

# Criar e Instalar Pacotes com Sbopkg

**Sbopkg** é uma ferramenta de linha de comando e de GUI para sincronizar com o [SlackBuilds.org](https://slackbuilds.org) ("SBo"), uma coleção de scripts de terceiros do SlackBuild para construir pacotes do Slackware. O programa tem uma interface baseada em curses que lhe permite escolher e montar os programas que você quer compilar a partir do código fonte dos pacotes. Ele também pode ser usado de forma não interativa, caso você saiba de antemão qual é o seu objetivo - nesse caso, é "disparar e esquecer".

As informações a seguir ajudarão você a instalar o sbopkg e executá-lo no Slackware.



Note que o sbopkg cria pacotes *Slackware*. Como acontece com qualquer programa de gerenciamento de pacotes do Slackware, você terá que ser root para usar o programa! Execute

```
su -
```

para obter um terminal como root com o ambiente correto configurado.

## Baixar o sbopkg:

Baixe o sbopkg de: <https://www.sbopkg.org/downloads.php>

Você pode obter as fontes e construir um pacote sozinho, ou apenas pegar o pacote do Slackware já pronto. A seguinte secção deste artigo presume que você baixou o pacote pronto.

## Instalar o sbopkg

1. Assumindo que o nome do arquivo de pacote sbopkg que você baixou para o diretório atual é *sbopkg-0.38.0-noarch-1\_wsr.tgz*, o comando `installpkg` para instalar o sbopkg (ou atualizá-lo se ele já esta(va) presente em seu sistema) é o seguinte:

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

## Configurar o sbopkg

1. Leia a documentação do sbopkg! Veja <https://www.sbopkg.org/docs.php>, e também não se esqueça que existe uma página man do sbopkg.
2. A primeira vez que o sbopkg é executado, você será perguntado se está OK que o programa crie os arquivos de configuração necessários:

```
# sbopkg
```

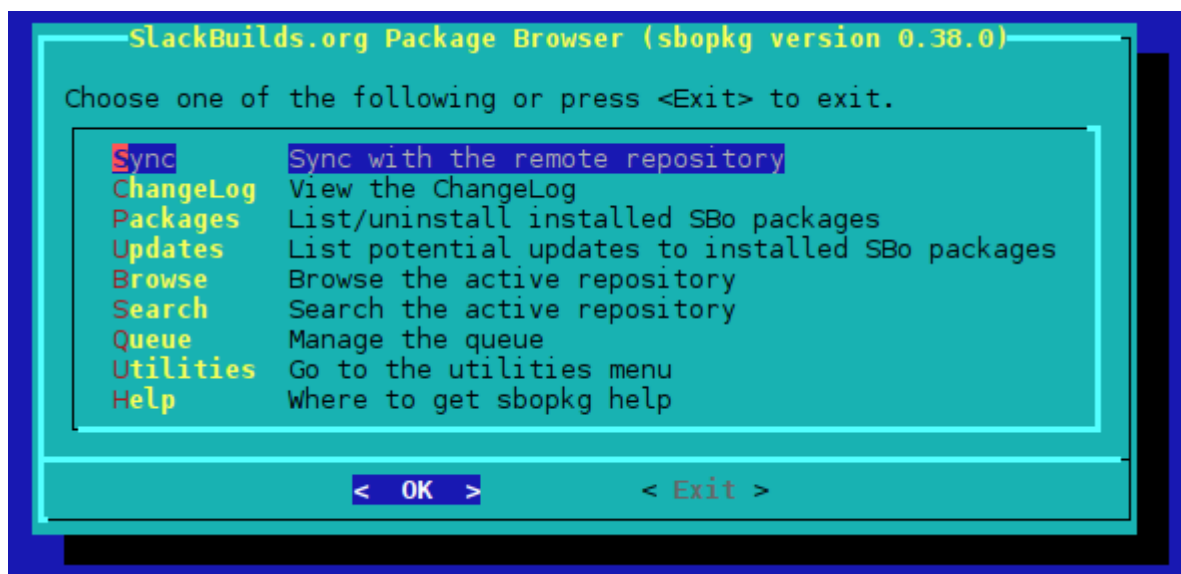
The following directories do not exist:

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?:

. Pressione **C** para criar os respectivos diretórios.  
O programa apresentará sua tela principal:



1. Sua primeira tarefa é sincronizar com o repositório do SlackBuilds.org - Ou seja, você permite que o sbopkg crie uma cópia local de todas as entradas do SlackBuild do servidor remoto, para a versão do Slackware que você está executando. Você pode usar o primeiro item de menu na tela principal baseada em curses, ou então você pode fazer isso a partir da linha de comando:

```
# 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. Agora você está pronto para selecionar os pacotes que você deseja que sbopkg construa a partir da fonte.

## Usar arquivos de fila de espera com sbopkg

O programa sbopkg permite uma grande quantidade de automação. A interface de curses interativa ajuda você a gerenciar o download dos arquivos de código-fonte e permitir que você selecione vários programas e, em seguida, compile e empacote todos de uma só vez.

Mas você pode automatizar ainda mais esse processo, usando o gerador sqg - sbopkg queueu (incluído no pacote principal desde 0.38.0).

O sqg pode gerar o arquivo de filas para cada um ou todos os pacotes no repositório do 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
```

Observe que toda vez que uma atualização pública é anunciada ou um novo repositório está disponível, é recomendado executar o sqg -a para gerar arquivos de fila atualizados, pois as dependências podem ser adicionadas ou removidas.

## Exemplo do Uso do sbopkg

Por exemplo, vamos instalar [Gramps](#). A lista com os arquivos para Gramps lista as seguintes dependências, em ordem de instalação:

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

Carregue o arquivo “gramps.sqf” na interface curses do sbopkg, e o Gramps será construído com sucesso no seu sistema Slackware 14.2 e instalado, junto com todas as suas dependências.

Como alternativa, você pode usar a interface de linha de comando menos interativa:

```
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
```

etc.

## Referências para sbopkg

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

## Fontes

- Original escrito por [ldkraemer](#)
- Contribuição de [Eric Hameleers](#)
- Contribuição de [Willy Sudiarto Raharjo](#)
- Tradução Português [MacgyverPT \(Miguel Rosa\)](#)

Last  
update:  
2020/11/19 pt-br:howtos:slackware\_admin:building\_packages\_with\_sbopkg [https://docs.slackware.com/pt-br:howtos:slackware\\_admin:building\\_packages\\_with\\_sbopkg](https://docs.slackware.com/pt-br:howtos:slackware_admin:building_packages_with_sbopkg)  
10:08  
(UTC)

---

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

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

Permanent link:  
[https://docs.slackware.com/pt-br:howtos:slackware\\_admin:building\\_packages\\_with\\_sbopkg](https://docs.slackware.com/pt-br:howtos:slackware_admin:building_packages_with_sbopkg)

Last update: **2020/11/19 10:08 (UTC)**

