



Slackware:
Quickly and Easily Manage
Your KDE SC Hacking

Camp KDE 2011
Vincent Batts, vbatts@slackware.com

Overview

- What brought me to Slackware?
- Dependencies, and their components
- Vanilla involvement with upstream source
- Manage your builds
- Additional packages
- Recovering your system



What brought me to Slackware?



What brought me to Slackware?

- Learn more about KDE and where is it headed
- Familiarity with new dependencies
- Simplicity in component handling and upgrades

Dependencies, and their components

- Immediate Dependencies are present
- Packages are not split out
(bin, doc, dev, lib)
- Monolithic style

Dependencies, and their components

- Immediate Dependencies are present

PACKAGE SERIES SELECTION

Now it's time to select which general categories of software to install on your system. Use the spacebar to select or unselect the software you wish to install. You can use the up and down arrows to see all the possible choices. Recommended choices have been preselected. Press the ENTER key when you are finished.

<input checked="" type="checkbox"/>	A	Base Linux system	
<input checked="" type="checkbox"/>	AP	Various Applications that do not need X	
<input checked="" type="checkbox"/>	D	Program Development (C, C++, Lisp, Perl, etc.)	
<input checked="" type="checkbox"/>	E	GNU Emacs	
<input checked="" type="checkbox"/>	F	FAQ lists, HOWTO documentation	
<input checked="" type="checkbox"/>	K	Linux kernel source	
<input checked="" type="checkbox"/>	KDE	Qt and the K Desktop Environment for X	
<input type="checkbox"/>	KDEI	International language support for KDE	
<input checked="" type="checkbox"/>	L	System Libraries (needed by KDE, GNOME, X, and more)	60%

(+)

The A (base) series contains the kernel and main system utilities.

SELECT PROMPTING MODE

Now you must select the type of prompts you'd like to see during the installation process. If you have the drive space, the 'full' option is quick, easy, and by far the most foolproof choice. The 'newbie' mode provides the most information but is much more time-consuming (presenting the packages one by one) than the menu-based choices. Otherwise, you can pick packages from menus using 'expert' or 'menu' mode. Which type of prompting would you like to use?

<input checked="" type="checkbox"/>	full	Install everything (6.5+ GB of software, RECOMMENDED!)	
<input type="checkbox"/>	terse	Like 'full', but display one line per package during install	
<input type="checkbox"/>	menu	Choose individual packages from interactive menus	
<input type="checkbox"/>	expert	This is actually the same as the "menu" option	
<input type="checkbox"/>	newbie	Use verbose prompting (the X series takes one year)	
<input type="checkbox"/>	custom	Use custom tagfiles in the package directories	
<input type="checkbox"/>	tagpath	Use tagfiles in the subdirectories of a custom path	87%

(+)

NOTICE

Do Not Fear The Command Line :)

“If you continue this simple practice every day, you will obtain some wonderful power. Before you attain it, it is something wonderful, but after you attain it, it is nothing special.”

-- Shunryu Suzuki

Vanilla involvement in the upstream source

- `./configure && make`
- `mkdir build && \
 cd build && \
 cmake .. && \
 make`

Managing your own builds

- `makepkg(8)` for system or user (`kdesrc-build`)
- SlackBuilds
- Tagging your packages/builds

Managing your own builds

- makepkg(8)

```
$> make install DESTDIR=`pwd`/tmp
```

```
$> cd tmp/
```

```
$> makepkg -l y ../<pkg_name>.tgz
```

```
$> sudo upgradepkg --reinstall  
../<pkg_name>.tgz
```

Managing your own builds

- makepkg(8)

```
$> make install DESTDIR=`pwd`/tmp
```

```
$> cd tmp/
```

```
$> makepkg -l y ../<pkg_name>.tgz
```

```
$> sudo upgradepkg --reinstall  
../<pkg_name>.tgz
```

Managing your own builds

- explodepkg(8) + makepkg(8)
\$> mkdir tmp && cd tmp/
\$> explodepkg ../<pkg_name>.tgz
[...]
\$> makepkg -l y ../<pkg_name>.tgz
\$> sudo upgradepkg --reinstall
../<pkg_name>.tgz

Managing your own builds

- SlackBuilds
an *.sh script, with packing intentions

Managing your own builds

- Tagging your packages/builds
 <PKG-NAME>-<VERSION>-<ARCH>-
 <BUILD>_<TAG>.t?z

Additional Packages

<http://slackbuilds.org>

and

<http://sbopkg.org>

(2700+ packages available)



Recovering from a mess

- Managed core set of packages
- Benefits of the lack of dependency chains

Recovering from a mess

- Have a local mirror/DVD
 - Pkgtools

```
#> upgradepkg --reinstall */*.t?z
```
 - slackpkg

```
#> slackpkg clean-system
```
 - slack-utils

```
$> slp -t mytag | wc -l
```
- 23