```
#!/bin/bash

# This script allows you to easily and safely install Enlightenment, along with
# other applications based on the Enlightenment Foundation Libraries (EFL),
# in your Ubuntu LTS desktop system.

# Supported distribution: Ubuntu Noble Numbat.

# ELUCIDATE.SH takes care of downloading, configuring and building everything you
# need to enjoy the very latest version of the Enlightenment environment
# (DEB packages tend to lag far behind). Once installed, you can update
# your Enlightenment desktop whenever you like.

# Facultative: Additional steps may be taken in order to achieve optimal results.
# Please refer to the comments of the build_plain() function.

# Tip: Set your terminal scrollback to unlimited so that you can scroll up
# to look at earlier output at any time.

# See README.md for instructions on how to use this script.
# See also the repository's wiki for post-installation hints.

# Heads up!
# Enlightenment programs compiled from git source code will inevitably come into conflict
# with the ones installed from DEB packages. Therefore, remove any previous binary
# installations of EFL, Enlightenment and related applications before running
# this script.

# ELUCIDATE.SH is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License,
# in memory of Aaron Swartz.
# See https://creativecommons.org/licenses/by-sa/4.0/

# Got a GitHub account? Please consider starring our repositories to show your support.
# Thank you!

# ---------------
# USER VARIABLES
# ---------------
# (These variables are not available to be used outside of this script.)

BOLD="\e[1m"    # Bold text.
ITAL="\e[3m"    # Italic text.
BLDR="\e[1;31m" # Bold red text.
BLDG="\e[1;32m" # Bold green text.
BRTC="\e[1;96m" # Bright cyan text.
BLDP="\e[1;35m" # Bold purple text.
BLDY="\e[1;33m" # Bold yellow text.
LOWG="\e[2;32m" # Low intensity green text.
LOWP="\e[2;35m" # Low intensity purple text.
LOWY="\e[2;33m" # Low intensity yellow text.
OFF="\e[0m"     # Turn off ANSI colors and formatting.

DLDIR=$(xdg-user-dir DOWNLOAD)
DOCDIR=$(xdg-user-dir DOCUMENTS)
SCRFLDR=$HOME/.elucidate
REBASEF="git config pull.rebase false"
AUTGN="./autogen.sh --prefix=$PREFIX"
SNIN="sudo ninja -C build install"
SMIL="sudo make install"
DISTRO=$(lsb_release -sc)
DDCTL=2.0.0

# Build dependencies, recommended and script-related packages.
DEPS="acpid arc-theme automake build-essential ccache check cmake cowsay doxygen fonts-noto \
freeglut3-dev gettext graphviz gstreamer1.0-plugins-bad gstreamer1.0-plugins-ugly hwdata \
i2c-tools imagemagick libaom-dev libasound2-dev libavahi-client-dev libavif-dev \
libblkid-dev libbluetooth-dev libegl1-mesa-dev libexif-dev libfontconfig-dev libdrm-dev \
libfreetype-dev libfribidi-dev libgbm-dev libgeoclue-2-dev libgif-dev \
libgraphviz-dev libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev \
libharfbuzz-dev libheif-dev libi2c-dev libibus-1.0-dev libinput-dev libinput-tools \
```

```
libjansson-dev libjpeg-dev libjson-c-dev libkmod-dev liblua5.2-dev liblz4-dev \
libmenu-cache-dev libmount-dev libopenjp2-7-dev libosmesa6-dev libpam0g-dev \
libpoppler-cpp-dev libpoppler-dev libpoppler-private-dev libpulse-dev libraw-dev \
librsvg2-dev libsdl1.2-dev libscim-dev libsndfile1-dev libspectre-dev \
libssl-dev libsystemd-dev libtiff5-dev libtool libudev-dev libudisks2-dev \
libunibreak-dev libunwind-dev libusb-1.0-0-dev libwebp-dev \
libxcb-keysyms1-dev libxcursor-dev libxinerama-dev libxkbcommon-x11-dev \
libxkbfile-dev lxmenu-data libxrandr-dev libxss-dev libxtst-dev libyuv-dev \
lolcat manpages-dev manpages-posix-dev meson ninja-build papirus-icon-theme \
texlive-base unity-greeter-badges valgrind wayland-protocols wmctrl xdotool"

# Latest source code available.
CLONEFL="git clone https://git.enlightenment.org/enlightenment/efl.git"
CLONETY="git clone https://git.enlightenment.org/enlightenment/terminology.git"
CLONENL="git clone https://git.enlightenment.org/enlightenment/enlightenment.git"
CLONEPH="git clone https://git.enlightenment.org/enlightenment/ephoto.git"
CLONERG="git clone https://git.enlightenment.org/enlightenment/rage.git"
CLONEVI="git clone https://git.enlightenment.org/enlightenment/evisum.git"
CLONEXP="git clone https://git.enlightenment.org/enlightenment/express.git"
CLONECR="git clone https://git.enlightenment.org/enlightenment/ecrire.git"
CLONEVE="git clone https://git.enlightenment.org/enlightenment/enventor.git"
CLONEDI="git clone https://git.enlightenment.org/enlightenment/edi.git"
CLONENT="git clone https://git.enlightenment.org/vtorri/entice.git"
CLONEFT="git clone https://git.enlightenment.org/enlightenment/enlightenment-module-forecasts.git"
CLONEPN="git clone https://git.enlightenment.org/enlightenment/enlightenment-module-penguins.git"
CLONETE="git clone https://github.com/dimmus/eflete.git"

# “MBS” stands for Meson Build System.
PROG_MBS="
efl
terminology
enlightenment
ephoto
rage
evisum
express
ecrire
enventor
edi
entice
enlightenment-module-forecasts
enlightenment-module-penguins
eflete"

# Bug reporting: Uncomment the following (remove the leading # character) to force messages to
# display in English during the build process.
#
# export LC_ALL=C

# ---------
# FUNCTIONS
# ---------

# Audible feedback (event, sudo prompt...) on most systems.
beep_dl_complete() {
  aplay --quiet /usr/share/sounds/sound-icons/glass-water-1.wav 2>/dev/null
}

beep_attention() {
  aplay --quiet /usr/share/sounds/sound-icons/percussion-50.wav 2>/dev/null
}

beep_question() {
  aplay --quiet /usr/share/sounds/sound-icons/guitar-13.wav 2>/dev/null
}

beep_exit() {
  aplay --quiet /usr/share/sounds/sound-icons/pipe.wav 2>/dev/null
}

beep_ok() {
```

```
  aplay --quiet /usr/share/sounds/sound-icons/trumpet-12.wav 2>/dev/null
}

# Hints.
# 1: A no frill, plain build.
# 2: A feature-rich, decently optimized build on Xorg; recommended for most users.
# 3: Similar to the above, but running Enlightenment as a Wayland compositor is still considered experimental
# Avoid the third option with Nvidia drivers.
#
menu_selec() {
  if [ $INPUT -lt 1 ]; then
    echo
    printf "1  $BLDG%s $OFF%s\n\n" "INSTALL the Enlightenment ecosystem now"
    printf "2  $LOWP%s $OFF%s\n\n" "Update and rebuild the ecosystem in release mode"
    printf "3  $LOWY%s $OFF%s\n\n" "Update and rebuild the ecosystem with Wayland support"

    sleep 1 && printf "$ITAL%s $OFF%s\n\n" "Or press Ctrl+C to quit."
    read INPUT
  fi
}

selec_menu() {
  if [ $INPUT -lt 1 ]; then
    echo
    printf "1  $LOWG%s $OFF%s\n\n" "Install the Enlightenment ecosystem now"
    printf "2  $BLDP%s $OFF%s\n\n" "Update and rebuild the ecosystem in RELEASE mode"
    printf "3  $BLDY%s $OFF%s\n\n" "Update and rebuild the ecosystem with WAYLAND support"

    sleep 1 && printf "$ITAL%s $OFF%s\n\n" "Or press Ctrl+C to quit."
    read INPUT
  fi
}

# Check binary dependencies.
bin_deps() {
  sudo apt update && sudo apt full-upgrade

  if ! sudo apt install --no-install-recommends $DEPS; then
    printf "\n$BLDR%s %s\n" "CONFLICTING OR MISSING DEB PACKAGES"
    printf "$BLDR%s %s\n" "OR DPKG DATABASE IS LOCKED."
    printf "$BLDR%s $OFF%s\n\n" "SCRIPT ABORTED."
    beep_exit
    exit 1
  fi
}

# Check source dependencies.
cnt_dir() {
  COUNT=$(find . -mindepth 1 -maxdepth 1 -type d | wc -l)

  if [ ! -d efl ] || [ ! -d enlightenment ]; then
    printf "\n$BLDR%s %s\n" "FAILED TO DOWNLOAD MAIN COMPONENT."
    printf "$BLDR%s $OFF%s\n\n" "SCRIPT ABORTED."
    beep_exit
    exit 1
    # You can try downloading the missing file(s) manually (see CLONEFL or CLONENL), then relaunch
    # the script and select option 1 again; or relaunch the script at a later time.
    # In both cases, be sure to enter the same path for the Enlightenment source
    # folders as you previously used.
  fi

  case $COUNT in
  14)
    printf "$BLDG%s $OFF%s\n\n" "All programs have been downloaded successfully."
    beep_dl_complete
    sleep 2
    ;;
  0)
    printf "\n$BLDR%s %s\n" "OOPS! SOMETHING WENT WRONG."
    printf "$BLDR%s $OFF%s\n\n" "SCRIPT ABORTED."
    beep_exit
```

```
    exit 1
    ;;
  *)
    printf "\n$BLDY%s %s\n" "WARNING: ONLY $COUNT OF 14 PROGRAMS HAVE BEEN DOWNLOADED!"
    printf "\n$BLDY%s $OFF%s\n\n" "WAIT 12 SECONDS OR HIT CTRL+C TO EXIT NOW."
    beep_attention
    sleep 12
    ;;
  esac
}

mng_err() {
  printf "\n$BLDR%s $OFF%s\n\n" "BUILD ERROR--TRY AGAIN LATER."
  beep_exit
  exit 1
}

# Timestamp: See the date man page to convert epoch to human-readable date
# or visit https://www.epochconverter.com/
#
# To restore a backup, use the same commands that were executed but with
# the source and destination reversed, similar to this:
# cp -aR /home/riley/Documents/ebackups/E_1703833338/.elementary/ /home/riley/
# cp -aR /home/riley/Documents/ebackups/E_1703833338/.e/ /home/riley/
# (Then press Ctrl+Alt+End to restart Enlightenment if you are currently logged into.)
#
e_bkp() {
  TSTAMP=$(date +%s)
  mkdir -p $DOCDIR/ebackups

  mkdir $DOCDIR/ebackups/E_$TSTAMP &&
    cp -aR $HOME/.elementary $DOCDIR/ebackups/E_$TSTAMP &&
    cp -aR $HOME/.e $DOCDIR/ebackups/E_$TSTAMP
  sleep 2
}

e_tokens() {
  echo $(date +%s) >>$HOME/.cache/ebuilds/etokens

  TOKEN=$(wc -l <$HOME/.cache/ebuilds/etokens)
  if [ "$TOKEN" -gt 2 ]; then
    echo
    # Questions: Enter either y or n, or press Enter to accept the default value (capital letter).
    beep_question
    read -t 12 -p "Do you want to back up your Enlightenment settings now? [y/N] " answer
    case $answer in
    y | Y)
      e_bkp
      ;;
    n | N)
      printf "\n$ITAL%s $OFF%s\n\n" "(no backup made... OK)"
      ;;
    *)
      printf "\n$ITAL%s $OFF%s\n\n" "(no backup made... OK)"
      ;;
    esac
  fi
}

rstrt_e() {
  if [ "$XDG_CURRENT_DESKTOP" == "Enlightenment" ]; then
    enlightenment_remote -restart
    if [ -x /usr/bin/spd-say ]; then
      spd-say --language Rob 'enlightenment is awesome'
    fi
  fi
}

# BEFORE EXECUTING THE SCRIPT...
#
# Add optional JXL support?
```

```
# For best results, jpeg xl has to be compiled from source. If you really need jxl
# support in efl, please follow the instructions below:
# https://gist.github.com/batden/0f45f8b8578ec70ee911b920b6eacd39
#
# Then change the option “-Devas-loaders-disabler=jxl” to
# "-Devas-loaders-disabler=” whenever it's applicable.
#
# Note: If building jxl is too much of a hassle for you, then install
# the libjxl-dev package instead (this older version still works).
#
# Fetch EDI's additional dependencies?
# If you want edi to compile, you will also need to install the packages
# listed in the link below:
# https://gist.github.com/batden/99a7ebdd5ba9d9e83b2446ab5f05f3dc
#
build_plain() {
  sudo ln -sf /usr/lib/x86_64-linux-gnu/preloadable_libintl.so /usr/lib/libintl.so
  sudo ldconfig

  for I in $PROG_MBS; do
    cd $ESRCDIR/enlighten/$I
    printf "\n$BOLD%s $OFF%s\n\n" "Building $I..."

    case $I in
    efl)
      if [ $DISTRO == noble ]; then
        meson setup build -Dbuildtype=plain \
          -Dfb=true \
          -Dbuild-tests=false \
          -Dlua-interpreter=lua \
          -Devas-loaders-disabler=jxl \
          -Dglib=true
        ninja -C build || mng_err
      fi
      ;;
    enlightenment)
      meson setup build -Dbuildtype=plain
      ninja -C build || mng_err
      ;;
    edi)
      meson setup build -Dbuildtype=plain \
        -Dlibclang-headerdir=/usr/lib/llvm-11/include \
        -Dlibclang-libdir=/usr/lib/llvm-11/lib
      ninja -C build
      ;;
    eflete)
      meson setup build -Dbuildtype=plain \
        -Dwerror=false
      ninja -C build
      ;;
    *)
      meson setup build -Dbuildtype=plain
      ninja -C build
      ;;
    esac

    beep_attention
    $SNIN
    sudo ldconfig
  done
}

rebuild_optim() {
  ESRCDIR=$(cat $HOME/.cache/ebuilds/storepath)
  bin_deps
  e_tokens

  cd $ESRCDIR/rlottie
  printf "\n$BOLD%s $OFF%s\n\n" "Updating rlottie..."
  git reset --hard &>/dev/null
  $REBASEF && git pull
```

```
  echo
  sudo chown $USER build/.ninja*
  meson setup --reconfigure build -Dbuildtype=release \
    -Dexample=false
  ninja -C build
  beep_attention
  $SNIN
  sudo ldconfig

  for I in $PROG_MBS; do
    cd $ESRCDIR/enlighten/$I
    printf "\n$BOLD%s $OFF%s\n\n" "Updating $I..."
    git reset --hard &>/dev/null
    $REBASEF && git pull

    case $I in
    efl)
      if [ $DISTRO == noble ]; then
        sudo chown $USER build/.ninja*
        meson setup --reconfigure build -Dbuildtype=release \
          -Dnative-arch-optimization=true \
          -Dfb=true \
          -Dharfbuzz=true \
          -Dlua-interpreter=lua \
          -Delua=true \
          -Dbindings=lua,cxx \
          -Devas-loaders-disabler=jxl \
          -Dglib=true \
          -Dopengl=full \
          -Ddrm=false \
          -Dwl=false \
          -Dbuild-tests=false
        ninja -C build || mng_err
      fi
      ;;
    enlightenment)
      sudo chown $USER build/.ninja*
      meson setup --reconfigure build -Dbuildtype=release \
        -Dwl=false
      ninja -C build || mng_err
      ;;
    edi)
      sudo chown $USER build/.ninja*
      meson setup --reconfigure build -Dbuildtype=release \
        -Dlibclang-headerdir=/usr/lib/llvm-11/include \
        -Dlibclang-libdir=/usr/lib/llvm-11/lib
      ninja -C build
      ;;
    eflete)
      sudo chown $USER build/.ninja*
      meson setup --reconfigure build -Dbuildtype=release \
        -Denable-audio=true -Dwerror=false
      ninja -C build
      ;;
    *)
      sudo chown $USER build/.ninja*
      meson setup --reconfigure build -Dbuildtype=release
      ninja -C build
      ;;
    esac

    beep_attention
    $SNIN
    sudo ldconfig
  done
}

rebuild_wayld() {
  if [ "$XDG_SESSION_TYPE" == "tty" ] && [ "$XDG_CURRENT_DESKTOP" == "Enlightenment" ]; then
    printf "\n$BLDR%s $OFF%s\n\n" "PLEASE LOG IN TO THE DEFAULT DESKTOP ENVIRONMENT TO EXECUTE THIS SCRIPT."
    beep_exit
```

```
  exit 1
fi

ESRCDIR=$(cat $HOME/.cache/ebuilds/storepath)
bin_deps
e_tokens

cd $ESRCDIR/rlottie
printf "\n$BOLD%s $OFF%s\n\n" "Updating rlottie..."
git reset --hard &>/dev/null
$REBASEF && git pull
echo
sudo chown $USER build/.ninja*
meson setup --reconfigure build -Dbuildtype=release \
  -Dexample=false
ninja -C build
beep_attention
$SNIN
sudo ldconfig

for I in $PROG_MBS; do
  cd $ESRCDIR/enlighten/$I
  printf "\n$BOLD%s $OFF%s\n\n" "Updating $I..."
  git reset --hard &>/dev/null
  $REBASEF && git pull

  case $I in
  efl)
    if [ $DISTRO == noble ]; then
      sudo chown $USER build/.ninja*
      meson setup --reconfigure build -Dbuildtype=release \
        -Dnative-arch-optimization=true \
        -Dfb=true \
        -Dharfbuzz=true \
        -Dlua-interpreter=lua \
        -Delua=true \
        -Dbindings=lua,cxx \
        -Devas-loaders-disabler=jxl \
        -Dglib=true \
        -Ddrm=true \
        -Dwl=true \
        -Dopengl=es-egl \
        -Dbuild-tests=false
      ninja -C build || mng_err
    fi
    ;;
  enlightenment)
    sudo chown $USER build/.ninja*
    meson setup --reconfigure build -Dbuildtype=release \
      -Dwl=true
    ninja -C build || mng_err
    ;;
  edi)
    sudo chown $USER build/.ninja*
    meson setup --reconfigure build -Dbuildtype=release \
      -Dlibclang-headerdir=/usr/lib/llvm-11/include \
      -Dlibclang-libdir=/usr/lib/llvm-11/lib
    ninja -C build
    ;;
  eflete)
    sudo chown $USER build/.ninja*
    meson setup --reconfigure build -Dbuildtype=release \
      -Denable-audio=true -Dwerror=false
    ninja -C build
    ;;
  *)
    sudo chown $USER build/.ninja*
    meson setup --reconfigure build -Dbuildtype=release
    ninja -C build
    ;;
  esac
```

```
    beep_attention
    $SNIN
    sudo ldconfig
  done
}

do_tests() {
  if [ -x /usr/bin/wmctrl ]; then
    if [ "$XDG_SESSION_TYPE" == "x11" ]; then
      wmctrl -r :ACTIVE: -b add,maximized_vert,maximized_horz
    fi
  fi

  printf "\n\n$BOLD%s $OFF%s\n" "System check..."

  if systemd-detect-virt -q --container; then
    printf "\n$BLDR%s %s\n" "ELUCIDATE IS NOT INTENDED FOR USE INSIDE CONTAINERS."
    printf "$BLDR%s $OFF%s\n\n" "SCRIPT ABORTED."
    beep_exit
    exit 1
  fi

  if [ $DISTRO == noble ]; then
    printf "\n$BLDG%s $OFF%s\n\n" "Ubuntu ${DISTRO^}... OK"
    sleep 1
  else
    printf "\n$BLDR%s $OFF%s\n\n" "UNSUPPORTED OPERATING SYSTEM [ $(lsb_release -d | cut -f2) ]."
    beep_exit
    exit 1
  fi

  if ! git ls-remote http://git.enlightenment.org/enlightenment/efl.git HEAD &>/dev/null; then
    printf "\n$BLDR%s %s\n" "REMOTE HOST IS UNREACHABLE--TRY AGAIN LATER"
    printf "$BLDR%s $OFF%s\n\n" "OR CHECK YOUR INTERNET CONNECTION."
    beep_exit
    exit 1
  fi

  if ! test -d "$HOME/.local/bin"; then
    mkdir -p "$HOME/.local/bin"
  fi

  if ! test -d "$HOME/.cache/ebuilds"; then
    mkdir -p "$HOME/.cache/ebuilds"
  fi
}

do_bsh_alias() {
  if [ -f $HOME/.bash_aliases ]; then
    mv -vb $HOME/.bash_aliases $HOME/.bash_aliases_bak
    echo
    touch $HOME/.bash_aliases
  else
    touch $HOME/.bash_aliases
  fi

  cat >$HOME/.bash_aliases <<EOF
    # ---------------------
    # ENVIRONMENT VARIABLES
    # ---------------------
    # (These variables can be accessed from any shell sessions.)

    # Compiler and linker flags added by ELUCIDATE.
    export CC="ccache gcc"
    export CXX="ccache g++"
    export USE_CCACHE=1
    export CCACHE_COMPRESS=1
    export CPPFLAGS=-I/usr/local/include
    export LDFLAGS=-L/usr/local/lib
    export PKG_CONFIG_PATH=/usr/local/lib/x86_64-linux-gnu/pkgconfig:/usr/local/lib/pkgconfig
```

```
EOF

  source $HOME/.bash_aliases
}

set_p_src() {
  echo
  beep_attention

  # Do not append a trailing slash (/) to the end of the path prefix,
  # and double-check the path you entered before validating.
  #
  read -p "Please enter a path for the Enlightenment source folders \
  (e.g. /home/$LOGNAME/Documents or /home/$LOGNAME/testing): " mypath
  mkdir -p "$mypath"/sources
  SRCDIR="$mypath"/sources
  echo $SRCDIR >$HOME/.cache/ebuilds/storepath
  printf "\n%s\n\n" "You have chosen: $SRCDIR"
  sleep 2
}

# Fetch and install prerequisites.
get_preq() {
  ESRCDIR=$(cat $HOME/.cache/ebuilds/storepath)
  printf "\n\n$BOLD%s $OFF%s\n\n" "Installing prerequisites..."

  cd $DLDIR
  wget https://github.com/rockowitz/ddcutil/archive/refs/tags/v$DDCTL.tar.gz

  tar xzvf v$DDCTL.tar.gz -C $ESRCDIR
  cd $ESRCDIR/ddcutil-$DDCTL
  $AUTGN
  make
  $SMIL
  sudo ldconfig
  rm -rf $DLDIR/v$DDCTL.tar.gz
  echo

  cd $ESRCDIR
  git clone https://github.com/Samsung/rlottie.git
  cd $ESRCDIR/rlottie
  meson setup build -Dbuildtype=plain \
    -Dexample=false
  ninja -C build
  $SNIN
  sudo ldconfig
  echo
}

do_link() {
  sudo ln -sf /usr/local/etc/enlightenment/sysactions.conf /etc/enlightenment/sysactions.conf
  sudo ln -sf /usr/local/etc/enlightenment/system.conf /etc/enlightenment/system.conf
  sudo ln -sf /usr/local/etc/xdg/menus/e-applications.menu /etc/xdg/menus/e-applications.menu
}

install_now() {
  clear
  printf "\n$BLDG%s $OFF%s\n\n" "* INSTALLING ENLIGHTENMENT DESKTOP ENVIRONMENT: PLAIN BUILD ON XORG SERVER *
  do_bsh_alias
  beep_attention
  bin_deps
  set_p_src
  get_preq

  cd $HOME
  mkdir -p $ESRCDIR/enlighten
  cd $ESRCDIR/enlighten

  printf "\n\n$BOLD%s $OFF%s\n\n" "Fetching source code from the Enlightenment git repositories..."
  $CLONEFL
  echo
```

```
  $CLONETY
  echo
  $CLONENL
  echo
  $CLONEPH
  echo
  $CLONERG
  echo
  $CLONEVI
  echo
  $CLONEXP
  echo
  $CLONECR
  echo
  $CLONEVE
  echo
  $CLONEDI
  echo
  $CLONENT
  echo
  $CLONEFT
  echo
  $CLONEPN
  printf "\n\n$BOLD%s $OFF%s\n\n" "Fetching source code from Dimmus' git repository..."
  $CLONETE
  echo

  cnt_dir
  build_plain

  sudo mkdir -p /etc/enlightenment
  do_link

  sudo ln -sf /usr/local/share/xsessions/enlightenment.desktop \
    /usr/share/xsessions/enlightenment.desktop

  # Protect this file from accidental deletion.
  sudo chattr +i $HOME/.cache/ebuilds/storepath

  printf "\n%s\n\n" "All done!"
  beep_ok

  printf "\n\n$BRTC%s %s" "INITIAL SETUP WIZARD TIPS:"
  printf "\n$BRTC%s %s" '“Update checking” -- you can disable this feature because it serves no useful purpos
  printf "\n$BRTC%s $OFF%s\n\n" '“Network management support” -- Connman is not needed (ignore the warning me

  # Note: Enlightenment adds three shortcut icons (namely home.desktop, root.desktop and tmp.desktop)
  # to your Gnome Desktop, you can safely delete them if it bothers you.

  echo
  cowsay "Now log out of your existing session then select Enlightenment on the login screen... \
  That's All Folks!" | lolcat -a
  echo

  cp -f $DLDIR/elucidate.sh $HOME/.local/bin

  exit 0
}

release_go() {
  clear
  printf "\n$BLDP%s $OFF%s\n\n" "* UPDATING ENLIGHTENMENT DESKTOP ENVIRONMENT: RELEASE BUILD ON XORG SERVER *

  # Check for available updates of the script folder first.
  cd $SCRFLDR && git pull &>/dev/null
  cp -f elucidate.sh $HOME/.local/bin
  chmod +x $HOME/.local/bin/elucidate.sh
  sleep 1

  rebuild_optim
```

```
  sudo ln -sf /usr/local/share/xsessions/enlightenment.desktop \
    /usr/share/xsessions/enlightenment.desktop

  if [ -f /usr/share/wayland-sessions/enlightenment.desktop ]; then
    sudo rm -rf /usr/share/wayland-sessions/enlightenment.desktop
  fi

  if [ -f /usr/local/share/wayland-sessions/enlightenment.desktop ]; then
    sudo rm -rf /usr/local/share/wayland-sessions/enlightenment.desktop
  fi

  beep_ok
  rstrt_e
  echo
  cowsay -f www "That's All Folks!"
  echo

  exit 0
}

wayld_go() {
  clear
  printf "\n$BLDY%s $OFF%s\n\n" "* UPDATING ENLIGHTENMENT DESKTOP ENVIRONMENT: RELEASE BUILD ON WAYLAND *"

  cd $SCRFLDR && git pull &>/dev/null
  cp -f elucidate.sh $HOME/.local/bin
  chmod +x $HOME/.local/bin/elucidate.sh
  sleep 1

  rebuild_wayld

  sudo mkdir -p /usr/share/wayland-sessions
  sudo ln -sf /usr/local/share/wayland-sessions/enlightenment.desktop \
    /usr/share/wayland-sessions/enlightenment.desktop

  beep_ok

  if [ "$XDG_SESSION_TYPE" == "x11" ] || [ "$XDG_SESSION_TYPE" == "wayland" ]; then
    echo
    cowsay -f www "Now log out of your existing session and press Ctrl+Alt+F3 to switch to tty3, \
        then enter your credentials and type: enlightenment_start" | lolcat -a
    echo
    # Wait a few seconds for the Wayland session to start.
    # When you're done, type exit
    # Pressing Ctrl+Alt+F1 will bring you back to the login screen.
  else
    echo
    cowsay -f www "That's it. Now type: enlightenment_start"
    echo
    # If Enlightenment fails to start, relaunch the script and select option 2.
    # After the build is complete type exit, then go back to the login screen.
  fi

  exit 0
}

# Lo and behold (“bhd”)!
#
# Display the selection menu...
#
lo() {
  trap '{ printf "\n$BLDR%s $OFF%s\n\n" "KEYBOARD INTERRUPT."; exit 130; }' INT

  INPUT=0
  printf "\n$BOLD%s $OFF%s\n" "Please enter the number of your choice:"

  if [ ! -x /usr/local/bin/enlightenment_start ]; then
    menu_selec
  else
    selec_menu
  fi
```

```
}

# and get the user's choice.
bhd() {
  if [ $INPUT == 1 ]; then
    do_tests
    install_now
  elif [ $INPUT == 2 ]; then
    do_tests
    release_go
  elif [ $INPUT == 3 ]; then
    do_tests
    wayld_go
  else
    beep_exit
    exit 1
  fi
}

lo
bhd
```