

Construyendo e instalando paquetes con sbopkg

Sbopkg es una herramienta de línea de comandos y de diálogo para sincronizar con el repositorio [SlackBuilds.org](https://www.slackbuilds.org) ("SBo"), una colección de scripts SlackBuild de terceros para construir paquetes Slackware. El programa tiene una interfaz basada en curses que le permite seleccionar y ensamblar los programas que desea compilar desde la fuente hasta los paquetes. También se puede usar de forma no interactiva en caso de que sepa de antemano cuál es su objetivo, en ese caso es "fire and forget (dispara y olvida)".

La siguiente información lo ayudará a instalar sbopkg y ejecutar Slackware.

Tenga en cuenta que sbopkg construye *paquetes Slackware*. Al igual que con cualquier programa de administración de paquetes Slackware, ¡tendrá que ser root para usar el programa! Ejecute

```
su -
```

para obtener un prompt root con el entorno correcto configurado.

Descargar sbopkg:

Descargue sbopkg desde: <https://www.sbopkg.org/downloads.php>

Puede obtener las fuentes y crear un paquete usted mismo, o simplemente tomar el paquete de Slackware ya hecho. La siguiente sección de este artículo asume que descargó el paquete listo para usar.

Instalar sbopkg

1. Suponiendo que el nombre del archivo del paquete sbopkg que descargó en el directorio actual es *sbopkg-0.38.0-noarch-1_wsr.tgz*, el comando `installpkg` para instalar sbopkg (o actualizarlo si ya estaba presente en su sistema) es el siguiente:

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

Configurar sbopkg

1. ¡Lea la documentación sbopkg !Vea <https://www.sbopkg.org/docs.php>, y también no olvide que hay una página de manual para sbopkg.
2. La primera vez que se ejecuta sbopkg, se le preguntará si está bien que el programa cree los archivos de configuración necesarios:

```
# sbopkg
```

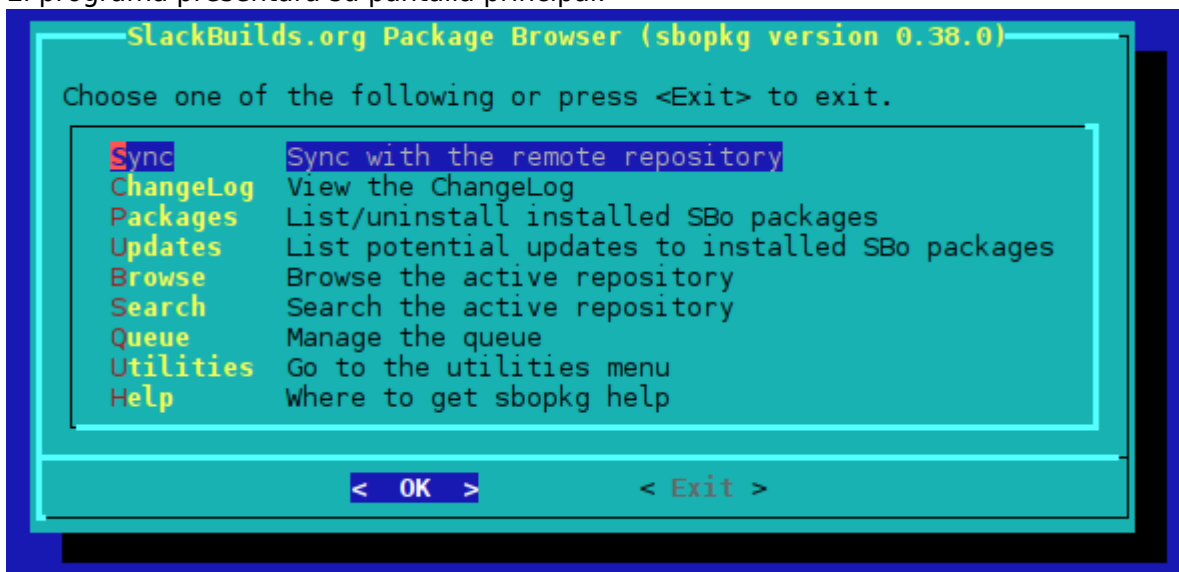
The following directories do not exist:

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)abort?:

Seleccione **C** para crear estos directorios requeridos.
El programa presentará su pantalla principal:



1. Su primera tarea es sincronizar con el repositorio de SlackBuilds.org, es decir, dejar que sbopkg cree una copia local de todas las entradas de SlackBuild en el servidor remoto, para la versión de Slackware que está ejecutando. Puede utilizar el primer elemento del menú en la pantalla principal basada en curses, o puede hacerlo desde la línea de comandos:

```
# 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. Ahora está listo para seleccionar los paquetes que desea "sbopkg" para compilar desde la fuente.

Usando Queuefiles con sbopkg

El programa sbopkg en sí mismo permite una gran cantidad de automatización: la interfaz de curses interactivas le ayuda a administrar la carga de descargar archivos de código fuente, y le permite seleccionar múltiples programas y luego compilar y empaquetar todos estos a la vez.

Pero puede automatizar este proceso aún más, utilizando el generador de cola de sqg - sbopkg (incluido en el paquete principal desde 0.38.0).

sqg puede generar un archivo de cola para cada uno o todos los paquetes en el repositorio de SBo:

```
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
```

Tenga en cuenta que cada vez que se anuncie una actualización pública o que haya un nuevo repositorio disponible, se recomienda ejecutar sqg -a para generar una cola actualizada a medida que se agreguen o eliminen dependencias.

Ejemplo de uso de sbopkg

Como ejemplo, instalemos [Gramps](#). El archivo de cola para Gramps enumera las siguientes dependencias, en orden de instalación:

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

Cargue el archivo de cola "gramps.sqf" en la interfaz de curses de sbopkg, y Gramps se construirá con éxito en su sistema Slackware 14.2 e instalará, junto con todas sus dependencias.

Alternativamente, puede utilizar la interfaz de línea de comandos menos interactiva:

```
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.

Referencias para sbopkg

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

Fuentes

- Escrito originalmente por [ldkraemer](#)
- Contribuciones por [Eric Hameleers](#)
- Contribuciones por [Willy Sudiarto Raharjo](#)
- Traducido por: [Victor](#) 2019/02/04 11:35 (UTC)

[howtos](#), [software](#), [sbo](#), [package management](#), [author ldkraemer](#)

Last update: 2019/04/05 es:howtos:slackware_admin:building_packages_with_sbopkg https://docs.slackware.com/es:howtos:slackware_admin:building_packages_with_sbopkg 22:29 (UTC)

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

Permanent link: https://docs.slackware.com/es:howtos:slackware_admin:building_packages_with_sbopkg

Last update: **2019/04/05 22:29 (UTC)**

