Customize Slax

Even if Slax is running from a read-only media such as CD/DVD, it still provides fully writable root filesystem (with changes stored in memory), allowing you to install additional software from online repository or other sources. Unless you are using Presistent changes, everything you install is kept in memory only and is lost after you reboot. The tools to add software to Slax as well as tools to make the changes permanent are described below.

Adding software with apt

Slax based on Debian fully supports `apt` command, which is a powerful command-line tool suitable for tasks such as installation of new software packages or upgrading of existing ones. For example, in order to install vlc video player, simply execute `apt install vlc` and you are all set up, Slax will download the requested package including all libraries and other dependencies which are necessary to run it. If you are unsure what package name to install, you can search using `apt search your_keyword`. Keep in mind that apt will automatically download its database of available packages when you run it for the first time. If your system is continuously running for several days, you may need to run `apt update` manually from time to time to ensure the database is up to date.

Adding software with slackpkg

Slax based on Slackware supports `slackpkg` command with slackpkg+ extension, which provides the ability to install precompiled software from official Slackware repository as well as several other unofficial repositories, especially slackonly repo which includes precompiled packages created for slackbuilds.org. For example, in order to install MPlayer video player, simply execute `slackpkg install mplayer` and it will be downloaded and installed. Slackpk does not resolve dependencies so you need to do that manually. If you wish to see what libraries are required to run mplayer, use `showdeps /usr/bin/mplayer`. You will get list of missing dependencies including packages which contain them. Keep in mind that you need to run `slackpkg update` before first use.

Adding software manually

If you prefer to add software to Slax manually, you can do so by downloading and compiling source codes from the internet. Just remember that in order to be able to compile source codes in Slax, you will need to `apt install build-essential`, which provides gcc compiler, make, and other tools necessary for that task, or install the tools in any other way. If you are downloading sources from github using `git clone`, you may also need to install git.

Making your changes permanent

If you are running Slax from a writable media, your changes are stored permanenly and there is no need for any further steps. However, if your are running Slax from a read-only media such as CD/DVD, or if you selected to disable Persistent changes feature during boot, there is still a way to save your changes by creating a module. At any point, when you have your desired applications installed and configured, use `savechanges` command to save all modifications made to the current system into a module. The command requires one parameter, which is a path to file where the changes will be saved. So for example, use `savechanges /tmp/mychanges.sb`. All modifications you made to the system will be stored, except of some predefined files and directories such as temporary files, caches, and so on.

Once your changes are stored in a module, all you need is to include this module in your Slax and it will be automatically activated during boot. Copy the .sb module to `/slax/modules/` directory and reboot. If you are running Slax from a writable media such as USB drive, you can copy the module directly to

`/run/initramfs/memory/data/slax/modules/`, which is already mapped to your USB disk. If you are running Slax from a read-only media such as CD/DVD, you may need to re-generate the ISO image by using `genslaxiso` command.

If you wish to examine the contents of your module or modify files in it, you can use two commands available in Slax: `sb2dir` to convert your module into a directory, and then `dir2sb` to convert it back into a .sb file. Important thing to keep in mind is that a module converted into a directory is stored in memory (technically the directory is overmounted with tmpfs), so you may find it impossible to remove that directory with rmdir, even if empty (you will get EBUSY error). Use `rmsbdir` command to remove the directory created by sb2dir.

Note that these commands worked a bit differently in older Slax versions, so make sure you are using Slax 9.6.4 or newer.

Activating modules on the fly

If you have a Slax module (.sb) and you wish to activate it while running Slax, without rebooting, you can use `slax activate` command. Similarly, you can deactivate any module with `slax deactivate`. However, if any of the files from your module are open or used, deactivation will not be possible, so make sure to kill all processes which may use the files from your module before you attempt to deactivate it.

To recap, here are the commands we just learned

`apt search keyword`	to search for available package by keyword
`apt install package`	to download and install package
`savechanges target.sb`	to save all filesystem modifications made during current session into a module file
`sb2dir module.sb`	to convert a .sb module into a directory
`dir2sb directory.sb`	to convert a directory into .sb module
`rmsbdir directory.sb`	to erase directory created by sb2dir
`genslaxiso slax.iso module.sb`	to generate Slax iso file with new modules
`slax activate module.sb`	to activate module on the fly
`slax deactivate module.sb`	to deactivate module on the fly

Example: make module with new wallpaper

```
# Slax uses wallpaper from /usr/share/wallpapers/slax_wallpeper.jpg
# So we are going to replace this file
# First, create a temporary module tree
# Note that the directory itself has .sb extension
DIR=/tmp/module.sb
mkdir $DIR
mkdir -p $DIR/usr/share/wallpapers/

# Now copy wallpaper file to that directory
cat new_wallpaper.jpg > $DIR/usr/share/wallpapers/slax_wallpaper.jpg

# Finally, convert the module directory to a file
# The directory will be erased and module file with the same name will be created
dir2sb $DIR
```

Example: add FireFox to Slax

```
# First, we boot fresh Slax and try apt
apt install firefox

# As you can see, there is no package called 'firefox'. Lets try to search for the word
apt search firefox | more

# Ah, now we get something. The package is called firefox-esr. So lets install it
apt install firefox-esr

# We want it called just Firefox, so lets edit the desktop file responsible for the name
sed -i -r "s/^Name=FireFox ESR/Name=Firefox/" /usr/share/applications/firefox-esr.desktop

# Now save the newly installed software including our changes to a module
# In our example, the module will be saved in current directory
savechanges firefox.sb

# If we are running from readonly CD, we can create a new ISO with our module
# Then just burn the ISO to new CD and Firefox will be available when you boot it
# In our example, the ISO file will be created in current directory
genslaxiso slax.iso firefox.sb

# If we are running Slax from USB drive, we can put it directly to modules folder
# and that's it, firefox will be available after reboot
mv firefox.sb /run/initramfs/memory/data/slax/modules/
```

Read more

Quick links

Changelog
Next version
Blog
Support Slax
Privacy policy

Contact

Slax discord

Google group

Facebook