Installing Software

Overview

There are three basic ways to install software in Slackware: install a pre-build binary package, make your own package, or compile the sources and install the resulting binaries manually. If you need to compile a program from source code, making it into your own package is easy and useful. Here's how these methods work.

Installing Pre-built Binary Packages

To install a package you first need to get the package.

- Official packages can be found on the Slackware CD-ROM or the Slackware FTP/HTTP mirrors. The package browser can be found at http://packages.slackware.com/.
- Unofficial, community contributed binaries can be searched for at http://slackfind.net/ which indexes the most well-known community repositories and Slackware-compatible spin-offs like http://slacky.eu/ http://salixos.org/ and more.
  Usually a package file name ends with `.tgz` or `.txz` and looks something like "`mozilla-firefox-14.0-x86_64-1.txz`". The more universal naming scheme is "`PRGNAM-VERSION-ARCH-BUILD.txz`" where ARCH denotes the architecture for which the package was built (examples are `x86_64`, `i486`. `armv6l`, `armv7hl`, `noarch`) and the BUILD number is not just used to mark incremental package updates but also allows community packagers to add their own *build tag*[1] which makes them unique.
  Make sure you get the Slackware package built for the correct architecture and made for your version of Slackware!

To install the package you need to change to the root user. Run either the command `su - ` or `sudo -i` and enter your root password. Now run the command

```
upgradepkg --install-new your-long-package-file-name
```

You should see a bunch of text running across your screen telling you about the package being installed, and possibly about an old package it might be replacing.

Make a Package

From Source

The recommended way to build packages from source is to use or create SlackBuild scripts.

However, there are other utilities such as slacktrack (part of Slackware itself), src2pkg and checkinstall which automate the package creation process entirely (with a few user inputs), but these are not guaranteed to work with all possible kinds of source projects.

From third-party package formats

Binary packages from other distributions can be converted to Slackware packages by using conversion tools. Although building packages from source is always better, there are some situations which might warrant the use of these tools.

- `rpm2tgz` which is part of Slackware can be used to convert .rpm packages to Slackware packages.
- deb2tgz converts .deb packages to Slackware packages.
- alien is a tool used to convert packages to and from different distribution package formats.

Compiling from scratch

Some users still prefer the traditional method of compiling from source directly and installing using `make install` or a similar command.

The generic steps for standard GNU Autotools Makefiles are:

```
./configure
make
make install # as root
```

The configure script may take additional switches to enable or disable features or configure other build options.

Other build systems may involve different steps, depending on how the source project is set up. As always, it is best to read the upstream developer documentation to figure out how to compile and build a program for your system, including which dependencies to fulfill.

Community Repositories

- Slackware Team:
    - Eric Hameleers regular packages; packages with patent restrictions; KDE bleeding edge; multilib
    - Robby Workman
- Wider Community:
    - Dropline Gnome
    - Micro Linux Enterprise Desktop (MLED)
    - MATE SlackBuilds (MSB)
    - Ponce's Pimp My Slack
    - http://slacky.eu/
    - http://salixos.org/

Sources

- Originally written by peterwwillis for the SlackDocs Wiki Project
- Additional contributions by V.Harishankar
- Additional contributions by Eric Hameleers, with input from *maciuszek*

howtos, software, installing, package management, author peterwwillis
[1)]
-Well-known tags are "_SBo" (slackbuilds.org), "alien" (Eric Hameleers), "_rlw" (Robby Workman), "ponce" (Matteo Bernardini)-