

---

# **Docs: GNU/Linux Documentation**

***Release 0.1.0***

**Joon Ro**

**Oct 08, 2017**



---

## Contents

---

<b>1</b>	<b>System Settings</b>	<b>3</b>
<b>2</b>	<b>Packages Installation</b>	<b>7</b>
<b>3</b>	<b>KDE</b>	<b>11</b>
<b>4</b>	<b>Enthought Python Distribution (EPD) Installation</b>	<b>13</b>
<b>5</b>	<b>PyQT Installation</b>	<b>15</b>
<b>6</b>	<b>virtualenv and virtualenvwrapper</b>	<b>19</b>
<b>7</b>	<b>Files to Backup</b>	<b>23</b>
<b>8</b>	<b>Indices and tables</b>	<b>25</b>



Contents:



# CHAPTER 1

---

## System Settings

---

### Root home Settings

```
$ su
$ chsh
$ cd ~
$ for f in .vim .vimrc .zshrc .oh-my-zsh; do; ln -s /home/joon/$f .; done;
```

### Font Settings

Generic Fonts Settings (Monospace, Serif, etc)

openSUSE applies fonts in order following files defined in `/etc/fonts/conf.d/56-user.conf` and then `/etc/fonts/suse-post-user.conf`. By default, the Microsoft fonts are on the top of the lists. Hence, when you install Microsoft fonts, they will become generic fonts and they look ugly. Hence make sure you move those fonts (Times New Roman, Arial, Consolas, etc) lower than Linux fonts such as DejaVu-Sans.

Put your settings into `~/.config/fontconfig/fonts.conf` and then add the following to `/etc/fonts/conf.d/56-user.conf`:

```
<include ignore_missing="yes">~/.config/fontconfig/fonts.conf</include>
```

### ksuperkey

```
$ git clone https://github.com/hanschen/ksuperkey.git
$ cd ksuperkey
$ make

$ ksuperkey -e 'ISO_Level3_Shift=Escape;Alt_R=Escape;Super_L=Alt_L|F1'
```

## ssh Server Setting

```
$ su
```

Change the ssh server settings:

```
$ vim /etc/ssh/sshd_config
```

check the ssh server status:

```
$ sudo netstat --inet -lpn | grep sshd
```

restart ssh server:

```
$ sudo /etc/init.d/sshd restart
```

In YaST, open a firewall port:

```
Allowed Services
Advanced...
Add your port to TCP port
```

You have to install terminfo to fix Error opening terminal: cygwin. error:

```
zypper in terminfo
```

Also, if you get Error: Can't open display: error when you login, you need to manually export DISPLAY:

```
export DISPLAY=:0.0
```

## Unlock encrypted LVM in CLI

```
$ su -
$ cryptsetup luksOpen /dev/sda2 cheer # Unlocks the partition sda2 and names it cheer
$ vgscan # find out the name of lvm; system for openSUSE LVM
$ vgchange -a y system
$ lvscan
$ mkdir /media/home
$ mount /dev/system/home /media/home
```

When it is done,

```
$ umount /media/home
$ vgchange -a n system
$ cryptsetup luksClose cheer
```

## Enabling network in CLI

<http://www.cyberciti.biz/faq/novell-sles-suse-restart-stop-networking-services-networking/>



```
# /etc/init.d/network start
# /etc/init.d/network start eth0
```

## Disable Touchpad while typing

<http://forums.opensuse.org/archives/sls-archives/archives-suse-linux/archives-general-questions/383041-syndaemon.html>

Just add the following line to `/etc/rc.d/boot.local`:

```
syndaemon -i 1 -d -t -K
```

If you want to disable the touchpad completely:

```
synclient TouchpadOff=1
```



## CHAPTER 2

---

### Packages Installation

---

#### Add Repos

```
$ baseURL=http://download.opensuse.org/repositories  
$ verSUSE=12.3 && SUSE=openSUSE_"$verSUSE" && verKDE=410
```

#### Shell

```
$ zypper in zsh screen  
$ chsh  
$ /bin/zsh
```

#### Editors

```
$ zypper in emacs gvim kate
```

#### Devel

```
$ zypper ar -f -n devel:tools:scm "$baseURL"/devel:/tools:/scm/"$SUSE"/  
→devel:tools:scm  
$ zypper in libpng12-0 gsl-devel  
$ zypper in gcc gcc-c++ gcc-fortran git git-cola qgit
```

## KDE

```
$ zypper in kalarm kdegraphics-thumbnailers krename krusader ksshaskpass
$ zypper in plasma-theme-caledonia plasma-theme-glassified plasma-theme-nitrogenium
$ zypper in kde4-ROSA-icontheme
```

## Python

```
$ zypper ar -f -n devel:languages:python "$baseURL"/devel:/languages:/python/"$SUSE"/
↪devel:languages:python
$ zypper in python-mechanize python-pycallgraph python-rst2pdf
$ zypper in python-sphinx python-virtualenvwrapper RunSnakeRun winpdb
```

For spynner:

```
$ zypper in python-devel libxml2-devel libxslt-devel python-spyner
```

## Scientific Python

```
$ zypper in python-numpy-devel python-scipy IPython python-matplotlib python-pandas
$ zypper in python-Cython python-tables python-Bottleneck
```

## Utilities

```
$ zypper in autokey-common autokey-qt backintime-kde libnotify-tools unison qsynergy
$ zypper ar -f -n google-talkplugin http://dl.google.com/linux/talkplugin/rpm/stable/
↪x86_64 google-talkplugin
$ zypper in google-talkplugin
```

## Multimedia

```
$ zypper in comix transmission-qt umplayer youtube-dl
```

## Fonts & Hangul

```
$ zypper ar -f -n M17N:fonts "$baseURL"/M17N:/fonts/"$SUSE"/ M17N:fonts
$ zypper in fetchmsttfonts
$ zypper in make gcc gtk2-devel gtk2-immodule-xim gtk3-devel gtk3-immodule-xim
↪libhangul-devel
$ zypper in cantarell-fonts google-inconsolata-fonts kde-oxygen-fonts
$ zypper in nanum-fonts nanum-gothic-coding-fonts ubuntu-fonts
```

## LaTeX

```
$ zypper in kile lyx
```



### Restart Plasma Desktop

```
$ kbuildsyco4 && kquitapp plasma-desktop && kstart plasma-desktop
```

### Klipper shortcut settings

KDE 4.10.0-1.2.1

If you do it in Global Keyboard Shortcuts, it will be reset on logout.

[https://bugs.kde.org/show\\_bug.cgi?id=300532#c1](https://bugs.kde.org/show_bug.cgi?id=300532#c1)

- right click on system tray expander arrow
- select “System Tray Settings”
- in “Entries” list, change keyboard shortcut for Klipper





---

## Enthought Python Distribution (EPD) Installation

---

### Download

EPD repo login page: <https://www.enthought.com/accounts/login/?next=/repo/epd/>

Set the environmental variable:

```
$ PYTHON=$EPD
```

### Update Packages

### Manual

```
$ $PYTHON/bin/easy_install pip
```

### Using enpkg

```
$ $PYTHON/bin/enpkg matplotlib pandas scipy
```

### Using pip

```
$ $PYTHON/bin/pip install bottleneck
```

## Manual updates

### IPython

```
$ rm -rf $PYTHON/lib/python2.7/site-packages/IPython $PYTHON/lib/python2.7/site-  
→packages/ipython*  
$ rm -rf build  
$ $PYTHON/bin/python setup.py install
```

### Cython

```
$ rm -rf $PYTHON/lib/python2.7/site-packages/Cython* $PYTHON/lib/python2.7/site-  
→packages/cython.*  
$ rm -rf build  
$ $PYTHON/bin/python setup.py install
```

### CythonGSL

```
$ rm -rf build  
$ $PYTHON/bin/python setup.py install
```

### Statsmodels

```
$ rm -rf $PYTHON/lib/python2.7/site-packages/statsmodels*  
$ $PYTHON/bin/pip install patsy  
  
$ git clone git@github.com:joonro/statsmodels.git  
$ cd statsmodels  
$ $PYTHON/bin/python setup.py install
```

---

## PyQT Installation

---

Need to install sip and PyQt

Make sure you have `libQtWebKit-devel` package before you install PyQt to get the QtWebKit installed properly.

I still does not know how to fix the following error:

```
>>> from PyQt4.QtNetwork import QNetworkCookie, QNetworkAccessManager

ImportError: /home/joon/virtualenvs/epd/lib/python2.7/site-packages/PyQt4/QtNetwork.
↳so: undefined symbol: _ZTI14QHttpMultiPart
```

## Dependencies

- make
- gcc
- gcc-c++
- gcc-fortran
- libqt4-devel
- libqtwebkit-devel
- libQtNetworkManager-devel

```
$ sudo zypper in make gcc gcc-c++ gcc-fortran libqt4-devel libqtwebkit-devel_
↳libQtNetworkManager-devel
```

## Download

The link to the download pages for both sip and PyQt4 are at: <http://pyqt.sourceforge.net/Docs/PyQt4/installation.html>

---

**Note:** The installation fails if you do the following in the NTFS external drive such as a USB stick. Extract the source into a directory in ~ first.

---

## Set the environmental variable PYTHON

```
$ PYTHON=$EPD
$ PYTHON=$anaconda
```

## Sip

### Remove previous installation

```
$ rm $PYTHON/lib/python2.7/site-packages/sip*
```

### Configure

```
$ $PYTHON/bin/python configure.py
```

### make & make install

```
$ make -j $(expr $(nproc) + 1) && make install
```

## PyQT4

### Remove previous installation

```
$ sudo rm -rf $PYTHON/lib/python2.7/site-packages/PyQt4
```

### Configure

```
$ $PYTHON/bin/python configure.py
```

## make & make install

```
$ make -j $(expr $(nproc) + 1)
$ sudo make install
```



## CHAPTER 6

---

### virtualenv and virtualenvwrapper

---

#### **PATH**

```
export EPD=~/.Programs/epd-7.3-2-rh5-x86_64
export anaconda=~/.Programs/anaconda
export canopy=~/.Enthought/Canopy_64bit/User
```

#### **Install pip**

```
$ sudo zypper install python-pip
```

or

```
$ sudo easy_install pip
```

#### **Installation of virtualenv and virtualenvwrapper on GNU/Linux:**

```
$ sudo zypper install python-virtualenvwrapper
```

or

```
$ sudo pip install virtualenvwrapper
```

or

```
$ sudo easy_install python-virtualenvwrapper
```

## Wrapping Enthought Python Distribution

Note that since you will have multiple projects using EPD, it is not good to make this EPD virtualenv and install stuff which will be used in many different project on it. Install common packages to EPD directory, and make virtualenvs for each of your projects.

### Make virtualenvs: epd

```
$ mkvirtualenv epd --python=$EPD/bin/python --system-site-packages
$ mkvirtualenv canopy --python=$canopy/bin/python --system-site-packages
```

### Make virtualenvs: downloadable-content-game

```
$ mkvirtualenv dlc --python=$EPD/bin/python --system-site-packages
$ workon dlc && cd ~/Dropbox/downloadable-content-game/simulation &&
↪setvirtualenvproject $VIRTUAL_ENV $(pwd)
```

### Make virtualenvs: elimination-used-market-game

```
$ mkvirtualenv elimination-used-market-game --python=$EPD/bin/python --system-site-
↪packages
$ workon elimination-used-market-game && cd ~/Dropbox/elimination-used-market-game/
↪paper && setvirtualenvproject $VIRTUAL_ENV $(pwd)
```

### Make virtualenvs: variety-seeking-movie

```
$ mkvirtualenv variety-seeking-movie --python=$EPD/bin/python --system-site-packages
$ workon variety-seeking-movie && cd ~/Dropbox/variety-seeking-movie/datawork &&
↪setvirtualenvproject $VIRTUAL_ENV $(pwd)
```

### Make virtualenvs: sphinx

```
$ mkvirtualenv sphinx --system-site-packages
$ workon sphinx && cd ~/Dropbox/docs && setvirtualenvproject $VIRTUAL_ENV $(pwd)
```

### Add EPD's bin to the PATH of virtualenv:

```
$ vim $WORKON_HOME/dlc/bin/activate
```

Change the following:

```
_OLD_VIRTUAL_PATH="$PATH"
PATH="$VIRTUAL_ENV/bin:$PATH"
export PATH
```



To:

```
_OLD_VIRTUAL_PATH="$PATH"  
PATH="$VIRTUAL_ENV/bin:$EPD/bin:$PATH"  
export PATH
```

```
PATH="$VIRTUAL_ENV/bin:$canopy/bin:$canopy/../System/bin:$PATH"
```

## Show virtualenv's name on the name of Konsole tab automatically

Add the following to the end of \$WORKON\_HOME/postactivate:

```
qdbus $KONSOLE_DBUS_SERVICE $KONSOLE_DBUS_SESSION setTitle 1 `basename $VIRTUAL_ENV`
```

Back to the default (put the following line to the \$WORKON\_HOME/postdeactivate:

```
qdbus $KONSOLE_DBUS_SERVICE $KONSOLE_DBUS_SESSION setTitle 1 "Konsole: %d : %n"
```



---

### Files to Backup

---

#### System

cronjobs:

```
crontab -l
```

#### KDE

open with settings:

```
~/.local/share/applications/mimeapps.list
```



## CHAPTER 8

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`