information about software repositories available through SMT, configuring these repositories and setting up custom repositories on the command line. For details on the YaST SMT Server Management module, see “[Managing Repositories with YaST SMT Server Management](#)” on page 29.

### 4.2.1 Updating the local SMT database

The local SMT database needs to be updated periodically with the information downloaded from Micro Focus Customer Center. These periodical updates can be configured with the SMT Management module, as described in “[Setting the SMT Job Schedule with YaST](#)” on page 18.

To update the SMT database manually, use the `smt-sync` command. For more information about the `smt-sync` command, see “[smt-sync](#)” on page 46.

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

The database installed with SMT contains information about all software repositories available on Micro Focus Customer Center. However, the used mirror credentials determine which repositories can really be mirrored. For more information about getting and setting organization credentials, see “[Mirroring Credentials](#)” on page 21.

The mirrorability of repositories is determined by retrieving using the provided organization 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 up, which repositories should be mirrored, see [“Selecting Repositories to be Mirrored” on page 22](#).

## 4.2.3 Getting Information About Repositories

Use the `smt-repos` 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 columns show whether the repository can be mirrored and staging is enabled.

Use the `--verbose` option, to get additional information about the 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-repos -m`.

To list only enabled repositories, use the `-o` or `--only-enabled` option: `smt-repos -o` (see [“Listing All Enabled Repositories” on page 22](#)).

### Example 4-1 Listing All Enabled Repositories

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

| Mirror? | ID | Type | Name                           | Target        | Description                                      | Can be Mirrored | Staging |
|---------|----|------|--------------------------------|---------------|--|-----------------|---------|
| Yes     | 1  | nu   | Filr-4-Updates                 | sle-12-x86_64 | Filr-4-Updates for sle-12-x86_64                 | Yes             | No      |
| Yes     | 2  | nu   | iPrint-4-Updates               | sle-12-x86_64 | iPrint-4-Updates for sle-12-x86_64               | Yes             | No      |
| Yes     | 3  | nu   | OES2018-SP1-Pool               | sle-12-x86_64 | OES2018-SP1-Pool for sle-12-x86_64               | Yes             | No      |
| Yes     | 4  | nu   | OES2018-SP1-SLES12-SP3-Pool    | sle-12-x86_64 | OES2018-SP1-SLES12-SP3-Pool for sle-12-x86_64    | Yes             | No      |
| Yes     | 5  | nu   | OES2018-SP1-SLES12-SP3-Updates | sle-12-x86_64 | OES2018-SP1-SLES12-SP3-Updates for sle-12-x86_64 | Yes             | No      |
| Yes     | 6  | nu   | OES2018-SP1-Updates            | sle-12-x86_64 | OES2018-SP1-Updates for sle-12-x86_64            | Yes             | 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-repos repository_name` command. To show information about a repository with a particular name and target, use the `smt-repos repository_name target` command.

To get a list of installation repositories from remote, see [“Listing Accessible Repositories” on page 67](#).

## 4.2.4 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-repos` command.

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-repos -m -e`. To limit the list to only repositories with a particular name, use the `smt-repos -e repository_name` command. To list only a repository with a particular name and target, use the command `smt-repos -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-repos -p product[,version[,architecture[,release]]]
```

For example, to enable all repositories belonging to SUSE Linux Enterprise Server 10 SP4 for PowerPC architecture, use the following command:

```
smt-repos -p Open_Enterprise_Server,2023,x86_64
```

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-repos -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-repos -o -d`. To limit the list to only repositories with a particular name, use the `smt-repos -d repository_name` command. To list only a repository with a particular name and target, use the `smt-repos -d repository_name target` command.

- 2 If more than one repository 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.

## 4.2.5 Deleting Mirrored Repositories

You can delete mirrored repositories that are no longer used. If you delete a repository, it will be physically removed from the SMT storage area.

To delete a repository with a particular name, use the `smt-repos --delete` command. To delete the repository in a namespace, specify the `--namespace dirname` option.

`--delete` lists all repositories, and by entering the ID number or by entering the name and target you can delete the specified repositories. If you want to delete all repositories, enter `a`.

---

**NOTE:** Detecting Repository IDs

Every repository has a sha1sum that you can use as an ID. You can get the repository's sha1sum by calling `smt-repos -v`.

---

## 4.2.6 Mirroring Custom Repositories

Using SMT you can also mirror repositories that are not available at the Micro Focus Customer Center. Those repositories are called “custom repositories”. Use the `smt-setup-custom-repos` command for this purpose. Custom repositories can also be deleted.

When adding a new custom repository, `smt-setup-custom-repos` adds a new record in the database, and sets the `mirror` flag to `true` by default. If needed, you can disable mirroring later.

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 “[smt-list-products](#)” on page 45.
- 2 Run

```
smt-setup-custom-repos --productid product_id \  
--name repository_name --exturl repository_url
```

In this command *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'
```

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.

## 4.3 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 “[/etc/smt.conf](#)” on page 53. If the `MirrorTo` option is not set to the Apache `htdocs` directory `/srv/www/htdocs/`, the following links need to be created. In case the directories already exist, they need to be removed prior to creating the link (the data from that directories will be lost!). `MIRRORTO` has to be replaced with the path defined with `MirrorTo`:

- ♦ `/srv/www/htdocs/repo/$RCE` should point to `MIRRORTO/repo/$RCE/`
- ♦ `/srv/www/htdocs/repo/RPMD` to `MIRRORTO/repo/RPMD/`
- ♦ `/srv/www/htdocs/repo/testing` to `MIRRORTO/repo/testing/` and
- ♦ `/srv/www/htdocs/repo/full` to `MIRRORTO/repo/full/`

The directory specified by the option `MirrorTo` and the subdirectories listed above must exist. Files and directories in `/MirrorTo` as well as the links need to belong to the user `smt` and the group `www`.

For example, if the `MirrorTo` is set to `/mirror/data`:



```

l /srv/www/htdocs/repo/
total 16
lrwxrwxrwx 1 smt www 22 Feb 9 14:23 $RCE -> /mirror/data/repo/$RCE/
drwxr-xr-x 4 smt www 4096 Feb 9 14:23 ./
drwxr-xr-x 4 root root 4096 Feb 8 15:44 ../
lrwxrwxrwx 1 smt www 23 Feb 9 14:23 RPMMD -> /mirror/data/repo/RPMMD/
lrwxrwxrwx 1 smt www 22 Feb 9 14:23 full -> /mirror/data/repo/full/
drwxr-xr-x 2 smt www 4096 Feb 8 11:12 keys/
lrwxrwxrwx 1 smt www 25 Feb 9 14:23 testing -> /mirror/data/repo/
testing/
drwxr-xr-x 2 smt www 4096 Feb 8 14:14 tools/

```

The links can be created using the `ln -s` commands. For example:

```

cd /srv/www/htdocs/repo/
for LINK in \ $RCE RPMMD full testing; do
  ln -s /mirror/data/repo/${LINK}/ && chown -h smt.www ${LINK}
done

```

---

**IMPORTANT:** The `/srv/www/htdocs/repo` directory must not be a symbolic link.

---

#### **IMPORTANT:** Apache and Symbolic Links

By default Apache on SUSE Linux Enterprise is configured to not follow symbolic links. To enable symbolic links for `/srv/www/htdocs/repo/` add the following snippet to `/etc/apache2/default-server.conf` (or the respective virtual host configuration in case you are running SMT on a virtual host):

```

<Directory "/srv/www/htdocs/repo">
  Options FollowSymLinks
</Directory>

```

After having made the change, test the syntax and reload the Apache configuration files to activate the change:

```
systemctl restart apache2.service
```

---

## 4.4 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/repo/testing/` subdirectory.

To mirror a repository to the testing environment, you can use the **Staging** tab in the YaSTSMT 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=register&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 copy it into `repo/testing/`. After tests are finished you can copy the contents 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 must mirror the complete package.

Recommended work flow:

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

## 4.5 Testing and Filtering Update Repositories with Staging

You can test repositories on any clients with `smt-staging` 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 YaST SMT Management module. For more details, see [“Staging Repositories” on page 31](#).

Repositories with staging enabled are mirrored to the `/MirrorTo/repo/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/repo` 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`. `repo/testing/` and `/repo` is called the “default” staging group. You can create additional staging groups as needed with the `smt-staging creategroup` command.

### Enabling Staging

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

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

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 in the “default” staging group enter:

```
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
```

To create testing and production repositories in a named staging group first create the group and then the repositories in this group:

```
smt-staging creategroup Groupname Testingdir Productiondir
smt-staging createrepo --group Groupname Repository_ID --testing
smt-staging createrepo --group Groupname Repository_ID --production
```

This can help you, if you for example, want to combine OES2018-Updates and OES2018-SP1-Updates of the sle-12-x86\_64 architecture into one repository of a group:

```
smt-staging creategroup OES2018-SP0-SP1 test-sp0-sp1 prod-sp0-sp1
smt-staging createrepo --group OES2018-SP0-SP1 \
OES2018-Updates sle-12-x86_64 --testing
smt-staging createrepo --group OES2018-SP0-SP1 \
OES2018-SP1-Updates sle-12-x86_64 --testing
smt-staging createrepo --group OES2018-SP0-SP1 \
OES2018-Updates sle-12-x86_64 --production
smt-staging createrepo --group OES2018-SP0-SP1 \
OES2018-SP1-Updates sle-12-x86_64 --production
```

For group names, these characters are allowed: `_`, `a-zA-Z`, and `0-9`.

### 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-staging 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 a new signing key. This can be done with GPG like this:

```
mkdir some_dir
gpg --gen-key --homedir some_dir
sudo mv some_dir /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 as seen in the `gpg --gen-key` command output, must be written into `/etc/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 signingKeyID \  
> /MirrorTo/repo/keys/smt-signing-key.key
```

In this example, *MirrorTo* stands for the base directory where repositories will be mirrored. Once done, clients can import this key during the registration process.

### **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=register&namespace=testing
```

# 5 Managing Repositories with YaST SMT Server Management

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

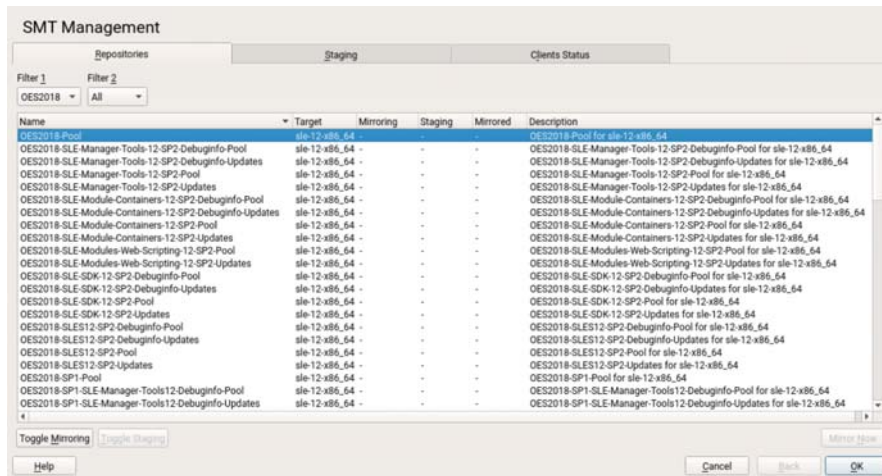
## 5.1 Starting SMT Management Module

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**.
- ♦ Enter `yast2 smt` in the command line as `root`.

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

*Figure 5-1 List of Repositories*



## 5.2 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 using the scrollbars on the window's right side.

## 5.2.1 Filtering Repositories

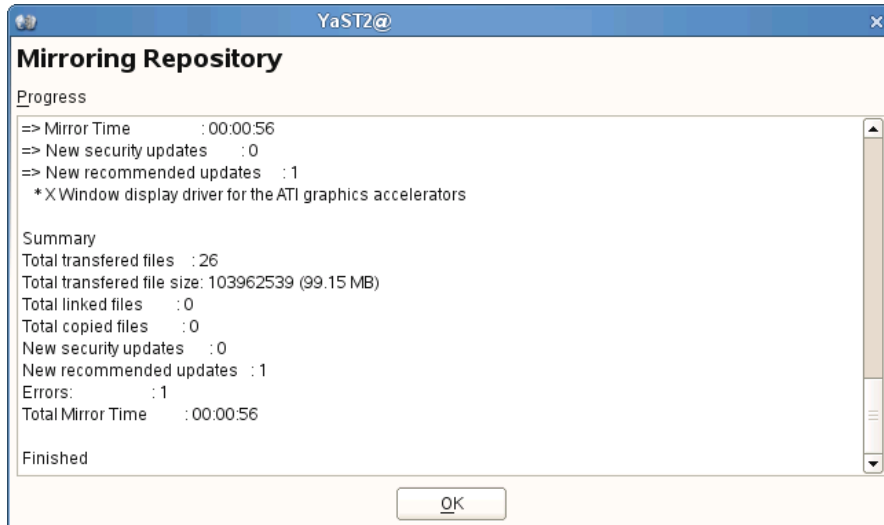
You can also filter out groups of repositories with the **Filter** drop-down list in the upper left part of the window. The filter list items are collected and assembled dynamically from the first word of the repositories' names. If you use a filter to limit the number of displayed repositories, you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

## 5.2.2 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 5-2 Status of Mirroring Process*



## 5.3 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 directory whose staging flag you wish to change.

---

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

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

---

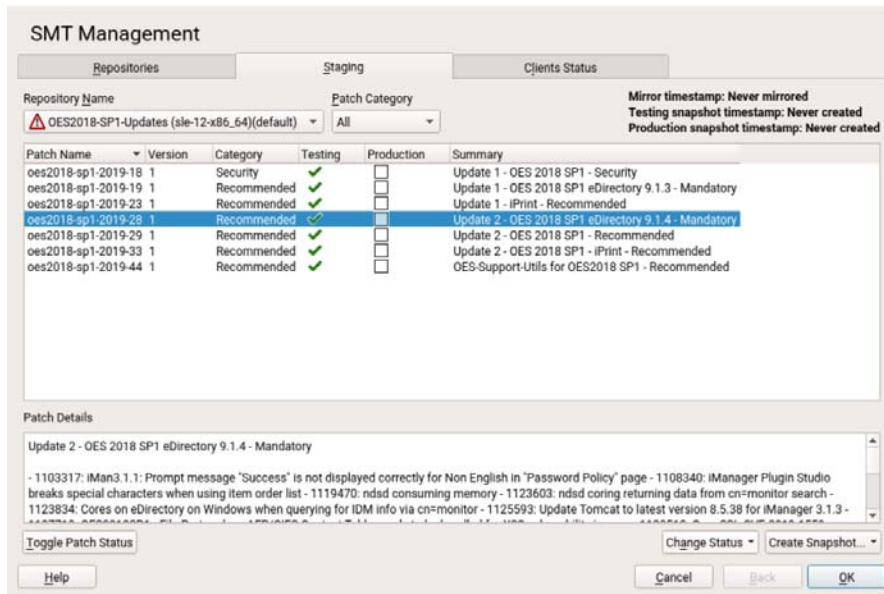
Once you mirror the repositories and make them available for staging, click on the **Staging** tab. In the upper left part of the window, there is a **Repository Name** drop-down list of all repositories which are available for staging. There the repository names have the name of the staging group attached in parentheses. 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 to 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 5-3 Testing Snapshot Created



### IMPORTANT: Creating a Production Snapshot

After you enable *staging* for an update repository, you have to create its *production* snapshot to make it available to the clients. Otherwise the clients will not be able to find the update repository.

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.

## 5.4 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 are either *optional* or *recommended*.

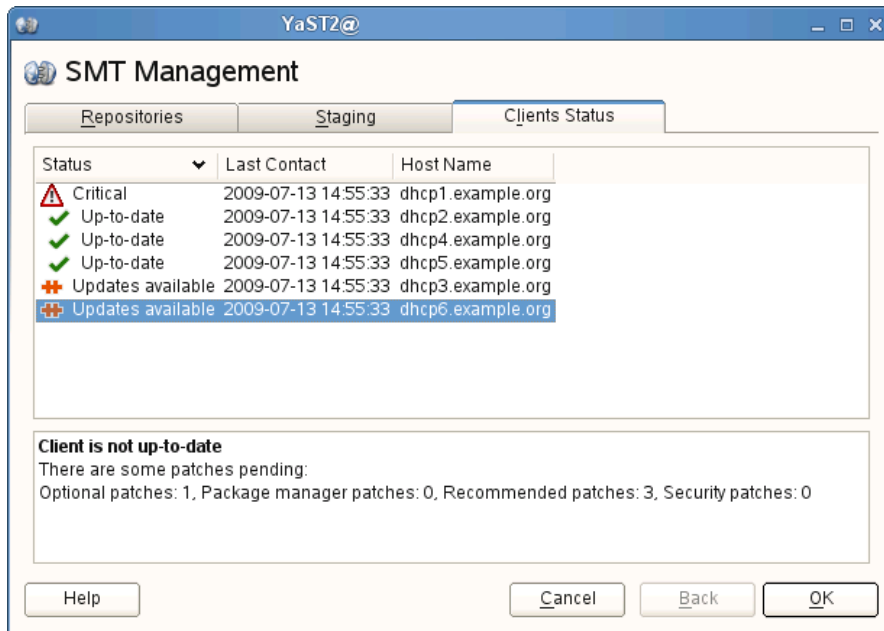
### Critical

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 5-4** Clients Status





# 6 Managing Client Machines with Micro Focus SMT

Micro Focus SMT lets you register and manage client machines on Micro Focus Customer Center. Client machines must be configured to use Micro Focus SMT. For information about configuring clients to use SMT, see [“Configuring Clients to Use Micro Focus SMT” on page 63](#).

## 6.1 Listing Registered Clients

To list Micro Focus 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 Micro Focus SMT server, and the Software **Product** the client uses.

## 6.2 Deleting Registrations

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

```
smt-delete-registration -g Client_ID
```

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

## 6.3 Manual Registration of Clients at Micro Focus Customer Center

The `smt-register` command registers clients at Micro Focus 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.

## 6.4 Scheduling Periodic Registrations of Clients at Micro Focus 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-server`).
- 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 maximum 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 next 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 [“Setting the SMT Job Schedule with YaST” on page 18](#).

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

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

# 7 SMT Reports

This chapter introduces reports based on the SMT and Micro Focus Customer Center data. The reports contain statistics of all the registered machines, products used and all active, expiring or missing subscriptions.

---

**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 Micro Focus Customer Center.

---

## 7.1 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 [“Setting E-mail Addresses to Receive Reports with YaST” on page 18](#). Configuration of the report schedule is described in [“Setting the SMT Job Schedule with YaST” on page 18](#).

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 [“\[REPORT\] Section of /etc/smt.conf” on page 57](#). To set the frequency of reports manually, you can directly edit the `smt-gen-report` lines 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 [“/etc/smt.d/smt-cron.conf” on page 60](#).

## 7.2 Types of SMT Reports

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

**--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 Micro Focus 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 Micro Focus 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 Micro Focus Customer Center at all, use the `--nonccsync` option together with `--local` option.

## 7.3 Report Output Formats and Targets

SMT reports can be printed to the standard output, exported to one or multiple files (in CSV 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. It is recommended to use the `--csv` parameter 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.

---





# 8

## SMT Tools and Configuration Files

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

### 8.1 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 `systemd` `smt.target` script is used for starting, stopping, restarting SMT services, and for checking their status.

#### 8.1.1 SMT JobQueue

There is a new SMT service called SMT JobQueue. It is a system to delegate *jobs* to the registered clients.

To enable JobQueue, the `smt-client` package needs to be installed on the SMT client. The client then pulls jobs from the server via a cron job (every 3 hours by default). The list of jobs is maintained on the server. 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.

---

**NOTE:** Install `mf-smt-client` rpm bundled in Micro Focus SMT Media/ISO. This package is only supported for OES 2018 or later.

---

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 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 `smt-job`, see [“smt-job” on page 43](#).

---

**NOTE:** Overriding the Automatic Creation of Patch Status Job

---

When creating a software push or an update job, normally a non-persistent patch status 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.

---

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.

---

**NOTE:** Job Time Lag 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 time lag of only a few seconds, it will be repeated forever and block other jobs of this client. Therefore, adding jobs with a time lag smaller than one minute is not supported.

---

## 8.1.2 `/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-sync`

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 -L *file***

Specify the file the log will be written to.

**--hostname or -h *name***

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

**--guid or -g *guid***

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

**--severity or -s *level***

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 Micro Focus Customer Center. It will deregister machines from the system. The following options are available:

**--guid *ID* or -g *ID***

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 -v *level***

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 -g *guid***

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

**--jobid or -j *id***

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 -A *id***

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 -t *type***

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.

**--parent *id***

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 -n *name***

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.

### **--targeted *time***

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.

### **--expires *time***

Define when the job will no longer be executed anymore.

### **--timelag *time***

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 -f *format***

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-sync**

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

## **smt-ncc-sync**

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

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

**--fromdir *directory***

Read Micro Focus Customer Center data from a directory instead of downloading it from Micro Focus Customer Center.

**--todir *directory***

Write Micro Focus 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 Micro Focus 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 Micro Focus Customer Center registrations.

The following options are available:

**--local**

Force the creation of a report based on a local calculation without accessing Micro Focus Customer Center data.

**--ncc**

Force the creation of a report based on Micro Focus Customer Center data.

**--nonccsync**

Disable synchronizing with Micro Focus 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 `smt-repos` (or `smt repositories`) to list all available repositories and for enabling, disabling, or deleting 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.

**--delete**  
List repositories and delete them from disk.

**--namespace *dirname***  
Delete the repository in the specified namespace.

**--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 use 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 the 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.example.com/My_Catalog
```

To remove a configured 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 by *Name* and *Target* from the table scheme returned by the `smt-repos` command. Alternatively, it can be specified by its `Repository ID`, which is returned when running the command `smt-repos -v`. 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.

### **listgroups**

List staging groups.

There is always one group available with the name "default". The default group has the path `repo/full`, `repo/testing` and `repo`. With creating a new group, new paths can be specified.

### **creategroup**

Create a staging group. Required parameters are: group name, testing directory name, and production directory name.

### **removegroup**

Remove a staging group. Required parameter is: group name.

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

**--logfile or -L *file 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` commands:

**--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.

**--individually**

Allow or forbid multiple patches (e.g. by category) one by one, that is, as if the `--patch` option had been used on each of the patches.

**--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.

**--group**

Specify on which group the command should work on. The default for `--group` is the name default.

**--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.

**--sort-by-version**

Sort the `listupdates` table by patch version. The higher the version, the newer the patch should be.

**--sort-by-category**

Sort the `listupdates` table by patch category.

## **smt-support**

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

**--incoming or -i *directory***

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 -r *archive***

Delete the specified archive.

**--empty or -R**

Delete all archives in the incoming directory.

**--upload or -u *archive***

Upload the specified archive to SUSE. 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 SUSE.

**--srnum or -s *SR number***

Accept the Novell Service Request 11 digit number.

**--name or -n *name***

Enter the first and last name of the contact, in quotes.

**--company or -c *company***

Enter the company name.

**--storeid or -d *id***

Enter the store ID, if applicable.

**--terminalid or -t *id***

Enter the terminal ID, if applicable.

**--phone or -p *phone***

Enter the phone number of the contact person.

**--email or -e *email***

Enter the email address of the contact.

### 8.1.3 SMT `systemd` Commands

You can manage SMT related services with the standard `systemd` commands:

```
systemctl start smt.target
```

Start the SMT services.

```
systemctl stop smt.target
```

Stop the SMT services.

```
systemctl status smt.target
```

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

```
systemctl restart smt.target
```

Restart the SMT services.

```
systemctl try-restart smt.target
```

Check whether the SMT services are enabled and if so, restart them.

You can enable and disable SMT with the YaST SMT Server module.

## 8.2 SMT Configuration Files

The main SMT configuration file is `/etc/smt.conf`. You can set most of the options with the YaSTSMT Server module. Another important configuration file is `/etc/smt.d/smt-cron.conf`, which contains parameters for commands launched as SMT scheduled jobs.

### 8.2.1 `/etc/smt.conf`

The `/etc/smt.conf` file has several sections. The `[NU]` section contains the update credentials and URL. The `[DB]` section contains the configuration of the MySQL database for SMT. The `[LOCAL]` section includes other configuration data. The `[REPORT]` section contains the configuration of SMT reports.

---

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

### **NURegUrl**

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

### **NUUser**

`NUUser` should contain the username for update service. For information about getting organization credentials, see [“Mirroring Credentials” on page 21](#). You can set this value with the SMT Server.

### **NUPass**

`NUPass` is the password for the user defined in `NUUser`. For information about getting organization credentials, see [“Mirroring Credentials” on page 21](#). You can set this value with the SMT Server.

### **ApiType**

`ApiType` is the type of the service SMT uses, it is `NCC` for Micro Focus Customer Center.

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

The three options defined in the [DB] section are used for configuring the database for SMT. 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* is 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 YaST SMT Server module.

## [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 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 Micro Focus Customer Center. The SMT administrator can set this value with the YaST SMT Server module.

**MirrorTo**

Determines the path to mirror to.

**MirrorAll**

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

**MirrorSRC**

If the `MirrorSRC` option is set to `true`, source RPM packages are mirrored.

**forwardRegistration**

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

**rndRegister**

Specify a delay in seconds before registration of clients at Micro Focus Customer Center. The value is a random number between 0 and 450, generated by the YaST SMT Server module. The purpose of this random delay is to prevent a high load on the Micro Focus 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://proxyexample.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://proxyexample.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 user name 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.

---

### **TIP: Global User Authentication Setting**

If you configure the global proxy settings with YaST, manually copy `/root/.curlrc` to the home directory of the `smt` and adjust the permissions with the following commands as `root`:

```
cp /root/.curlrc /var/lib/smt/  
chown smt:www /var/lib/smt/.curlrc
```

---

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

### **signingKeyID**

Specify the 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 user name 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. An 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 YaST SMT Server module.

### **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. Note that these credentials differ from the mirroring credentials used for SLES 10 and 11. You need to use the NCC login credentials here.

## Example /etc/smt.conf

### Example 8-1 `smt.conf`

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

[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 = false
forwardRegistration = true
rndRegister = 127
# The hook script that should be called before the smt-mirror script
removes its lock
mirror_preunlock_hook =
# The hook script that should be called after the smt-mirror script removed
its lock
mirror_postunlock_hook =
# 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 =
#
# This string is send in HTTP requests as UserAgent.
# If the key UserAgent does not exist, a default is used.
# If UserAgent is empty, no UserAgent string is set.
#
#UserAgent=
# Organization credentials for this SMT server.
# These are currently only used to get list of all available repositories
# from https://your.smt.url/repo/repoindex.xml
# Note: if authenticated as a client machine instead of these mirrorUser,
# the above URL returns only repositories relevant for that client.
mirrorUser =
mirrorPassword =

[REST]
# Enable administrative access to the SMT RESTService by setting
enableRESTAdminAccess=1
# default: 0
enableRESTAdminAccess = 0
# Define the username the REST-Admin uses for login
# default: RESTroot
RESTAdminUser = RESTroot
# Define the password for the REST-Admin (note: empty password is invalid)
# default: <empty>
RESTAdminPassword =

[JOBQUEUE]
# maximum age of finished (non-persistent) jobs in days
# default: 8
maxFinishedJobAge = 8
# comma separated list of JobQueue status IDs that should be interpreted as
successful
# See smt-job --help for more information about possible Status IDs

```

```

# Please note: An empty string will be interpreted as default (1,4).
# default: 1,4
# useful: 1,4,6
jobStatusIsSuccess = 1,4

[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 =

```

## 8.2.2 /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 [“Setting the SMT Job Schedule with YaST” on page 18](#)). 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" .`

## **8.3 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, you may have to generate a new certificate. For more information, see [Chapter 10, "SMT server Certificate Authority and Server Certificate,"](#) on page 69

### **8.3.1 Certificate Expiration**

The common server certificate SMT uses is valid for one year. After that time, a new certificate is needed. If the certificate does not exist, you may have to generate a new certificate. For more information, see [Chapter 10, "SMT server Certificate Authority and Server Certificate,"](#) on page 69.

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.

## 8.3.2 Synchronizing Time Between SMT Server and Clients

The synchronization of time between the SMT server and clients is highly recommended. 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 the time using NTP (network time protocol). Use `yast2 ntp-client` to configure an NTP client. Find detailed information about NTP in the *Administration Guide*.

# 9 Configuring Clients to Use Micro Focus SMT

All Micro Focus products having their update repositories hosted in NU can be configured to register against Micro Focus SMT and download software updates from there, instead of communicating directly with the 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 must enter a path to the server's CA certificate if the certificate was issued by an external certificate authority.

There are several ways to provide the registration information and to configure the client machine to use SMT:

1. Provide the needed information via kernel parameters at boot time ([“Using Kernel Parameters to Access an SMT Server” on page 63](#)).
2. Configure the clients using an AutoYaST profile ([“Configuring Clients with AutoYaST Profile” on page 65](#)).
3. Use the `clientSetup4SMT.sh` script ([“Configuring Clients with the clientSetup4SMT.sh Script” on page 65](#)). This script can be run on a client to make it register against a specified SMT server.
4. You can now set the SMT server URL with the YaST registration module during installation ([“Configuring Clients with YaST” on page 66](#)).

These methods are described in the following sections.

## 9.1 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.

---

**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 an invalid 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.

---

### **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:

```
example.comregurl=https://smt./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:

```
example.comregcert=http://smt./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.crt
```

#### **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:** 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.

---



## 9.2 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 *AutoYaST Guide*. In this section, only SMT specific configuration is described.

To configure SMT specific data using AutoYaST you need to specify the required data with the `reg_server*` properties. Refer to the example below for details.

```
<suse_register>
  <do_registration config:type="boolean">true</do_registration>
  <reg_server>https://smt.example.com/center/regsvc</reg_server>
  <reg_server_cert>http://smt.example.com/smt.crt</reg_server_cert>
  <install_updates config:type="boolean">true</install_updates>
  <slp_discovery config:type="boolean">>false</slp_discovery>
</suse_register>
```

suse\_register Values

**Table 9-1** suse\_register Values

| Attribute Name   | Description   |
|------------------|---|
| ◆do_registration | <ul style="list-style-type: none"><li>◆ Specify whether the system should be registered or not. If set to <code>false</code>, all other options are ignored, and the system is not registered.</li><li>◆ Example:<br/><pre>&lt;do_registration config:type="boolean"&gt;true&lt;/ do_registration&gt;</pre></li></ul> |
| ◆reg_server      | <ul style="list-style-type: none"><li>◆ Specify the URL of the SMT Server</li><li>◆ Example:<br/><pre>&lt;reg_server&gt;https://smt.example.com/ center/regsvc &lt;/reg_server&gt;</pre></li></ul>  |
| ◆reg_server_cert | <ul style="list-style-type: none"><li>◆ URL of the SSL certificate on the server.</li><li>◆ Example:<br/><pre>&lt;reg_server_cert&gt;http:// smt.example.com/smt.crt&lt;/ reg_server_cert&gt;</pre></li></ul> <p><b>NOTE:</b> System will be prompted for trusting the certificate in the automated install.</p>      |

## 9.3 Configuring Clients with the clientSetup4SMT.sh Script

This script allows you to configure a client machine to use an SMT server. It can also be used to reconfigure an existing client to use a different SMT server.

---

**NOTE:** Installation of `wget`

The script `clientSetup4SMT.sh` itself uses `wget`, so `wget` must be installed on the client.

---

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

**1** Copy the `clientSetup4SMT.sh` script from your SMT server to the client machine. The script is available at `<SMT_HOSTNAME>/repo/tools/clientSetup4SMT.sh` and `/srv/www/htdocs/repo/tools/clientSetup4SMT.sh`. You can download it with a browser, using `wget`, or by another means, such as with `scp`.

**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; for example:

```
example.com./clientSetup4SMT.sh https://smt./center/regsvc/
```

In the second case, the script uses the `--host` option followed by the hostname of the SMT server, and `--regcert` followed by the URL of the SSL certificate; for example:

```
./clientSetup4SMT.sh --host smt.example.com \  
--regcert http://smt.example.com/smt.crt
```

In this case, without any “namespace” specified, the client will be configured to use the default production repositories. If `--namespace groupname` is specified, the client will use that staging group.

**3** The script downloads the server's CA certificate. Accept it by pressing `y`.

**4** The script performs all necessary modifications on the client.

**5** The script downloads and asks to accept additional GPG keys to sign repositories with.

**6** The script performs the registration by executing `suse_register` command on the client.

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

```
wget http://smt.example.com/repo/tools/clientSetup4SMT.sh
```

### 9.3.1 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, add before the `Disallow: /` statement:

```
Allow: /repo/keys
```

## 9.4 Configuring Clients with YaST

## 9.4.1 Configuring Clients with YaST

To configure a client to perform the registration against an SMT server use the YaST registration module (`yast2 inst_suse_register`).

Click **Advanced > Local Registration Server** and enter the name of the SMT server plus the path to the registration internals (`/center/regsvc/`), e.g.:

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

After confirmation the certificate is loaded and the user is asked to accept it. Then continue.

---

### **WARNING:** Staging Groups Registration

If a staging group is used, make sure that settings in `/etc/suseRegister.conf` are done accordingly. If not already done, modify the `register=` parameter and append `&namespace=namespace`. For more information about staging groups, see [“Staging Repositories” on page 31](#).

Alternatively, use the `clientSetup4SMT.sh` script (see [“Configuring Clients with the clientSetup4SMT.sh Script” on page 65](#)).

---

## 9.5 Registering Clients Against SMT Test Environment

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

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

For more information about using SMT with a test environment, see [“Using the Test Environment” on page 25](#).

## 9.6 Listing Accessible Repositories

To retrieve the accessible repositories for a client, download `repo/repoindex.xml` from the SMT server with the client's credentials. The credentials are stored in `/etc/zypp/credentials.d/NCCredentials` on the client machine. Using `wget`, the command for testing could be as follows:

```
wget https://USER:PASS@smt.example.com/repo/repoindex.xml
```

`repoindex.xml` returns the complete repository list as they come from the vendor. If a repository is marked for staging, `repoindex.xml` lists the repository in the full namespace (`repos/full/$RCE`).

To get a list of all repositories available on the SMT server, use the credentials specified in the `[LOCAL]` section of `/etc/smt.conf` on the server as `mirrorUser` and `mirrorPassword`.



# 10 SMT server Certificate Authority and Server Certificate

This chapter describes the certificate management for Micro Focus SMT 2.0.

On OES 2023, SMT uses eDirectory Server and CA certificates. On SLES 15 SP4, an administrator has to manually generate certificates or use their own organizational policy based certificates.

## 10.1 Create a SMT server CA and server certificate

The steps in this section cover certificate generation for an organization called “acme” and servers within the acme organization, the administrator has to change the organization details while generating the certificates.

- 1 Execute the following commands to generate certificate on SMT Server. If an administrator wants to use already issued certificates, go to [Step 2 on page 69](#).

- ♦ Generate CA certificate for an organization:

```
~#: openssl req -nodes -x509 -newkey rsa:2048 -days 365 -keyout acmekey.key -out acmekey.pem -subj "/O=acme"
```

- ♦ Generate certificate signing request (CSR) for the server:

```
~#: openssl req -nodes -newkey rsa:2048 -keyout acmeserver.key -out acmeserver.csr -subj "/O=acme/OU=Servers/CN=`hostname -f`"
```

- ♦ Before signing the CSR to get the MFSMT Server Certificate, create a text file with DNS Names and IP Addresses of the SMT server to add SAN extension to the certificate. Adding all the DNS names and IP Addresses to certificate helps connecting to SMT Servers using host name or IP Address of the SMT Server.

```
# extfile.txt
```

```
subjectAltName = DNS:smt2-server.labs.blr.novell.com, DNS:smt2-server, IP:192.168.10.1
```

- ♦ Sign the CSR to get the server certificate:

```
~#: openssl x509 -req -in acmeserver.csr -days 365 -CA acmekey.pem -CAkey acmekey.key -CAcreateserial -out acmeserver.pem -extfile extfile.txt
```

- 2 Set-up SMT server to use CA Certificate.

Following are the steps to use the newly generated CA certificate in [Step 1 on page 69](#). The same steps are applicable if you want to use your own certificates. Replace certificate files with your own certificate paths.

- ♦ Copy the newly generated CA Certificate:

```
~#: cp acmeqa.pem to /etc/ssl/certs
```

- ◆ Generate hash for new CA certificate using command:

```
~#: c_rehash
```

- ◆ Execute the command to verify if the hash is generated:

```
~#: ls /etc/ssl/certs/ | grep acmeqa.pem
```

This lists the new symlink to `acmeqa.pem`

- 3** The following are the steps to use the newly generated server certificate in [Step 1 on page 69](#). Same steps are applicable if you want to use your own certificates. Replace certificate files with your own certificate paths.

- ◆ Modify `/etc/apache2/vhosts.d/vhost-ssl.conf` and change the below path specifying the location of the certificate file and key file.

```
SSLCertificateFile <path to certificate file>/acmeserver.pem
```

```
SSLCertificateKeyFile <path to certificate file>/acmeserver.key
```

- 4** Configure SMT server to use the new CA Certificate `acmeqa.pem`

Execute the following command:

```
~#: systemctl restart smt.target
```

The CA certificate `/etc/ssl/certs/acmeqa.pem` to `/srv/www/htdocs/smt.crt` is copied.

- 5** Execute the following command to verify the certificate on SMT server:

```
~#: openssl s_client -connect <IP or Hostname of the SMT Server>:443
```

This command should display the content of the newly configured server certificate

To verify the CA certificate run command

```
~#: openssl x509 -in /srv/www/htdocs/smt.crt -text
```

The newly configured CA certificate content is displayed.

On configuring SMT Server with new CA and Server Certificates successfully, the clients can successfully communicate and register with the SMT Server. For more information see [Chapter 9, “Configuring Clients to Use Micro Focus SMT,” on page 63](#).