

Software Management

In Slint software is provided as packages that can be installed, removed and updated.

The packages shipped in the installation ISO come from these repositories:

For version **14.2**:

- [Slackware 64-bit](#) or [Slackware 32-bit](#)
- [Salix 64-bit](#) or [Salix 32-bit](#)
- [Extra-14.2 64-bit](#) or [Extra-14.2 32-bit](#): packages built by Salix from building stuff provided by <https://slackbuilds.org>
- [Slint 64-bit](#) or [Slint 32-bit](#)

From version **14.2.1** onward (64-bit only):

- [Slackware](#)
- [Slint](#)

These sources of packages are registered in the file `/etc/slapt-get/slapt-getrc`.

Do yourself a favor: read the comments in `/etc/slapt-get/slapt-getrc` and `/usr/doc/slapt-get-0.10.2t/README.slaptgetrc.Slint` in the installed system.

After having installed Slint, you be automatically notified of updates of installed packages coming from repositories registered in `/etc/slapt-get/slapt-getrc`

You can install additional packages coming from the `Salix` and `Extra-14.2` repositories using the commands `slapt-get`, or the graphical application `gs_lapt`.

You want an application not available in one of these repositories? If the components needed to build it are available in one of the "source" repositories [slkbuild](#) or [sbo](#) you can use one of the commands `slapt-src` or `spi`, or the graphical application `sourcery` to build and install a package for it.

However, this is more advised for experienced users, as in this case you can occasionally encounter issues like missing dependencies or conflicts with installed packages: the components used to build the packages are provided by volunteers who can not check the results in every possible configuration.

If you have enough experience you can also build packages yourself. If you are interested see how to do that in the [Salix](#) and [Slackware](#) documentations on the web: these distributions use two different methods with identical results: packages in the Slackware format, that can be installed in both distributions as in Slint, and Slint provides all needed tools, like `makepkg`, `fakeroot` and `slkbuild`.

- Always use the provided tools to perform software packages management.
- Do not compile and install *manually* a software, without (possibly making then) installing a package, else your system would soon be messed up.

Some package management tools are accessed in Graphical mode (from a menu or clicking on an icon) others typing a command in terminal. All the commands should be typed as `root`, i.e. with

administrative privileges. To use them:

- In a Terminal type su (for: *Super User*) and when asked, type root's password.
- In Graphical mode, just click on the menu entry or in the icon in the Slint Control Center and when asked, type the root's password.

To know the usage of the command accessed from a terminal, type "<command name> -help". example for the spi command:

```
didier[~]$ spi --help
```

You can get more in-depth information typing e.g. man spi instead of spi -help (no need to be root for that).

Here are the tools intended for end users. Those in bold are the ones you will use most of the time for packages installation and updates

Command	Access	Purpose	Comments
installpkg	Terminal	Install a package already downloaded	You can use spkg instead
removepkg	Terminal	Remove an installed package	You can use spkg instead
upgradepkg	Terminal	Replace an installed package	You can use spkg instead
spkg	Terminal	Install, remove and upgrade packages	Does not download packages
slapt-get	Terminal	Search, download and install packages	Manage dependencies
slapt-src	Terminal	Search, downloads the components, builds and install packages	Used when a already built package is not available
spi	Terminal	Do most tasks of slapt-get for already built packages, else of slapt-src	
gslapt	Graphical	Graphical front-end for slapt-get	Accessible from the Slint Control Center
sourcery	Graphical	graphical front-end for slapt-src	Accessible from the Slint Control Center

Keep your system up to date

You need to keep your system safe installing the software updates as soon as available:


- After Slint installation run once this command:

```
slapt-get --add-keys
```

- in a terminal run periodically as root these commands:

```
slapt-get -u
slapt-get --upgrade
```

- Alternatively from the Slint Control Center (Applications) or the menu click on Gslapt, give the password for root then click Update, then Mark All Upgrade then Execute if updates are available.

A small icon like this  is displayed in the notification area of the panel (top panel, in MATE) to notify of available software updates. Just make a left click on it and follow instructions.

Be aware that some packages are blacklisted in `/etc/slapt-get/slapt-getrc`, i.e. can not be automatically upgraded or installed.

This is the case for kernel upgrade, provided only as security fixes. See below how to apply them.

How-to apply kernel upgrades for security fixes

The Linux kernel packages are sometimes upgraded for Slackware version 14.2 to provide security fixes.

This is noted in this [Slackware ChangeLog](#) and we will also mention it in the Slint ChangeLogs like [this one](#).

Slint version 14.2 onward should be upgraded accordingly, but kernel packages are blacklisted in `/etc/slapt-get/slapt-getrc`. For this reason you will know that an upgrade is available as `gslapt` or `slapt-get -upgrade` will mention these packages as EXCLUDED.

When this occurs follow the instructions below upgrading. TALL commands are to be typed in a terminal as root:

- The last upgrade command will have listed the blacklisted kernel-* packages. If you didn't remove any package, the list will be like this for Slint64-14.2 or Slint64-14.2.1 (64-bit):

```
kernel-firmware kernel-generic kernel-modules kernel-huge kernel-headers kernel-source
```

And like that for Slint-14.2 (32-bit):

```
kernel-firmware kernel-generic-smp kernel-generic kernel-huge kernel-headers kernel-modules kernel-huge-smp kernel-modules-smp kernel-source
```

- Upgrade all these packages with this command:

```
slapt-get -i <list>
```

where `<list>` is the list of packages to be upgraded like: `kernel-firmware kernel-generic kernel-modules ...`

- If you are using a generic kernel with an `initrd` you should now rebuild the `initrd`. If it is not named `/boot/initrd.gz`, `/boot/initrd.gz` should be a symlink to it.
- Check that `/boot/vmlinuz` be a symbolic link to the kernel you want to use.
- Then, you **must** upgrade your bootloader, else Slint won't boot anymore! So do this:
- *lilo* users, be sure that the correct kernel and `initrd` be referred to in `lilo.conf`, then check the output of this command:

```
lilo -t -v # then if all goes well:  
lilo
```

- *elilo* users, to copy the kernel and case occurring the `initrd` to the EFI System Partition type:

eliloconfig

Search, install and remove packages

Both `spi` and `slapt-get` have a `--search` option or `target`, see their man pages or the `--help` option. `gslapt` has a search field. All these tools can then install the chosen packages, and also remove already installed packages.

Avoid to remove packages shipped in the Slint ISO, even if you do not use them at all. Removing an included package will result in no increase in performance and could prevent other applications to run. Also, if you have added a package not included in Slint you may remove it but be careful that the removed package be not a dependency of other ones that you also installed and intend to continue using.

From:
[/wiki/](#) - **Slint**

Permanent link:
[/wiki/doku.php?id=en:software_management](#)



Last update: **2019/11/18 12:54**