Building and Installing Packages with sbopkg

Sbopkg is a command-line and dialog-based tool to synchronize with the SlackBuilds.org (“SBo”) repository, a collection of third-party SlackBuild scripts to build Slackware packages. The program has a curses based interface which lets you pick and assemble the programs which you want to compile from source into packages. It can also be used non-interactively in case you know beforehand what your goal is - in that case it is “fire and forget”.

The following information will assist you in getting sbopkg installed, and running on Slackware.

Note that `sbopkg` builds *Slackware* packages. As with any Slackware package management program, you will have to be root to use the program! Execute

```
su -
```

to get a root command prompt with the correct environment configured.

Download sbopkg:

Download sbopkg from: https://www.sbopkg.org/downloads.php
You can get the sources and build a package yourself, or just grab th ready-made Slackware package. The following section of this article assumes that you downloaded the ready-built package.

Install sbopkg

1. Assuming that the `sbopkg` package file name you downloaded to the current directory is *sbopkg-0.38.0-noarch-1_wsr.tgz*, the `installpkg` command to install sbopkg (or upgrade it if it was already present on your system) is as follows:

```
$ su -
# cd /path/to/downloaded/file/
# upgradepkg --install-new sbopkg-0.38.0-noarch-1_wsr.tgz
```

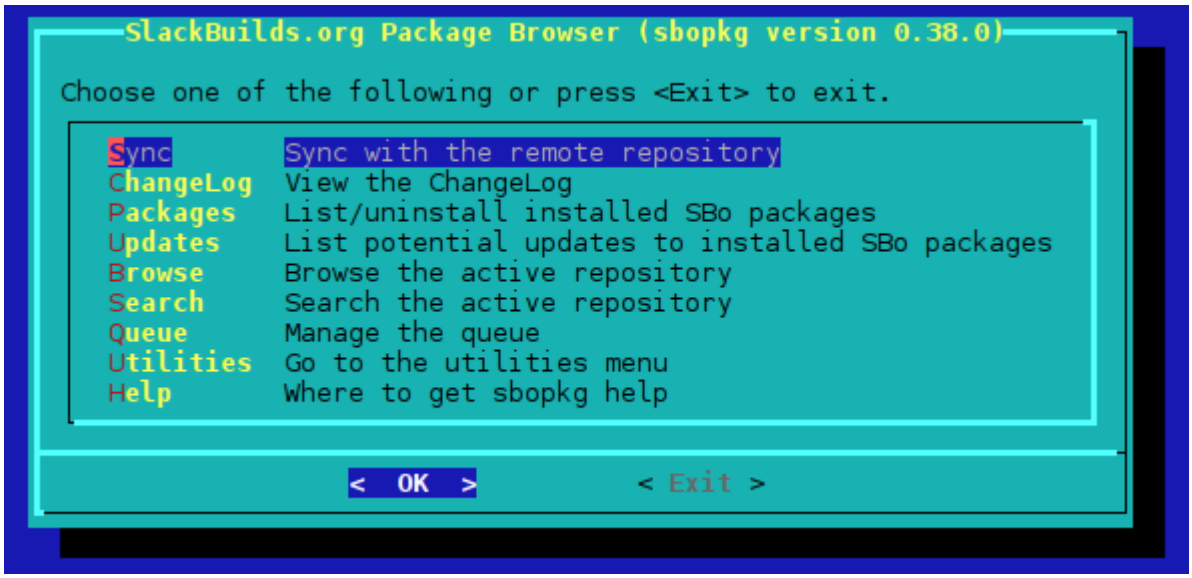
Configure sbopkg

1. Read the `sbopkg` documentation! See https://www.sbopkg.org/docs.php , and also do not forget there is a man page for `sbopkg`.
2. The first time `sbopkg` is executed, you will be asked if it is OK that the program creates the necessary configuration files:

```
# sbopkg

The following directories do not exist:
```

```
Variable                    Assignment
--------                    ----------
REPO_{ROOT,NAME,BRANCH} -> /var/lib/sbopkg/,SBo/,14.2
LOGFILE directory -------> /var/log/sbopkg
QUEUEDIR ----------------> /var/lib/sbopkg/queues
SRCDIR ------------------> /var/cache/sbopkg
TMP ---------------------> /tmp/SBo

You can have sbopkg create them or, if these values are incorrect, you
can
abort to edit your config files or pass different flags.

(C)reate or (A)bort?:
```

. Select C to create these required directories.
The program will then present its main screen:



1. Your first task is to synchronize with the SlackBuilds.org repository - i.e. you let `sbopkg` create a local copy of all the SlackBuild entries on the remote server, for the Slackware release which you are running. You either use the first menu item in the curses-based main screen, or else you can do this from the commandline:

```
# sbopkg -r
Syncing with the remote repository into /var/lib/sbopkg/SBo/14.2.
receiving incremental file list

<long list of filenames...>

sent 451288 bytes  received 36904793 bytes  371702.30 bytes/sec
total size is 35271012  speedup is 0.94

Rsync with the SBo repository for Slackware 14.2 complete.
```

```
Importing SBo repository for Slackware 14.2 GPG Key...
Import done.

***SYNC COMPLETE***
```

2. You are now ready to select the packages you want `sbopkg` to build from source.

Using Queuefiles with sbopkg

The `sbopkg` program itself allows a great deal of automation: the interactive curses interface helps you manage the burden of downloading source code archives, and allowing you to select multiple programs and then compile and package all of those in one go.
But you can automate this process even further, by using sqg - sbopkg queueu generator (included in main package since 0.38.0).

sqg can generate queuefile for each or all packages in SBo repository:

```
sqg -p <filezilla> # this will only generate queue file for filezilla
package only
sqg -a # this will generate queue files for all packages in SBo repository
```

Note that everytime a public update is announced or new repository is available, it's recommended to run sqg -a to generate an updated queue files as deps may be added or removed.

Example of Using sbopkg

As an example, let's install Gramps. The queuefile for Gramps lists the following dependencies, in order of install:

```
orbit2
pyorbit
libbonobo
gnome-mime-data
gnome-vfs
libgnome
gnome-python
gramps
```

Load the "`gramps.sqf`" queue file in sbopkg's curses interface, and Gramps will be successfully built on your Slackware 14.2 system and installed, along with all its dependencies.

Alternatively you can use the less interactive command-line interface:

```
sbopkg -i gramps
Both a queuefile and a package were found with the name "gramps".

Use (Q)ueuefile, (P)ackage, or (A)bort?: q
```

```
##########################################
       New queue process started on:
       Fri Aug 19 04:57:12 WIB 2016
##########################################

++++++++++++++++++++++++++++++++++++++++++
PRE-CHECK LOG
Using the SBo repository for Slackware 14.2
Queue Process:  Download, build, and install

ORBit2:
  Checking GPG for ORBit2.tar.gz ... OK
  Processing ORBit2 2.14.19-3
  Using original .info file
  Using original SlackBuild file
  No build options selected.

pyorbit:
  Checking GPG for pyorbit.tar.gz ... OK
  Processing pyorbit 2.24.0-1
  Using original .info file
  Using original SlackBuild file
  No build options selected.

libbonobo:
  Checking GPG for libbonobo.tar.gz ... OK
  Processing libbonobo 2.32.1-3
  Using original .info file
  Using original SlackBuild file
  No build options selected.

gnome-mime-data:
  Checking GPG for gnome-mime-data.tar.gz ... OK
  Processing gnome-mime-data 2.18.0-2
  Using original .info file
  Using original SlackBuild file
  No build options selected.

gnome-vfs:
  Checking GPG for gnome-vfs.tar.gz ... OK
  Processing gnome-vfs 2.24.4-3
  Using original .info file
  Using original SlackBuild file
  No build options selected.

libgnome:
  Checking GPG for libgnome.tar.gz ... OK
  Processing libgnome 2.32.1-2
  Using original .info file
```

```
  Using original SlackBuild file
  No build options selected.

gnome-python:
  Checking GPG for gnome-python.tar.gz ... OK
  Processing gnome-python 2.28.1-1
  Using original .info file
  Using original SlackBuild file
  No build options selected.

gramps:
  Checking GPG for gramps.tar.gz ... OK
  Processing gramps 3.4.3-1
  Using original .info file
  Using original SlackBuild file
  No build options selected.

++++++++++++++++++++++++++++++++++++++++++++

Pre-check complete.

Do you wish to proceed based on the search results above? Packages not
found will be skipped during the process.

(P)roceed or (Q)uit?: P
```

etcetera.

References for sbopkg

- https://www.sbopkg.org
- https://www.sbopkg.org/docs.php
- https://www.sbopkg.org/downloads.php
- https://www.sbopkg.org/queues.php

Sources

- Originally written by ldkraemer
- Contributions by Eric Hameleers
- Contributions by Willy Sudiarto Raharjo

howtos, software, sbo, package management, author ldkraemer