# SMART Notebook™ 11

## Linux® operating systems

**Installation and administration guide**

SMART®

# Product registration

If you register your SMART product, we'll notify you of new features and software upgrades.

Register online at smarttech.com/registration.

Keep the following information available in case you need to contact SMART Support.

Serial number: _____

Date of purchase: _____

# Contents

# Chapter 1

# Installing SMART Notebook 11 for Linux

## System requirements

- Pentium® 4 or higher processor

- 1 GB of RAM (2 GB recommended)

- 2.5 GB of free hard disk space for minimum installation

- Linux kernel 2.6.31 or later

- X Window System™ 11 release 6 (X11R6) or later that supports XRender and Xinerama™

- glibc 2.10.1

- libstdc++6

- GNOME™ 2.28 or KDE® 4.3.5 libraries or later (recommended)

# Linux distributions tested

- Debian® 7.2

- Ubuntu® 12.04.3 LTS

- openSUSE® 11.4

> **NOTE**
>
> 64-bit Linux distributions aren't supported.

If your computer doesn't meet these requirements, you might need to install an earlier version of SMART Notebook™ collaborative learning software. Visit smarttech.com/support for details.

# Installation packages

You can install SMART Notebook 11 software for Linux operating systems using one of the following:

- Debian packages

- RPM® packages

## Debian packages

Debian packages containing SMART Notebook software and SMART Product Drivers are available.

Debian packages are designed to allow automated, silent and unattended deployment over the network from a central location to remote target computers. This includes software pre-activation, as well as local/distributed mirroring of Gallery material.

## RPM Packages

RPM packages containing SMART Notebook software and SMART Product Drivers are also available.

Like Debian packages, RPM packages are designed to allow automated, silent and unattended deployment over the network from a central location to remote target computers. This includes software pre-activation, as well as local/distributed mirroring of Gallery content.

# Before installing SMART Notebook software

Before you install SMART Notebook 11 software, you must obtain a product key.

## Obtaining a product key

**To obtain a product key**

1. Go to smarttech.com/NB10ProductKey.

2. Type your SMART product's serial number, choose your organization type, and then click **Continue**.

3. Fill out the customer information form.

4. Continue the download and installation procedures while you wait for SMART to e-mail you a product key.

   During the installation, use the product key you receive to activate your software.

   **N O T E S**

   - You can activate every copy of SMART Notebook 11 software using the same product key.

   - Until you activate the software, you have only a 30-day trial version.

# Chapter 2

# Installing SMART Notebook 11 software from Debian packages

## Introduction

This section shows how you can install SMART Notebook software and SMART Product Drivers software shipped as Debian package files.

> ☞ **IMPORTANT**
>
> Before you install SMART Notebook 11 software, you must obtain a product key. See *Before installing SMART Notebook software* on page 3.

# Downloading SMART Notebook software Debian packages

**To download SMART Notebook 11 software**

1. Go to smarttech.com/downloads.

2. In the *SMART Notebook collaborative learning software* area, select **Choose a version**, and then select **11 for Linux** from the drop-down list.

3. Select the **Download Debian packages** section, and then click **Download** next to *Debian plain files*.

4. Fill out the *Download Software* form, and then click **Start Download**.

   The end user license agreement (EULA) appears.

5. Accept the end user license agreement (EULA)

   A file download dialog box appears.

6. Select **Save File**, and then click **OK**.

7. Browse to where you want to save the file, and then click **Save**.

   The *Download Complete* dialog box appears.

8. Click **Close**.

9. Browse to where you saved the file, and then double-click

   **SMART_software_deb_files.tar.gz**.

   Your file compression program opens.

10. Decompress **SMART_software_deb_files.tar.gz**, and then browse to the decompressed folder.

# About the Debian archive

The **SMART_software_deb_files.tar.gz** archive contains Debian installation packages for SMART Notebook software and SMART Product Drivers, as well as a customization template file.

To prepare the packages for use, you must perform the following steps:

- Verify the integrity of the files you received.

- Obtain a product key. See *Before installing SMART Notebook software* on page 3.

- Select a master signing key to sign the relevant data with the GNU Privacy Guard (GPG).

- Edit the customization template file to include your product key and your GPG key ID, to select the content galleries you wish, etc.

- Execute the automated procedure that merges the customization information into the packages.

> ☞ **I M P O R T A N T**
>
> To activate your software for a silent remote deployment and to customize the installation, you must edit the customization file named **customization** before you install the software.

The installation data can be shipped in two forms: package set or Debian archive.

## Before you start

Ensure that the packages dpkg, dpkg-dev, dpkg-deb and binutils are installed. You also need GPG, md5sum, sha1sum, sha256sum and utilities like sed, cut and tr.

# Installing a package set

SMART provides an archive named **SMART_software_deb_files.tar.gz** containing the following:

- eight Debian package files

    - smart-common_<version>_i386.deb

    - smart-hwr_<version>_i386.deb

    - smart-languagesetup_<version>_i386.deb

    - smart-activation_<version>_i386.deb

    - smart-gallerysetup_<version>_i386.deb

    - smart-notebook_<version>_i386.deb

    - smart-product-drivers_<version>_i386.deb

    - nwfermi-<version>_i386.deb (for SMART Board® 480 interactive whiteboards only)

- eight corresponding integrity information files

    - smart-common_i386_sha256sum.asc

    - smart-hwr_i386_sha256sum.asc

    - smart-languagesetup_i386_sha256sum.asc

    - smart-activation_i386_sha256sum.asc

    - smart-gallerysetup_i386_sha256sum.asc

    - smart-notebook_i386_sha256sum.asc

    - smart-product-drivers_i386_sha256sum.asc

    - nwfermi_i386_sha256sum.asc

- the **swbuild.asc** file containing the public key to use for checking the signature

    - the key ID is 7DD63D3B

    - the fingerprint is 2907 DB4C CC5D 44D8 3694 5F8A BB8D 4237 7DD6 3D3B

- a **validate.sh** script

- a **customization** file

- a **customize.sh** script

**To verify the integrity of the packages, keeping all the files in the current working directory**

1. Import the key from **swbuild.asc** to GPG, and then find the fingerprint.

2. Confirm the fingerprint with SMART or your authorized SMART reseller.

3. Sign the key with your own ultimately trusted key.

4. Execute the command `./validate.sh files`.

   You can check the on-screen output to verify the integrity of the package files.

# Editing the customization file

After you verify the packages, you can edit the customization file. The file contains script variables, which are explained by the comments that precede them. To perform a remote silent mass deployment, you must insert the product key you received from SMART. You can also select the gallery components you want to make available.

After you modify the customization file, execute `customize.sh files`. This modifies all the DEB package files by inserting the contents of the customization file into the maintainer scripts of each package. The ASC integrity files are remade and signed with the key you specified on the SMART_ ARCHIVE_KEY line in the customization file. If you don't define the key in SMART_ARCHIVE_ KEY, the current user's default key is used for signing.

After you customize all the packages, you can install them with `dpkg -i`, or you can insert them into your particular deployment system.

# Installing a Debian archive

The archive named **SMART_software_deb_repo.tar.gz** can be downloaded (see *Downloading SMART Notebook software Debian archive* on the next page) and contains the following:

- the linear directory tree dists -> stable -> non-free -> binary-i386, where the last directory (binary-i386) contains eight Debian package files

  - smart-common_<version>_i386.deb

  - smart-hwr_<version>_i386.deb

  - smart-languagesetup_<version>_i386.deb

  - smart-activation_<version>_i386.deb

  - smart-gallerysetup_<version>_i386.deb

  - smart-notebook_<version>_i386.deb

- smart-product-drivers_<version>_i386.deb

- nwfermi-<version>_i386.deb (for SMART Board 480 interactive whiteboards only)

> **N O T E S**
>
> - The **stable** and **binary-i386** directories contain Debian archive metadata.
>
> - The customization script provided reconstructs part of this metadata after modifying the packages themselves.

- the **swbuild.asc** file containing the public key to use for checking the signature. You should also be able to import this key directly from the key servers. Our key ID is 7DD63D3B and the fingerprint is 2907 DB4C CC5D 44D8 3694 5F8A BB8D 4237 7DD6 3D3B.

- a **validate.sh** script

- a **customization** file

- a **customize.sh** script

## Downloading SMART Notebook software Debian archive

**To download SMART Notebook 11 software**

1. Go to smarttech.com/downloads.

2. In the *SMART Notebook collaborative learning software* area, select **Choose a version**, and then select **11 for Linux** from the drop-down list.

3. Select the **Debian Downloads** section, and then click **Download** next to *Debian prepackaged repository*.

4. Fill out the *Download Software* form, accept the end user license agreement (EULA) and then click **Start Download**.

   The *Download Confirmation* page appears, and then the *File Download* dialog box appears.

5. Click **Save**.

6. Browse to where you want to save the file, and then click **Save**.

   The *Download Complete* dialog box appears.

7. Click **Close**.

8. Browse to where you saved the file, and then double-click **SMART_software_deb_repo.tar.gz**.

   Your file compression program opens.

9. Decompress **SMART_software_deb_repo.tar.gz**, and then browse to the decompressed folder.

## Verifying the integrity of the packages

Verify the integrity of the packages, keeping all these files in the current working directory.

### To verify the integrity of the packages directory

1. Import the key from **swbuild.asc** to GPG, and then find the fingerprint.

2. Confirm the fingerprint with SMART or your authorized SMART reseller.

3. Sign the key with your own ultimately trusted key.

4. Execute the command `./validate.sh archive` to verify the integrity of the package files you have received, and then check the output.

## Following the validation process

If you want to follow the validation process, you can use the command `bash -x validate.sh archive`. The following is an overview of that process.

### To reproduce the validation process

1. cd to **dists/stable**.

2. Check the **Release** file with this command:

```
gpg --verify Release.gpg Release
```

3. Check the Packages metadata files under binary-i386 with these commands:

```
cat Release|tr -s ' ' |cut -d' ' --fields=2,4 |sed 's% % %' > t
sha256sum -c t
```

This checks the **non-free/binary-i386/Packages.*** files.

4. cd to **non-free/binary-i386**.

5. Check the actual package files using the **Packages** file with these commands:

```
grep MD5sum Packages > a
dpkg-scanpackages . ../../smart.override |grep MD5sum > b
diff a b
```

If the files are the same, the last command doesn't produce an output. If a line is different, then that line identifies a corrupt package file. In the **Packages** file, find the line that begins with MD5sum, and two lines above it, the line that begins with Filename identifies the corrupt package file. The script displays the names of packages discovered in this way.

## Editing the customization file

After you verify the packages, you can edit the customization file placed outside **dists**. The file contains script variables, which are explained by the comments that precede them. To perform a remote silent mass deployment, you must insert the product key you received from SMART. You can also select the Gallery components you want to make available. You need your own archive key ID to re-sign the main **Release** file, and you should assign it to SMART_ARCHIVE_KEY.

After you modify the customization file, execute `./customize.sh`. This modifies all the DEB package files by inserting the contents of the customization file into the maintainer scripts of each package. Then it rebuilds the archive metadata, recalculates the integrity checksums and signs the main **Release** file.

Sign the new **Release** file with your own archive key by editing the SMART_ARCHIVE_KEY line in the customization file with your key ID. If you don't define the key in SMART_ARCHIVE_KEY, the current user's default key is used for signing.

## Installing the archive

After you customize the packages and rebuild the Debian archive, you can copy all of the files into the document tree of your web server. Before you can install the packages with `apt-get install`, you must update the package list on each target computer.

**To update the list of customized packages to each computer**

1. Add a one-line file named **smart.list** under **/etc/apt/sources.list.d/** to identify the archive. It should contain the following:

```
deb http://webservername/myfolder/ stable non-free
```

where `myfolder` stands for the directory into which you have copied *dists* in the document tree of the web server.

OR

If you aren't using a web server, use the following line in the file:

```
deb file:/home/swbuild/myfolder/ stable non-free
```

2. Add the necessary public key to the apt system using **apt-key add <publickey>.asc**.

> 📝 **N O T E S**
>
> ○ If you didn't edit the customization file, the public key that verifies the integrity of the packages is the public key in **swbuild.asc**.
>
> ○ If you did edit the customization file, the integrity information is rebuilt with another private key. This key is defined on the `SMART_ARCHIVE_KEY` line in the customization file. You must export the corresponding public key to an ASCII file, and then specify that file.

3. Execute **apt-get update**.

> 📝 **N O T E**
>
> In Synaptic, go to **Settings/Repositories**, select the **Third Party Software** tab, click **Add**, type one of the two lines above that starts with *deb http* or *deb file,* and then click the **Reload** button on the toolbar.
>
> This adds a new source line at the end of **/etc/apt/sources.list**.

4. Install the software by executing **apt-get install smart-product-drivers smart-notebook**.

# Customizing the installation

For the procedures in this section, assume the following:

- Your website is **http://myweb.company.com**.

- The document root of your server is **/var/www**.

- The Gallery data will be in **/var/www/efe**, accessed as **http://myweb.company.com/efe**.

When you customize the installation, you can do the following:

- Provide the SMART Notebook software product key for automatic activation.

- Install SMART Notebook software, SMART Product Drivers or both.

- Disable the automatic start of SMART Board Service and SMART Board Tools.

- Customize the installation of Gallery Essentials, including the following:

  - selecting individual categories

  - installing the Gallery from previously downloaded source

  - installing the Gallery in a customized location

  - registering a previously installed Gallery, such as a Gallery on a shared network

  - registering Team Content

  - selecting a source URL for downloading Gallery content, so that you can set up Gallery content on your own web server as a download source.

## Setting up your server for Gallery downloads

After setup, you can download Gallery content.

**To set up your web server**

1. Make a **/var/www/efe** directory to receive the content.

2. Install the customized packages in the default location with their default Gallery source URL (currently **http://downloads01.smarttech.com/software/efe/**).

3. Download the data to its intended location with these commands:

```
/opt/SMART Technologies/Gallery Setup/bin/GallerySetup \
  --downloadall \
  --destination /var/www/efe \
  --configuration /opt/SMART Technologies/Gallery Setup/share/
  SMARTEssentials.gallerycollection
```

4. Use a browser to check your web server and check that you see a long list of files ending in .gallery or in .zip.

## Customizing the packages

To use your new download source you must customize the packages.

![icon] **To customize the packages**

1. In the customization file, edit the following line, making sure it isn't commented.

   ```
   export CONTENT_LOCAL_URL=http://myweb.company.com/efe
   ```

2. Add your other customizations.

3. Execute **customize.sh**.

4. Install the customized packages.

   Messages like the following one indicate that your web server connection is checked:

   ```
   Checking connection to: http://myweb.company.com/efe
   Check succeeded
   ```

   The selected Gallery components are downloaded and installed.

   > ✏️ **N O T E**
   >
   > To revert to the original download source that came built into the software, comment out the CONTENT_LOCAL_URL line, and then execute **customize.sh** again.

## Dependencies

The two main packages are **smart-notebook** and **smart-product-drivers**; the other five are their dependencies.

| Main package | Dependencies |
| --- | --- |
| smart-notebook_<version>.i686.deb | • smart-common<br>• smart-hwr<br>• smart-languagesetup<br>• smart-activation<br>• smart-gallerysetup |
| smart-product-drivers_<version>.i686.deb | • smart-common<br>• smart-hwr<br>• smart-languagesetup |

# Installation

To install package sets, use a `dpkg --install` command.

**EXAMPLE**

```
dpkg --install smart-common_10.2.286.0-1_i386.deb
```

To install archives accessed over a network, use an `apt-get install` command.

**EXAMPLE**

```
apt-get install smart-common
```

📝 **NOTES**

- After you make changes to the online archive, run **apt-get update** on the target computers.

- Using an `apt-get` command installs dependencies. For example, if you run

  ```
  apt-get install smart-product-drivers smart-notebook
  ```

  you install all seven of the SMART packages.

## Environment variables

You can set the following environment variables to follow the installation progress (they're intended for use with **dpkg**).

| Variable | Use |
| --- | --- |
| Logging | Set DEB_MAINTAINER_LOG to a value, and DEB_MAINTAINER_ LOGPATH to the desired path. The maintainer scripts in the packages then log the installation and removal events to a file named **smart_ debian_install.log** under the specified path. |
| Tracing | Set DEB_DEBUG_SCRIPTS to a value before installing or removing the software. The package scripts execute with `set -x` with the usual screen output. |
| Bypassing package scripts | If the scripts fail so severely that they prevent you from installing or removing the software, you can bypass them while purging. <br> For example, you can remove the package in this way: <br> ``` env DEB_MAINTAINER_NOSCRIPTS=1 dpkg --purge smartcommon ``` |

## Removing the software

When you installed the software, it was copied into the **/opt/SMART Technologies** folder. Other files are placed in **/etc/xdg, /etc/udev/rules.d**, **/usr/share** (such as **doc**, **man/man1**, **icons/ hicolor**, **pixmaps/hicolor**, **applnk**, **mimelnk/application**), **/usr/local/share/applications** and **$HOME/.config**.

Most files are removed during the package removal process, but the following files remain:

- global configuration files

    - /etc/xdg/SMART Technologies/*.conf

    - /etc/xdg/SMART Technologies.conf

- activation data

    - /opt/SMART Technologies/common/data/.mp.1.1.dat

    - /opt/SMART Technologies/common/data/.vp.1.1.dat

- MIME XML files wherever they're installed

- files created in the current user's home under **$HOME/.config**

# Chapter 3
# Installing SMART Notebook 11 software from RPM packages

## Introduction

This section shows how you can install SMART Notebook software and SMART Product Drivers software shipped as RPM package files.

> 👉 **IMPORTANT**
>
> Before you install SMART Notebook 11 software, you must obtain a product key. See *Before installing SMART Notebook software* on page 3.

# Downloading SMART Notebook software RPM packages

**To download SMART Notebook 11 software**

1. Go to www.smarttech.com, select **Support**, and then select **Downloads**.

2. In the *SMART Notebook collaborative learning software* area, select **Choose a version**, and then select **11 for Linux** from the drop-down list.

3. Select the **Download RPM packages** section, and then click **Download** next to *RPM plain files*.

4. Fill out the *Download Software* form, accept the EULA and then click **Start Download**.

   The *Download Confirmation* page appears, and then the *File Download* dialog box appears.

5. Click **Save**.

6. Browse to where you want to save the file, and then click **Save**.

   The *Download Complete* dialog box appears.

7. Click **Close**.

8. Browse to where you saved the file, and then double-click

   **SMART_software_rpm_files.tar.gz**.

   Your file compression program opens.

9. Decompress **SMART_software_rpm_files.tar.gz**, and then browse to the decompressed folder.

# Before you start

On some platforms you must install perl-Mail-Sendmail, providing perl(Mail::Sendmail).

You also need

- rpmbuild (in package rpm or rpm-build)
- rpm2cpio (probably in rpm)
- cpio
- expect

If you want to put the packages in a web repository, you also need the Python script **createrepo**.

You need to have GPG configured, with a secret key generated and set as the default-key in **~/.gnupg/gpg.conf**.

You must import the public key to RPM to check the released packages.

- To import the key, execute as root:

```
rpm --import swbuild.asc
```

- To check a package file, execute:

```
rpm --checksig smart-whatever.i686.rpm
```

> **NOTE**
>
> The output line must end in `gpg OK`.

You must import the public key to GPG and sign it to use a zypper repository.

- To import it, use:

```
gpg --import swbuild.asc
```

- To sign it, use:

```
gpg --sign-key 7DD63D3B
```

> **NOTE**
>
> The SMART key will be signed with your default key.

You must place the public key in **/etc/pki/rpm-gpg** to use a yum repository.

You need a file named **~/.rpmmacros,** which must include the following:

```
%_topdir <some topdir>
%_signature gpg
%_gpg_path /home/<accountname>/.gnupg
%_gpg_name <key id>
```

Where:

- <accountname> is the user name you use to run the customization. You don't need to be root to customize.

- <some topdir> is a directory to where your account can write, or to a directory that you'll make. Use an absolute path. The `customize` script will parse the **.rpmmacros** file and extract that path. That is where the new package files will be built before being moved back to the cwd.

- <key id> is the description string of the key you will use to sign the customized packages, as displayed by `gpg --list-secret-keys`.

---

**E X A M P L E**

```
%_topdir /home/swbuild/rpmbuild
%_signature gpg
%_gpg_path /home/swbuild/.gnupg
%_gpg_name SMART Technologies (Build and Install Group)
```

---

# About installing the RPM package

The archive contains RPM installation packages for SMART Notebook software and SMART Product Drivers, as well as a customization template file.

To prepare the packages for use, you must perform the following steps, which are explained in detail below:

- Provide the prerequisites detailed above.

- Verify the integrity of the files you received.

- Obtain a product key, either online from SMART or from your authorized SMART reseller. See *Before installing SMART Notebook software* on page 3.

- Select a master signing key to use for signing the relevant data with GPG (the above steps can happen in any order).

- Edit the customization template file to insert your product key, to select the content galleries you want, etc.

- Execute the automated procedure that merges the customization information into the packages.

- Optionally, place the customized packages in a web repository and configure client machines.

# About customization

> **☞ I M P O R T A N T**
>
> In order to activate your software for a silent unattended network deployment and to customize the installation, you must edit the customization file named **customization** before you install the software.

The installation data is shipped as a package set, which includes instructions for making a web repository accessible with yum or zypper and for configuring client computers.

## Before you start

Ensure that the packages rpm, cpio, expect and createrepo are installed. You also need GPG and utilities like sed, cut and tr.

# Installing an RPM package set

SMART provides an archive named **SMART_software_rpm_files.tar.gz** containing the following:

- ten RPM package files

  - smart-common_<version>.i686.rpm

  - smart-hwr_<version>.i686.rpm

  - smart-languagesetup_<version>.i686.rpm

  - smart-activation_<version>.i686.rpm

  - smart-gallerysetup_<version>.i686.rpm

  - smart-notebook_<version>.i686.rpm

  - smart-product-drivers_<version>.i686.rpm

  - nwfermi-<version>-i386.rpm (for SMART Board 480 interactive whiteboards, only)

  - nwfermi_kernel_default_devel-<version>.i686.rpm (for Linkat)

  - dkms-<version>.noarch.rpm (dependency for nwfermi)

- the **swbuild.asc** file containing the public key as the archive signing key

  - the key ID is 7DD63D3B

  - the fingerprint is 2907 DB4C CC5D 44D8 3694 5F8A BB8D 4237 7DD6 3D3B

- a customization file named **customization**

- a script named **customize.sh**

## Verifying the integrity of the packages

Verify the integrity of the packages, keeping all of the files in the current working directory.

**To verify the integrity of the packages**

1. Import the key from **swbuild.asc** to GPG, and then find the fingerprint.

2. Confirm the fingerprint with SMART or your authorized SMART reseller.

3. Sign the key with your own ultimately trusted key.

4. Import the key to rpm with **rpm --import swbuild.asc**.

5. Execute the command **rpm --checksig <package>** for every package file, where <package> denotes the package file.

   The output for each command should end in `gpg OK`.

After you verify the packages, you can edit the customization file. The file contains script variables, which are explained by the comments that precede them. To perform a remote silent mass deployment, you must insert the product key you received from SMART. You can also select which gallery components you want to make available

After you modify the customization file, execute `./customize.sh`, which modifies all the RPM package files by inserting the contents of the customization file into the maintainer scripts of each package. At this point, the RPM package files are remade and signed with the key specified in **.rpmmacros**.

After you customize all the packages, you can install them with `rpm -i`, or you can insert them into your particular deployment framework.

# Customizing the RPM installation

For the procedures in this section, assume the following:

- Your website is **http://myweb.company.com**

- The document root of your server is **/var/www**

- The Gallery data will be in **/var/www/efe**, accessed as **http://myweb.company.com/efe**

When you customize the installation, you can do the following:

- Provide the SMART Notebook software product key for automatic activation.

- Install SMART Notebook software, SMART Product Drivers or both.

- Disable the automatic start of SMART Board Service and SMART Board Tools.

- Customize the installation of Gallery Essentials, including the following:

  - selecting individual categories

  - installing the Gallery from previously downloaded source

  - installing the Gallery in a customized location

  - registering a previously installed Gallery, such as a Gallery on a shared network

  - registering Team Content

  - selecting a source URL for downloading Gallery content, so that you can set up Gallery content on your own web server as a download source

## Setting up your server for Gallery downloads

After setup, you can download Gallery content.

**To set up your web server**

1. Make a **/var/www/efe** directory to receive the content.

2. Install the customized packages in the default location with their default Gallery source URL (currently **http://downloads01.smarttech.com/software/efe/**).

3. Download the data to its intended location with these commands:

```
/opt/SMART Technologies/Gallery Setup/bin/GallerySetup \
  --downloadall \
  --destination /var/www/efe \
  --configuration /opt/SMART Technologies/Gallery Setup/share/
  SMARTEssentials.gallerycollection
```

4. Use a browser to check your web server and check that you see a long list of files ending in .gallery or in .zip.

## Customizing the packages

To use your new download source you must customize the packages.

▢ **To customize the packages**

1. In the customization file, edit the following line, making sure it isn't commented.

```
export CONTENT_LOCAL_URL=http://myweb.company.com/efe
```

2. Add your other customizations.

3. Execute **customize.sh**.

4. Install the customized packages.

   Messages like the following one indicate that your web server connection is checked:

```
Checking connection to: http://myweb.company.com/efe
Check succeeded
```

   The selected Gallery components are downloaded and installed.

   📝 **N O T E**

   To revert to the original download source that came built into the software, comment out the
   CONTENT_LOCAL_URL line, and then execute **customize.sh** again.

## Dependencies

The two main packages are **smart-notebook** and **smart-product-drivers**; the other five are their
dependencies.

| Main package | Dependencies |
|---|---|
| smart-notebook_<version>.i686.rpm | • smart-common<br>• smart-hwr<br>• smart-languagesetup<br>• smart-activation<br>• smart-gallerysetup |
| smart-product-drivers_<version>.i686.rpm | • smart-common<br>• smart-hwr<br>• smart-languagesetup |

# Installation

To install package sets, use `rpm -i`.

---

**EXAMPLE**

```
rpm -i smart-common_10.2.286.0-1.i686.rpm
```

---

To install archives accessed over a network, use **yum** install, or **zypper** install.

---

**EXAMPLE**

```
yum install smart-common
zypper install smart-common
```

---

📝 **NOTES**

- After you make changes to the online archive, run yum update, or zypper refresh on the target computers.

- Running yum and zypper installs dependencies. For example, if you run

  ```
  yum install smart-product-drivers smart-notebook
  ```

  you install all seven of the SMART packages.

## Environment variables

You can set the following environment variables to follow the installation progress (they're intended for use with RPM).

| Variable | Use |
| --- | --- |
| Logging | Set RPM_MAINTAINER_LOG to a value, and RPM_MAINTAINER_ LOGPATH to the desired path. The maintainer scripts in the packages then log the installation and removal events to a file named **smart_rpm_ install.log** in the specified path. |
| Tracing | Set RPM_DEBUG_SCRIPTS to a value before installing or removing the software. The package scripts execute with `set -x` with the usual screen output. |
| Bypassing package scripts | If the scripts fail so severely that they prevent you from installing or removing the software, you can bypass them while purging. <br> For example, you can remove the package in this way: <br><br> ```env RPM_MAINTAINER_NOSCRIPTS=1 rpm --nodeps -e smart-common``` <br><br> **✎ NOTE** <br><br> You shouldn't need to do this, because you can use the **RPM -- noscripts** flag. |

## Web repositories

Rather than installing packages locally with RPM commands, you can keep the same packages on a web server and install them using a client tool such as yum or up2date (Red Hat), or zypper or YaST (openSUSE).

**Making a repository**

⬛ **To make a repository**

1. Create a directory to contain your repository and place all the package files in it.

2. cd to the directory, and then execute **createrepo**.

   A subdirectory named **repodata** appears.

3. cd to repodata and check that the files **filelists.xml.gz**, **other.xml.gz**, **primary.xml.gz** and **repomd.xml** were created.

4. Sign **repomd.xml** with **gpg --clearsign repomd.xml**.

   This creates the file **repomd.xml.asc**.

5. Export the public key paired with the secret you have used for signing (that would be your default key ID specified in **.gnupg/gpg.conf)** using

   ```
   gpg --armor --export <key-ID> > repomd.xml.key.
   ```

   You can copy your repository directory into the document tree of any web server.

## Installing packages from the repositories

For the following procedures, assume that the directory is **myrepo**, and that you placed it under **rpm** in a web server named **mywww**. The URL of the repository is now **http://mywww/rpm/ myrepo**.

⬛ **To use yum to install packages from the new repository**

1. Create a file named "mysource.repo" looking like this:

   ```
   [SMART Technologies ULC]
     name=SMART Technologies Software
     baseurl=http://mywww/rpm/myrepo
     gpgcheck=1
     enabled=1
     gpgkey=file:///etc/pki/rpm-gpg/archivekey.asc
   ```

2. Put the file in the **/etc/yum.repos.d** folder.

3. Export the public key used for signing into the file **/etc/pki/rpm-gpg/archivekey.asc**.

4. Execute `yum update`.

   You should see your new repository being accessed, along with previous ones, while yum updates the cached lists it keeps on the client computer.

5. Execute this command:

```
yum install smart-notebook smart-product-drivers
```

   A message appears stating that seven packages will be installed.

**To use zypper to install packages from the new repository**

1. Create a file named mysource.repo like this:

```
[SMART Technologies U.L.C.]
  name=SMART Technologies Software
  enabled=1
  autorefresh=1
  baseurl=http://mywww/rpm/myrepo
  path=/
  type=rpm-md
  keeppackages=0
```

2. Put it in the **/etc/zypp/repos.d** folder.

3. Retrieve the public key used for signing using
   *wget http://mywww/rpm/myrepo/ repodata/repomd.xml.key*.

4. Import it using `gpg --import`, find out its ID using `gpg --list-keys`, and then sign it with your default key using `gpg --sign-key <key-ID>`.

5. Execute `zypper refresh`.

   You should see your new repository being accessed, along with previous ones, while zypper updates the cached lists it keeps on the client computer.

6.  Execute this command:

```
zypper install smart-notebook smart-product-drivers
```

A message appears stating that seven packages will be installed.

> 📝 **N O T E**
>
> A prepackaged repository can also be downloaded by following the instructions in *Downloading SMART Notebook software Debian archive* on page 10. If you use and customize the packages, then you must execute **createrepo** as a last step .

## Removing the software

When you installed the software, it was copied into the **/opt/SMART Technologies** folder. Other files are placed in **/etc/xdg, /etc/udev/rules.d**, **/usr/share** (such as **doc**, **man/man1**, **icons/ hicolor**, **pixmaps/hicolor**, **applnk**, **mimelnk/application**), **/usr/local/share/applications** and **$HOME/.config**.

Most files are removed during the package removal process, but the following files remain:

- global configuration files
    - /etc/xdg/SMART Technologies/*.conf
    - /etc/xdg/SMART Technologies.conf
- activation data
    - /opt/SMART Technologies/common/data/.mp.1.1.dat
    - /opt/SMART Technologies/common/data/.vp.1.1.dat
- MIME XML files wherever they're installed
- files created in the current user's home under **$HOME/.config**