Package Management & Software Installation

An entire book can be written on Slackware's package management systems, and indeed has been. This section will briefly cover the use of Slackware's official package management utilities **installpkg**, **upgradepkg**, and **removepkg**. Also **slackpkg** a utility program which has been included in Slackware since Slackware-12.2 which is used to manage the packages which are distributed with the operating system itself. This section also covers **sbotools**, a 3rd party utility suite that resolves dependencies and accesses the repositories of slackbuilds.org, Slackware's largest repository for non-official packages.

What is a Slackware Package?

A Slackware package is simply a **xz** or **gzip** compressed **tar** archive comprised of source code, scripts which will install and configure said code to the Slackware system in a manner befitting the established principles and filesystem hierarchy used by the Slackware community, and documentation.

The naming convention for a Slackware package is *package-version-architecture-build*. The filename extension will either be .tgz or .txz depending on how the package's maintainer has compressed the **tar** archive, with **gzip** or **xz** respectively.

Package-version-architecture-build.tgz
or
package-version-architecture-build.txz

pkgtools

Slackware’s official package management system **pkgtools**, is comprised of six utilities located in the */sbin* directory. Three of which, this section will cover **installpkg**, **upgradepkg**, and **removepkg**. Their functions are implied in their names. **pkgtools** utilities do not have access to an online repository of packages, they merely perform their functions on packages already acquired by the user.

```
# installpkg  mozilla-firefox-45.2.0esr-i586-1.txz
# upgradepkg mozilla-firefox-57.0-i686-1.txz
# removepkg mozilla-firefox-57.0-i686-1.txz
```

It is also important to note, that this utility suite does not check for dependencies. There may be other software packages which a package may depend upon, if so, it is considered the administrator's responsibility to be informed of such dependencies and install the dependencies as needed. This issue with dependencies is addressed with some of the community built package managers which will be covered later in this section.

Further information about these utilities and their uncovered counterparts can be obtained by reading their corresponding **man** pages, documentation located in the /var/log/packages/pkgtools* file, as well as online documentation.

```
$ man pkgtool
$ less /var/log/packages/pkgtools*
http://www.slackware.com/config/packages.php
https://www.slackbook.org/html/package-management-package-utilities.html
```

slackpkg

slackpkg is the utility responcible for managing all the packages included with the Slackware Linux distrobution. It is used to install new packages which are introduced to the distrobution, as well as upgrade existing packages to their latest versions as per the distrobution.

Before the user is able to use this utility, **slackpkg** must first updated with the current list of packages from a repository online. To do this, the user must first select a repository to update from. By editing the text file */etc/slackpkg/**mirrors**,* instruction are located in the comments at the top of the document*.*

```
# nano /etc/slackpkg/mirrors
```

Once this step is complete, **slackpkg** is ready to update.

```
# slackpkg update
```

Installing new packages found in the repository, and upgrading existing packages when available should be regularly preformed, as it is the primary method of maintaining security, and software updates.

```
# slackpkg install-new
# slackpkg upgrade-all
```

Further information about **slackpkg** can be obtained in the corrosponding **man** pages, documentation located in the */usr/doc/**slackpkg**** directory, in the */var/log/packages/**slackpkg**** file, as well as online.

```
$ man slackpkg
$ ls /usr/doc/slackpkg*
$ less /var/log/packages/slackpkg*
https://www.slackpkg.org/documentation.html
https://docs.slackware.com/slackware:slackpkg
```

sbotools

There are many online repositories hosting non-official Slackware packages. The largest of which is the community based slackbuilds.org. **sbotools**, a utility suite, can be downloaded from the developer's github located here https://pink-mist.github.io/sbotools/. Download the latest **sbotools** package by following the directions on said page and install using **installpkg**.

```
# installpkg sbotools*.tgz
```

After installing, similarly to **slackpkg**, **sbotools** must be updated with the slackbuilds repository, this is done using the **sbosnap** utility.

```
# sbosnap fetch
```

After having downloaded the repository, packages can be downloaded and installed using the **sboinstall** utility. For example, to download and install **unrar** a decompression utility used to decompress Roshal Archives.

```
# sboinstall unrar
```

When using **sbotools**, only the package-name is nessisary to refer to the package, the rest of the naming convention can be omited. One of the more appreciated features of **sbotools** is it's ability to handle dependancies. The user will be notified during the installation process if/when a dependancy is found, and asked to choose wether or not the dependency should be downloaded and installed before continuing installation of the dependant package.

Periodically it is nessisary to update the **sbotools** with the slackbuilds repository, as packages are updated and new packages are added regularly. To keep **sbotools** updated, simply run the **sbocheck** utility. This will also inform the user of packages installed which can be updated, and prompt the download & installation process of the same.

```
# sbocheck
```

Further information about **sbotools** can be obtained in the corrosponding **man** pages, documentation located in the */var/log/packages/**slackpkg**** file, as well as online.

```
$ man sboinstall
$ less /var/log/packages/sbotools*
https://pink-mist.github.io/sbotools/documentation/
```