

# Subscription Prescription

This article first appeared in the October 2008 issue of *Novell Connection* magazine.

*Novell IS&T manages 1,500 SUSE Linux Enterprise Desktop desktop subscriptions with the Subscription Management Tool for SUSE Linux Enterprise*

If you operate a significant Linux environment you already know that managing registrations, subscriptions and software updates can easily eat up time and network bandwidth. It can complicate your firewall policy compliance and even compromise security.

But did you know that a convenient, easy-to-use and fully supported tool for Linux subscription management is now bundled with Service Pack 2—one that's tightly integrated with both Novell Customer Center and Novell Update. The default configurations for new SUSE Linux servers and desktops require regular, direct connections to Novell Customer Center and Novell Update so that each system can register for entitlements and download updates.

The Subscription Management Tool for SUSE Linux Enterprise is a package proxy system that mirrors Novell Customer Center and Novell Update at your site, inside your firewall. It provides a local update repository identical to Novell Update, and a registration target synchronized with Novell Customer Center. By downloading updates once for subsequent distribution over the local network, Subscription Management Tool reduces WAN congestion and enables more restrictive firewall policies. It lets you manage registrations and entitlements and monitor subscription use for all your SUSE Linux Enterprise servers, desktops and point-of-sale devices running Service Pack 2 or subsequent releases, while preserving the familiar Novell Customer Center interface and all of its functionality. (See Figure 1.)

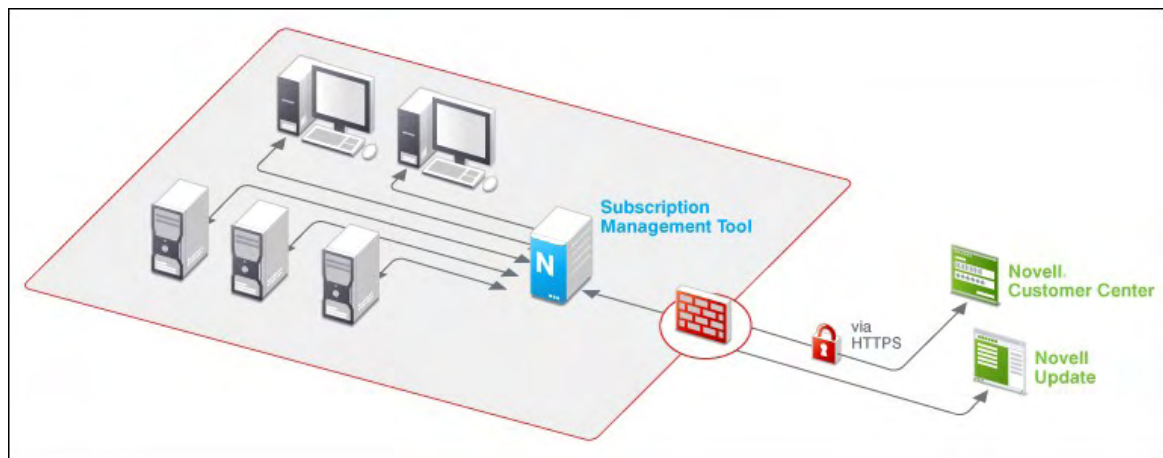
The Subscription Management Tool is in no way a substitute for an enterprise systems management solution such as Novell ZENworks Linux Management. It lacks the granular distribution controls and broader systems management capabilities. But where lightweight rights tracking and update management are the primary objectives, Subscription Management Tool offers a streamlined, cost-effective solution.

## > Deploying Subscription Management Tool at Novell

Lightweight subscription management was exactly the challenge facing Novell IS&T, which manages more than 1,500 SUSE Linux Enterprise Desktop systems located in three major campus sites and more than 20 sales offices worldwide. The group's legacy solution, a custom Novell ZENworks Linux Management build, was diverging from the product sold and supported by Novell.

With a largely technical user population, the team didn't need focused desktop management, but did require a new update management solution that would provide staging and testing both for SUSE Linux Enterprise and other Novell applications, including Novell GroupWise, Novell GroupWise Instant Messenger and Novell iFolder. When Subscription Management Tool became part of Service Pack 2, the team gave it a try. Simon Crute, Linux Services Manager, describes the installation process they followed and the customizations they developed for the distributed environment at Novell.

**Figure 1:** *The Subscription Management Tool for SUSE Linux Enterprise is a proxy server that mirrors Novell Customer Center and Novell Update, providing local registration, subscription management and update distribution services.*

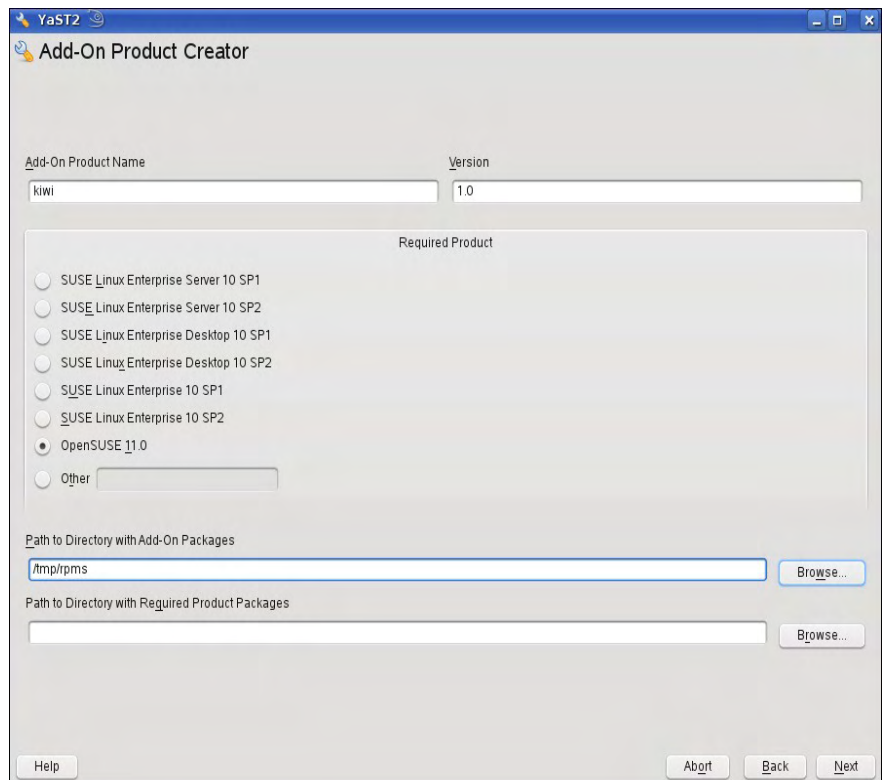


> **Subscription Management Tool Step One: Install and Configure**

“We began by setting up a new SUSE Linux Enterprise server,” Simon explains. “We installed Service Pack 2 on a dual-core Xeon system with 4GB of RAM and 750 GB of disk space. Subscription Management Tool installs as an Add-On Product in YaST. Go to Software and Add-On Product, then select Local Directory and ISO Image and browse to the correct ISO. YaST then offers to install the product and resolve all dependencies for you.” (See Figure 2.)

Once installed, Subscription Management Tool needs to be activated against Novell Customer Center. This requires the acquisition of mirroring credentials from Novell Customer Center through a process described in detail in [TID 3612166](#), which is available [online](#).

Figure 2: *Subscription Management Tool installs as an Add-On Product in YaST.*



> **Step Two: Selecting Which Catalogs to Mirror**

With Subscription Management Tool running and connected to Novell Customer Center, the next step was to decide which update catalogs to mirror locally. Because Service Pack 2 had just been released, IS&T was supporting a mix of SP1 and SP2 desktops and servers. “Several catalogs are available for each product,” Crute said. “There’s an online catalog of packages for migrating older systems to the current service pack, a pool catalog of base packages for the current service pack, and an update catalog containing packages released since the current service pack’s initial release.”

The team elected to carry all three catalogs for SUSE Linux Enterprise Server 10-SP1 and SP2, SUSE Linux Enterprise Desktop 10-SP1 and SP2, for both the i586 and x86-64 architectures. The command that controls catalog mirroring in Subscription Management Tool is **smt-catalogs**. Run by itself, it displays the available catalogs and architectures. The **smt-catalogs | grep sled** command displays all SUSE Linux Enterprise Desktop catalogs, and the **smt-catalogs sled10-sp1** command shows the architectures available for each.

Enabling mirroring for all three SUSE Linux Enterprise Desktop 10-SP2 catalogs required three commands:

```
smt-catalogs sled10-sp2-pool -e
```

```
smt-catalogs sled10-sp2-online -e
smt-catalogs sled10-sp2-updates -e
```

Each command elicits a prompt listing the available architectures.

> **Step Three: Creating Custom Catalogs**

The IS&T team also wanted the new update server to support several internal products and openSUSE RPMs. To do this, they first created a package repository from which to mirror (both YUM and ZYPP work well) using the Add-On Product module in YaST. Then they decided from which SUSE Linux Enterprise products these packages should be available.

“The **smt-list-products** command shows you all the products that Subscription Management Tool knows about,” Crute explained. “Depending on your subscriptions to Novell Customer Center, you’ll see quite a few different versions of SUSE Linux Enterprise Desktop, and you need the product ID to set up a custom catalog. So, for instance, if you’re in a country where VideoLAN is legal, you can instruct Subscription Management Tool to mirror it by entering the following command:

```
smt-setup-custom-catalogs --
productid 431 --name
'VLC_SLED_10_SP1' --exturl \
'http://download.videolan.or
```

```
g/pub/videolan/vlc/SuSE/10.1
' \ --description 'VideoLAN
repository for SLED 10 SP1
do not use in USA or EU
countries'
```

“If you’ve created a custom repository you want to mirror, simply enter the URL. If you’re not sure the IP address will be stable over time, you may want to create a separate Apache vhost on the Subscription Management Tool server and copy the repository contents there from the build location.”

To sign a repository before importing it into Subscription Management Tool, run the following commands from the repository directory:

```
gpg --default-key [keyid] -a
--detach-sign repodata/repomd.xml
gpg -a --export [keyid] > repodata/
repomd.xml.key
```

To get the public key into the client, either import it with a post-install script or modify the installation boot media.

#### > **Step Four: Creating a Staging Environment**

Like any rational administrator, the Novell team also wanted a staging environment where it could test new packages thoroughly before moving them into production. Subscription Management Tool doesn’t support that capability in its default configuration, but it is easy enough to add with a bit of Apache config file work.

“The idea is to create two vhost definitions,” Crute said. “One for smt.company.com, the other for smt-stage.company.com. Both are on the same box, but a little [Apache redirection](#) sends the production and staging requests to different directories.

To selectively move tested RPMs into production without disturbing those still under evaluation, or disrupting the physical repositories on disk, the Novell team wrote a script that makes daily backups of the staging catalogs, naming them by date. To minimize the time and disk space required, a link copy is made (see [man cp](#) for details) which, in essence, simply duplicates the inodes or directory information, leaving the actual files unchanged. [Click here to see the complete script code.](#)

Once administrators are satisfied with a set of updates, they simply rename the directories from the backup location into production. A daily e-mail is circulated that summarizes the differences between staging and production. [Click here to see the e-mail creation script.](#)

#### > **Stage Five: Migrating Existing Systems**

With its Subscription Management Tool server functioning correctly, the Novell team then repointed its client systems to the new source for registration and update services. Several methods are available. For new

installations, the following script can be added at boot time:

```
regurl=https://[servername]/center/regsvc
regcert=done
```

The same script can be easily adopted for PXE-based network installs, and in the physical boot media used in media-based installations. For installations via AutoYaST, [another script fragment](#) can also be used to register new machines. And finally, Subscription Management Tool includes the following script, which can be used to help migrate existing machines:

```
/
usr/share/doc/packages/smt/clientSetup4SM
T.sh https://[servername]/center/regsvc
```

This can be incorporated into existing login scripts, or wrapped and distributed for users to run. If you have an existing update infrastructure in place, you can even create a package around this script that moves users to the new service automatically.

#### > **Step Six: Transparently Caching Updates at Distributed Locations**

The final decision required in deploying Subscription Management Tool, at least in an installation with multiple locations under management, is how to best provide distributed access to the update service. Deploying separate Subscription Management Tool servers in each location increases the administrative overhead, sacrifices transparency, and reduces bandwidth efficiency, particularly for users who travel between branches.

Instead, the Novell team decided to cache update transfers locally using the Squid caching proxy server. This required configuring a Squid server at each location to accept transparent proxying, then configuring the router at the Subscription Management Tool site to send Subscription Management Tool-bound traffic to the local proxies. On Cisco routers this is known as a policy route. Details for configuring Squid can be found at <http://itdp.org/HOWTO/TransparentProxy.html>.

The Novell team is using [the following aging details](#) for its Squid servers. These are provided as an example only, with no claim of correctness or best practice.

#### > **Managing 1,500 SUSE Linux Enterprise Desktop Subscriptions with a Free Tool**

Today, Novell IS&T is managing registrations and subscription updates for its 1,500 SUSE Linux Enterprise Desktops with a high degree of automation, using the Subscription Management Tool proxy server that’s included in every copy of SUSE Linux Enterprise 10 Service Pack 2. More than 700 SP1 users have successfully used the service to update their machines to SP2. For more information, see the Subscription Management Tool home page at <http://www.novell.com/linux/smt/>. **N**

## Complete Script Files

### Script File #1

```
/etc/apache2/vhost.d/smt.conf
<VirtualHost *:80>
ServerAdmin [deleted]
ServerName smt.novell.com
alias /repo/ /srv/www/htdocs/production/repo/
alias /repo /srv/www/htdocs/production/repo
</VirtualHost>
```

```
/etc/apache2/vhsot.d/smt-stage.conf
<VirtualHost *:80>
ServerAdmin [deleted]
ServerName smt-staging.provo.novell.com
alias /repo/ /srv/www/htdocs/repo/
alias /repo /srv/www/htdocs/repo
</VirtualHost>
```

Somewhat confusingly, this means that the default directory where all the Novell-supplied Subscription Management Tool scripts work, and where all the mirrored RPMs end up, is the staging area /srv/www/htdocs/repo. If you think about it, this makes sense. You wouldn't want to change the core Subscription Management Tool scripts, because any updates to Subscription Management Tool might overwrite your own changes. The production area is then actually /srv/www/htdocs/production/repo

### Script File #2

```
#!/bin/bash
#
#script to keep multiple copies of the staging catalogue
to enable us to move a
#specific day's staging into production after testing.
#the backups are created as "hard link copies". See the
cp man page and the --link option.

# the number of copies to keep
COPIES=7
#where the staging catalogue are kept
STAGING_LOCATION=/srv/www/htdocs/repo
#where the backup catalogues are kept.
# Must be the same file system as
STGAGING_LOCATION.
# Do not use spaces in name
BACKUP_LOCATION=/srv/www/staging_backups
#the basename of the backup directory
BASENAME=staging-backup

#Test BACKUP_LOCATION
set -- $BACKUP_LOCATION
if [[ $1 != $BACKUP_LOCATION ]] ; then
```

```
echo "BACKUP_LOCATION has bad characters in it,
probably spaces, or IFS is set incorrectly"
echo "Failure to fix this may result in data loss on your
server"
exit
fi
#create new backup name.and check dir does not exist
NEW_NAME=$BACKUP_LOCATION/$BASENAME-$
(date +%F)
if [[ -e $NEW_NAME ]] ; then
echo "$NEW_NAME" already exists. Quitting.
exit
fi
#find out how many backups there are and remove old
ones.
BACKUP_DIRS=$(ls -d $BACKUP_LOCATION/
$BASENAME*)
NO_BACKUP_DIRS=$(ls -d $BACKUP_LOCATION/
$BASENAME* | wc -w)
#if there are more than COPIES, remove the oldest
while [[ $NO_BACKUP_DIRS -gt $COPIES-1 ]] ; do
set -- $BACKUP_DIRS
echo "Removing old backup: $q"
rm -rf $1
BACKUP_DIRS=$(ls -d $BACKUP_LOCATION/
$BASENAME*)
NO_BACKUP_DIRS=$(ls -d $BACKUP_LOCATION/
$BASENAME* | wc -w)
done
export COPIES NO_BACKUP_DIRS
#copy the newest one.
echo "Link-Copying staging to staging backup"
cp --archive --link $STAGING_LOCATION $NEW_NAME
```

### Script File #3

```
#!/bin/bash
#
#bash script to show the differences between two
repositories.
#
#
STAGING_LOCATION=/srv/www/htdocs/repo
BACKUP_LOCATION=/srv/www/staging_backups
BASENAME=staging-backup
PRODUCTION=/srv/www/htdocs/production/repo

REPOS[1]="\$RCE/SLED10-SP2-Updates/sled-10-
i586/rpm"
#
echo
"=====
=====
====="
echo "This script will report the differences between
staging and production"
```

```
echo -n "Running at "  
date  
echo  
"=====  
=====  
====="
```

```
for REPO in ${REPOS[*]}; do  
echo "Looking at $REPO"  
echo "-----"  
echo -n "Total new RPMs "  
diff -r $PRODUCTION/$REPO $STAGING_LOCATION/  
$REPO |cut -f 4 -d " " |grep -v .patch.rpm | grep -v  
.delta.rpm | wc -l  
echo "RPM list follows"  
echo  
diff -r $PRODUCTION/$REPO $STAGING_LOCATION/  
$REPO |cut -f 4 -d " " |grep -v .patch.rpm | grep -v  
.delta.rpm  
done
```

**Script File #4**

```
<customer_center>  
<do_registration  
config:type="boolean">true</do_registration>  
<reg_server>https://  
[servername]/center/regsvc</reg_server>  
<register_regularly  
config:type="boolean">>false</register_regularly>
```

```
<registration_data>  
<email>ist_autoyast_installed@novell.com</email>  
</registration_data>  
<submit_hwdata  
config:type="boolean">true</submit_hwdata>  
<submit_optional  
config:type="boolean">true</submit_optional>  
</customer_center>
```

**Script File #5**

```
refresh_pattern ^ ftp: 1440 20%  
10080  
refresh_pattern ^ gopher: 1440 0%  
1440  
refresh_pattern (cgi-bin|\\?) 0 0% 0  
#long age on rpms cos an rpm file name should not  
change  
refresh_pattern -i smt\.foo\.com/. *rpm 4320 100% 43200  
reload-into-ims  
#much shorter age on repodata cos it does change  
refresh_pattern -i smt\.foo\.com/. *repodata 10  
20% 60  
refresh_pattern -i smt-stage\.foo\.com/. *rpm 4320 100%  
43200 reload-i  
nto-ims  
refresh_pattern -i smt-stage\.foo\.com/. *repodata* 10  
20% 60  
refresh_pattern . 0 20%  
4320
```