

Enlightenment DR17

What is E17?

[Enlightenment DR17](#) (E17) is the long awaited successor of E16, a [window manager](#) which was distributed in Slackware 10 years ago. It is classed as a “desktop shell”, providing the things you need to operate your desktop (or laptop), but not a whole suite of applications. This includes launching applications, managing their windows, and doing other system tasks like suspending, rebooting, managing files, etc.

E17 doesn't come alone. Actually, it uses the [Enlightenment Foundation Libraries](#) (EFL) which can be used to develop other applications like in the Qt/KDE world for instance. There are, as in February 2013, 15 libraries in EFL and a lot of external modules or applications that can be used with E17.

Installing E17

Pre-built packages

SlackE17

The easiest and fastest way to get E17 running on Slackware is to install the E17 packages from the [SlackE17 project](#). You just need to have the latest release of Slackware, Slackware64 or ArmedSlack.

Just [download](#) the tarball corresponding to your architecture (i486, x86_64 or ARM), unpack it and read the doc. To install the packages, you have several options:

- Install everything using:

```
# upgradepkg --install-new e17/*.txz
```

- Use the supplied script to install only E17 and its dependencies using:

```
# ./slacke17-install.sh std
```

- Use the supplied script to install everything using:

```
# ./slacke17-install.sh max
```

- Or you can just:

```
# cd e17 && pkgtool
```

Compilation

First you need to install the dependencies needed to compile Enlightenment. These can be found in

SlackBuilds-website or with sbopkg:

- check
- lua
- luajit
- bullet

Once you have those packages installed we can proceed to make a build directory and commence with the fun stuff!

```
# mkdir -p /build/e22/core
# cd /build/e22/core
# git clone https://git.enlightenment.org/core/efl.git
# git clone https://git.enlightenment.org/core/enlightenment.git
# cd /build/e22/core/efl
# ./autogen.sh --prefix=/opt/e22 --disable-systemd --disable-avahi
```

Make symlink for dbus (installer tells you to do this so please check output if it has been updated just to be sure).

```
# ln -s /opt/e22/share/dbus-1/services/org.enlightenment.Ethumb.service
/usr/share/dbus-1/services/org.enlightenment.Ethumb.service
# make && make install
```

Now we need to add Enlightenment library directory to the system so it knows where to look for the libraries.

```
# echo "/opt/e22/lib" >> /etc/ld.so.conf
```

Load the new configuration.

```
# ldconfig
```

An environmental variable need to be set for PKGCONFIG, this is only needed when building and doesnt need to be permanent.

```
# export
PKG_CONFIG_PATH=/opt/e22/lib/pkgconfig:/usr/lib64/pkgconfig:/usr/share/pkgco
nfig
```

Add enlightenment bin to PATH for build, also make it persist after reboot.

```
# export PATH=/opt/e22/bin:$PATH
```

```
# vi /etc/profile
```

Add “/opt/e22/bin” to your PATH.

Start building Enlightenment.

```
# cd /build/e22/core/enlightenment
# meson --prefix=/opt/e22 -Dsystemd=false -Dvahi=false . build
# meson configure build
# ninja -C build install
```

(Optional, build Terminology. Terminology is the default terminal in Enlightenment.)

```
# mkdir /build/e22/apps
# cd /build/e22/apps
# git clone https://git.enlightenment.org/apps/terminology.git
# cd terminology
# meson --prefix=/opt/e22 . build
# meson configure build
# ninja -C build install
```

To make your sessionmanager find Enlightenment you need to create a .desktop file in the right directory.

```
# vi /usr/share/xsession/enlightenment.desktop
```

Should look something like this (please check already existing .desktop files for other language examples and please add relevant lines for your language):

```
[Desktop Entry]
Encoding=UTF-8
Type=XSession
Exec=/opt/e22/bin/enlightenment_start
TryExec=/opt/e22/bin/enlightenment_start
DesktopNames=Enlightenment
```

```
Name=Enlightenment
Comment[en_GB]=Enlightenment by Enlightenment
Comment[sv]=Enlightenment av Enlightenment
X-KDE-PluginInfo-Version=0.22.9
```

If you want to start enlightenment with startx (xinit): /etc/X11/xinit/xinitrc.e22

```
#!/bin/sh

userresources=$HOME/.Xresources
usermodmap=$HOME/.Xmodmap
sysresources=/etc/X11/xinit/.Xresources
sysmodmap=/etc/X11/xinit/.Xmodmap

# merge in defaults and keymaps

[ -f $sysresources ] && xrdp -merge $sysresources
[ -f $sysmodmap ] && xmodmap $sysmodmap
[ -f $userresources ] && xrdp -merge $userresources
[ -f $usermodmap ] && xmodmap $usermodmap

if [ -z "$DESKTOP_SESSION" -a -x /usr/bin/ck-launch-session ]; then
    export DESKTOP_SESSION="e22"
    exec ck-launch-session dbus-launch --exit-with-session enlightenment_start
else
    export DESKTOP_SESSION="e22"
    exec enlightenment_start
fi
```

SlackE17

SlackE17 provides all the source code and SlackBuilds in one tarball so you can rebuild everything easily. You can build the packages one by one, but it's more efficient to use the BUILD/slacke17-build.sh script. You can pass some options to the script:

- To get debug symbols:

```
# DEBUG=1 ./slacke17-build.sh
```

- To optimize for your box:

```
# ECFLAGS="-march=native -mtune=native" ./slacke17-build.sh
```

SBo

SBo provides SlackBuilds for EFL and E17, but you will have to download all the source code by

yourself and build the packages one by one.

Using E17

You need to update your `.xinitrc` using `xwmconfig` and choosing `xinitrc.e17`. Most of the configuration can be made by the right click menu. If you want to manage who can use the system menu, you should take a look at `/etc/enlightenment/sysactions.conf`.

Sources

[howtos](#), [e17](#), [author ngc891](#)

From:

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

Permanent link:

https://docs.slackware.com/howtos:window_managers:e17

Last update: **2019/02/21 10:40 (UTC)**

