

sbopkg

sbopkg is a third-party command-line and dialog-based tool to synchronize with the SlackBuilds.org (“SBo”) repository, a collection of third-party [SlackBuild scripts](#) to build and install Slackware packages.

The queue file use tool included in sbopkg can be utilized to handle dependency resolution for known applications and even build queues which can be used on other machines.



**This page is a WIP (work in progress),
Please do not edit or remove**

Installation

Obtaining sbopkg

The sbopkg application is found on the project's website at sbopkg.org. You can choose to download the pre-compiled package or the source code directly from the homepage, for this tutorial we will be using the pre-compile package.

Once you locate the download button use your favorite download manager to download the resulting package into your chosen directory.

Installing sbopkg

Once the download has completed you will need to enter your root or priveledged account to run the installation application.

Enter the chosen download directory and enter the following, substituting the version number with the current version.

```
#installpkg sbopkg-0.35.0.tar.gz
```

Application Configuration



In most cases the application's default configuration will work, changes can be performed to target alternate repositories and change the target locations in the local filesystem if needed.

All configuration files for sbopkg are kept in the **/etc/sbopkg** directory. The directory contains the sbopkg.conf file which is the main coniguration file, the rename.d directory which contains a list of

files to be renamed for simplicity and the `repos.d` directory which contains the list of usable repositories.

Editing `sbopkg.conf`



Prior to editing the configuration file, read the manual file by issuing the following command in a terminal: **`man 5 sbopkg.conf`**

The `sbopkg.conf` file is well documented with comments explaining the various options, read the comments associated with the option that you wish to change to confirm that the syntax and chosen options are correct.

The only modifications that should be performed on this file without potential issues are to move the directories that the application will use, which are listed below:

- **OUTPUT** - The directory which will contain the completed packages
- **QUEUEDIR** - The queue directory which will contain your saved queue files
- **REPO_ROOT** - The directory which will the synchronized slackbuild scripts
- **SRCDIR** - The directory which will contain the downloaded source archives for all builds

You can optionally modify `REPO_BRANCH` and `REPO_NAME` variables if you wish to use a custom repository, however the official SlackBuilds.org repo for the current stable Slackware release is enabled by default.

Selecting Repos (Optional)



Prior to changing the repos, read the application authors notes about repo use at **`/etc/sbopkg/repos.d/README`** which will explain the configuration syntax

All files in the `repos.d` will be replaced as the `sbopkg` application is updated, if you make changes to any of the file you will need to backup your changes in another location so they can be reverted.

The `repos.d` directory currently contains the following files:

- **README** - The author's notes about repo use
- **40-sbo-repo** - The author's list of official SlackBuild.org repos and branches, this should not be changed as it will be overwritten as updates to `sbopkg` are installed
- **50-local.repo** - This file can point to a custom repo such as a centralized repo for a corporation or other organization
- **60-SBo-current.repo** - This is another custom repo file which by default is pointing to ponce's slackware-current script repo.

Using the Rename Option



Prior to changing the repos, read the application authors notes about use of renaming criteria at **/etc/sbopkg/renames.d/README** which will explain the configuration syntax

All files in the `renames.d` will be replaced as the `sbopkg` application is updated, if you make changes to any of the file you will need to backup your changes in another location so they can be reverted.

Sometimes it may be necessary to rename the SlackBuild archive for an application to avoid naming conflicts and simplify the organization of the packages in `sbopkg`. If you wish to apply alternate names to known SlackBuilds you will should first create a new file in the **/etc/sbopkg/renames.d/** directory that follows the debian style naming of `{number}-{name}.renames`, I recommend a simple name like `70-custom.renames` so the default file takes precedence.

The file contents are very simple, `OldName=NewName`. An example would be to rename the `i3` window manager SlackBuild to `i3wm` so it is better reflected as a window manager. Create a new file named **/etc/sbopkg/renames.d/70-custom.renames**, open the file in your favorite text editor and add the following:

```
# oldname=newname  
i3=i3wm
```

As you can see the comment was added to make future modifications easier and the syntax is also very simple to understand.

Application Use

The `sbopkg` application has two possible modes, CLI and GUI. Both have pros and cons, as with any tool, the user should utilize the balance of tools that best fit their needs.

Application CLI Use



The listed options may be out-of-date

Refer to the command `sbopkg -h` to confirm which options are available in your installed version.

The most basic and functional use of the **sbopkg** utility is through the command line, the only real limitation is lacking the ability to generate queue files.

First Use

After the application is installed, your first task is to create the necessary files and pull a copy of the active SlackBuilds.org repo.

To accomplish all of the listed tasks you simply need to issue the command to sync the repo.

```
root@darkstar#sbopkg -r
```

This will also create the necessary directories to hold the repo, cache, queue files and output packages.

Upon completion of the synchronization procedure you are free to build, install or inspect all slackbuild from <http://www.SlackBuilds.org> which have been downloaded into your local filesystem.

General Use

Basic Commands

- **-u** - check for an update to sbopkg.
- **-r** - Sync the local repo
- **-c** - check for updates to installed SBo packages
- **-d** - Download the source archive for the selected application(s).
- **-b** - Download the source and build packages for the selected application(s).
- **-i** - Download the source, build packages and install/upgrade packages for the selected application(s).
- **-k** - skip selected process for installed application(s)

Search and Display Options

- **-R** - Display the README for the selected application.
- **-g** - Search for package name that matches a string.
- **-s** - Search for a specific package and display the package information.

Cache and File Maintenance

- **-o** - Display the obsolete source files and prompt for deletion.
- **-P** - Display the cache files for uninstalled apps and prompt for deletion.

Building Applications from the SlackBuilds.org Repo

Before each set of uses it is recommended to re-sync the repo to guarantee that all updated and new SlackBuilds are available, this is again accomplished with the **-r** option.

For this example we will conduct all steps necessary to install the pysolfc application. As the goal is to download the source, build and install this application, you will use the **-i** option as shown below.

```
root@darkstar#sbopkg -i pysolfc
```

Upon submitting the command all steps will begin running and the application will display the verbose output, you will also be prompted if there is an error or the checksum does not match the file.



If you wish to install more than one application you will include all requested applications in package name argument within a single set of quotes, for example "pysolfc qrencode".

Updating Installed Applications/Libraries

To search for updates to applications and libraries that were installed from SlackBuilds you must first re-sync your repo to gather all updates. Upon completing that task you will issue the **-c** option, after a quick scan the list of potential updates will be displayed.

```
root@darkstar# sbopkg -c
[ Checking for potential updates ]
This may take a few moments. Press <ESC> to abort.
100%[=====]
Listing installed SBo repository for Slackware 14.0 packages and flagging
potential updates...

google-talkplugin:
  INSTALLED PACKAGE IS NEWER THAN REPO
  Installed version:  google-talkplugin-3.10.2.0-i386-1_SBo
  Repo version:  google-talkplugin-2.6.1.0-i486-1_SBo
  Note: repo version not obtainable by standard method, may be inaccurate.
libevent:
  POTENTIAL UPDATE
  Installed version:  libevent-2.0.10-i486-1_SBo
  Repo version:  libevent-2.0.21-i486-1_SBo
mathomatic:
  INSTALLED PACKAGE IS NEWER THAN REPO
  Installed version:  mathomatic-16.0.5-i486-1_SBo
  Repo version:  mathomatic-15.8.5-i486-1_SBo
warzone2100:
  INSTALLED PACKAGE IS NEWER THAN REPO
  Installed version:  warzone2100-3.1_rc3-i486-1_SBo
  Repo version:  warzone2100-2.3.8-i486-1_SBo

Potential update list complete.
```

You can then install build and install the newly updated applications with the standard install and build options.



Current Working Point

Application GUI Use

Queue Files

Making Sbopkg Handle Dependencies

Notes about Hard Drive Use

Sources

* Originally written by [crod](#) * Contributions by [mfillpot](#)

author [mfillpot](#), work in progress

From:
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:
<https://docs.slackware.com/wiki:user:mfillpot:sbopkg>

Last update: **2012/12/20 04:14 (UTC)**

