Subscribe

Keith R. Bennett's Technical Blog

Search

About

Building A Great Ruby Development Environment and Desktop with Linux Mint 13 “Maya” Mate

By keithrbennett on November 9th, 2012

The purpose of this article is to provide for you a clear and simple guide to setting up a nice Linux environment for Ruby software development and more.

I’ve been using Linux as a development environment on and off for a decade. In recent years I’ve leaned towards Mac OS, partly because I’ve been very disappointed in the Linux desktops’ progress (or lack of it). Nevertheless, I use Linux on all my old PC laptops, and in VM’s on my Macs. Enter Linux Mint, version 13…

I really like the new Linux Mint 13 Mate distro and decided to install it on several systems. The desktop is simple, intuitive, and clean, and underneath it’s Ubuntu. Unlike the Ubuntu distro, however, Mint includes codecs that are needed for multimedia play. More information about multimedia software and the Mint installation itself is at http://www.howtoforge.com/the-perfect-desktop-linux-mint-13-maya. Besides functioning as a software development environment, another use for my Mint systems is to drive my HDTV with content from TV web sites, Hulu Plus, YouTube, Vimeo, etc. Unfortunately, Netflix streaming video does not work on Linux.

At some point I’d like to take the time to learn Chef and automate the process, but until then, I figured I’d at least document everything I did to reduce the time and effort with each new installation.

This article describes the development environment I settled on for now, and how to replicate it. It’s intended to enable you to get a high quality system up to speed as quickly as possible. A lot of my choices are subjective (e.g. `zsh` rather than `bash`), so feel free to skip or modify anything. I assume you have a minimal understanding of Linux, and I omit some detail that might be needed by Linux beginners. Where version numbers are embedded in file names, those versions may differ at the time of your installation, so modify the names accordingly.

Following is a step by step guide. Although I installed Linux Mint, most or all of these steps should work on standard Ubuntu distributions too.

Download the ISO

The first step is to download the ISO from http://www.linuxmint.com/download.php, and burn a DVD or save it somewhere.

Install the OS

Install the OS from the DVD or disk image.

Update the Installed Software

Update by clicking the shield icon on the lower right of the desktop. *Select All* and *Install the Updates* to update the updater, then do it again to install the updates themselves.

Install Extra Packages

Linux Mint is based on Ubuntu and therefore uses apt-get/aptitude/synaptic for software package management. I use the command line apt-get for simplicity. I install the following extra software:

- **ant** – for building Clojure and other Java based software
- **chromium-browser** – for an alternative to Firefox, this is the browser on which Google’s Chrome is based
- **curl** – for RVM installation and general use
- **fldiff** – graphical diff tool for files and directories
- **g++** – for compiling C++ source code
- **gedit** – a simple graphical editor
- **gftp** – excellent graphical app for ftp operations, can do sftp too
- **gitk** – graphical Git repo visualizer
- **gnome-alsamixer** – volume control; this enabled me to increase

maximum volume for tv
- **libreadline-dev** – for command line history editing
- **libyaml-0-2** – YAML support
- **MySQL, Postgres, SQLite** – plus supplementary software and Postgres admin app
- **ncftp** – an excellent full screen but text mode ftp client, can use this when logging into the system with ssh in a terminal
- **openssh-server** – for SSH access to this machine
- **parcellite** – multi-entry clipboard
- **rlwrap** – adds readline support, used for Clojure REPL
- **skype**
- **stopwatch** – a stopwatch/timer with lap field
- **vim-gnome** – for a VIM editor with graphical abilities, run as `gvim`
- **virtualbox** – for virtual machines
- **zsh** – my preferred shell

Here’s the command to install them:

```
sudo apt-get install \
     curl \
     zsh \
     gedit \
     ncftp \
     virtualbox \
     vim-gnome \
     openssh-server \
     mysql-server mysql-client \
     postgresql-9.1 postgresql-contrib postgresql
     sqlite3 libsqlite3-dev \
     g++ \
     libreadline-dev \
     skype \
     parcellite \
     stopwatch \
     gftp \
     gitk \
     gnome-alsamixer \
     chromium-browser \
     ant \
     libyaml-0-2 \
     rlwrap
```

Desktop and Panel Shortcuts

For each app you want (e.g. Firefox, Chromium and Terminal), find it in the main menu and right click on it to get the menu to make links on desktop and panel.

Change Default Shell to ZShell

Make sure zsh was installed successfully:

```
which zsh
```

This command should return `/usr/bin/zsh`; if it returns nothing, zsh was not installed.

Run the `chsh` command to change the shell, specifying `/usr/bin/zsh` as your desired shell.

Log out, then log in again.

Git Configuration

Configure git, replacing the dummy data in the example commands with your real name and email address:

```
git config --global user.name "First M. Last"
git config --global user.email "myaddress@domain.
git config -l | grep user  # to list git variable
```

Adobe Acrobat Reader

Go to the Acrobat Reader “Other Downloads” page at http://get.adobe.com/reader/otherversions/. Select Linux, your preferred language, and then the .deb file. Download it, then open it to install the software.

Postgres Configuration

I use Postgres because it’s a great open source data base, and so that I’m using the same data base as Heroku. The script below will initialize Postgres with what I need to run a sample Rails app. Provide values for the environment variables at the top of the code fragment below. (Of course, use whatever `create database` commands you need; you might need more, or none, or no production data base, or want a different naming convention, etc.)

```
# Fill in the appropriate values to the right of
POSTGRES_USER_PASSWORD=
USERNAME=
PASSWORD=
APPNAME=

DEVAPPNAME="$APPNAME"_dev
TESTAPPNAME="$APPNAME"_test
PRODAPPNAME="$APPNAME"_prod

PSQL_CMD=$(cat <<EOF

alter role postgres with password '$POSTGRES_USE
create role $USERNAME with password '$PASSWORD';
alter role $USERNAME with login;
create database $DEVAPPNAME with owner $USERNAME;
create database $TESTAPPNAME with owner $USERNAMI
create database $PRODAPPNAME with owner $USERNAMI
EOF
)


echo $PSQL_CMD | sudo -u postgres psql
```

Downcase Directory Names

I have an aversion to capitalized directory names, since I spend a lot of time on the command line and don’t really need or want the minimal readability improvement of capitalized names. I’m always downloading stuff, so I rename the `Downloads` folder to `downloads`. I also use the `documents` directory a lot, so I downcased that as well.

```
mkdir ~/downloads
mv ~/Downloads/* ~/downloads
mv ~/Documents ~/documents
```

Go to Firefox, select menu *Edit*, then *Preferences*, select *Save files to*, click the *Browse* button, then select the newly created `downloads` folder.

Then delete the `Downloads` directory:

```
rmdir ~/Downloads
```

Modify Terminal For RVM Usage

RVM uses shell magic to do its thing, and in order for it to work, the shell in which it is run needs to be a login shell. To accomplish this, do the following:

Run the Terminal application. Then, from the menu, select *Edit*, then *Profile Preferences*, then select the *Title and Command* tab and enable *Run command as a login shell.*

Install RVM, Rubies, and Gems

You might want to check the RVM web site for the most current installation information, but at the time of this writing (November 2012) the command below is the recommended way:

```
\curl -L https://get.rvm.io | bash -s stable --rul
```

This will install RVM and a current stable MRI Ruby. Below we’ll do some other things:

- install JRuby, Ruby implementation for the Java Virtual Machine, my user group presentation slides and audio here)
- install Rails for both MRI Ruby and JRuby
- create aliases 1.9 and jruby for easier typing
- make MRI Ruby 1.9 the default Ruby for new shells

Open a new terminal, or source the startup shell command (e.g. `./zshrc`). Then:

```
rvm alias create 1.9 ruby-1.9.3-p385
rvm --default 1.9
gem install rails

rvm install jruby
rvm alias create jruby jruby-1.7.2
rvm jruby
gem install rails
```

Java

Mint has an open source Java implementation, but I find the Oracle JDK’s to be more problem-free. If you want to install Oracle’s Java, download the JDK from:
http://www.oracle.com/technetwork/java/javase/downloads/index.html.
Then, install it in `/opt` and create a symbolic link to it named `current`.

```
sudo mkdir /opt/java
cd /opt/java
tar xzf jdk*tar.gz # replace w/real filespec of d

sudo ln -s /opt/java/jdk1.7.0_09 current
```

(Note: Due to recent security concerns with running Java in browsers, the following instructions should be avoided unless you really need it.)

For Java support in Firefox, create a symbolic link in the Firefox installation’s `plugins` directory to the appropriate library. On my system that would look like this:

```
sudo ln -s /opt/java/current/jre/lib/amd64/libnpj

# You can also install the symbolic link in your 
# mkdir -p ~/.mozilla/plugins
# ln -s /opt/java/current/jre/lib/amd64/libnpjp2.
```

To see if Java is working in your browser, you can view this page:
http://www.java.com/en/download/testjava.jsp.

Modify .zshrc to Specify the Path and Prompt

I use a `~/bin` directory for miscellaneous executables that I want to keep in my user space:

```
mkdir ~/bin
```

We’ll want to modify the PATH to contain this directory and the Java executable directory. Also, we define JAVA_HOME and JDK_HOME to be the Java software root; some software may look for these variables.

In addition, you’ll probably want to redefine the terminal prompt to at least display the current directory. I define a prompt (`PS1` variable) below, but this PS1 syntax may not work for bash, and you may want to customize it to your own taste. Mine displays the previous command’s return value (0 for success, nonzero for failure), the time, the host name (convenient to differentiate from other hosts to which you may be logged in with ssh, etc.), and the current directory.

Edit `~/.zshrc` to make the above changes:

```
export JAVA_HOME=/opt/java/current
export JDK_HOME=$JAVA_HOME

export PATH=~/bin:$JAVA_HOME/bin:$PATH

export PS1="
[%?] %T `hostname` %B%d%b
>"
```

SSH Keys

Create ssh keys, then upload them to anywhere you’ll need them (e.g. Github and Bitbucket). You can provide a passphrase when asked, but it’s not absolutely necessary.

```
ssh-keygen -t rsa
```

Heroku

Install Heroku Toolbelt from https://toolbelt.heroku.com. Then:

```
heroku login
heroku keys:add ~/.ssh/id_rsa.pub # upload ssh key
```

Vim Configuration

For vim, install the Janus extensions, which include NerdTree:

```
curl -Lo- https://bit.ly/janus-bootstrap | bash
```

VMWare Fusion

If you’re installing this as a VMWare virtual machine, download and install the VMWare Tools:

Go to the VMWare “Virtual Machine” menu, select “Install VMWare Tools”. This will download them to your Mac, and make them available to your VM as a logical mounted CD. Do this to install them:

```
cd /opt
sudo tar xzf /media/VMware\ Tools/*gz
cd vmware-tools-distrib/
sudo ./vmware-install.pl
```

Flash

Flash is already installed, but an upgrade is available. Check http://www.adobe.com/software/flash/about/ and, if necessary, follow the link to the download page. Both Chromium and Firefox use (via links) a `libflashplayer.so` in `/etc/alternatives`, which points to `/opt/mint-flashplugin-11/libflashplayer.so`, so download the new package and copy the .so file there.

Clojure

Download from http://clojure.org/downloads. Then…

```
cd /opt
sudo unzip clojure.zip # Replace clojure.zip with
sudo ln -s /opt/clojure-1.4.0 /opt/clojure
cd /opt
sudo chown -R `whoami`:`whoami` clojure # need to
cd clojure
ant  # build it
```

Android

Android developer tools can be downloaded from http://developer.android.com/sdk/index.html.

Erlang OTP

If you’d like to download Erlang, you can do it at https://www.erlang-solutions.com/downloads/download-erlang-otp. You’ll need to know which Ubuntu distribution you’re downloading for; Linux Mint 13 is Ubuntu 12.04, and 14 is 12.10.

Conclusion

I hope this has been helpful. If you have any questions, corrections, suggestions, etc., feel free to comment.

Categorized under: Uncategorized.

Tagged with: no tags.

4 Responses to “Building A Great Ruby Development Environment and Desktop with Linux Mint 13 “Maya” Mate”

ZPH says: December 24, 2012 at 1:52 am

Nice writeup!

A few tricks that I would add to it as someone frequently deploying my environment to new systems, use a dotfiles git repo to simplify the process :). I have all dotfiles, zshrc, aliases, vim configs, etc saved in a git repo folder (though not yet on Github). This makes it a matter of git clone REPO to setup a sane environment, paired with a few crude shell scripts to install basics & swap to ZSH.

Another one that I need to get better about is breaking my dotfiles into logical subunits. I’m good w/ my ZSHRC containing the basics and the remainder being ‘required/included’ from a zsh.d directory. The ZSH.D dir has nicely segregated files that pertain to aliases, git aliases, fasd aliases, extra functions, etc. Helps keep things logical and I need to work on that for my mess of a VIMRC =D.

Lastly, the RVM alias tool is very helpful. Thanks for reminding me about it.

-Cheers

Log in to Reply

keithrbennett says: December 24, 2012 at 11:05 am

Thanks for the tips. If you ever make any of it public (in blog article, on Github, etc.), feel free to stop by and give us links.

– Keith

Log in to Reply

December 24, 2012 at 1:54 am

ZPH says:

One more that jumps to mind, ssh-copy-id for copying your public key over to other machines and appending it to appropriate auth file. Makes public key security very simple 🙂

Log in to Reply

December 24, 2012 at 11:24 am

keithrbennett says:

Thanks for that tip too, Zander.

People, I recommend checking out Zander’s blog. It’s at http://www.civet.ws/.

Log in to Reply

Leave a Response

You must be logged in to post a comment.

← Intro to Functional Programming in Ruby

Copying (RVM) Data Between Hosts Using ssh, scp, and netcat →

Meta

- Register
- Log in
- Entries RSS
- Comments RSS

- WordPress.org

Search

Archives

- November 2015
- November 2013
- August 2013
- January 2013
- December 2012
- November 2012
- September 2012
- July 2012
- January 2012
- June 2011
- May 2011
- March 2010
- July 2009
- March 2009
- February 2009
- November 2008

Recent Activity

Posts **Comments**

- The Case for Nested Methods in Ruby
- Ruby’s inject/reduce and each_with_object
- in Your System Prompt
- Using Oracle in JRuby with Rails and Sequel
- Copying (RVM) Data Between Hosts Using ssh, scp, and netcat
- Building A Great Ruby Development Environment and Desktop with Linux Mint 13 “Maya” Mate
- Intro to Functional Programming in Ruby
- WordPress Administration with

Ruby

- Stealth Conditionals in Ruby
- Hello, Nailgun; Goodbye, JVM Startup Delays

Archives

Categories Tags Dates Authors

- Uncategorized