# Installing and Setting Up Linux Mint 15 (64-bit) Cinnamon in a VMware Virtual Machine

## Installation

* Downloaded ISO from <http://www.linuxmint.com/download.php>
* Create new VMware Virtual Machine and specify installation of OS from downloaded ISO
* Specify Guest Operating System as Linux – Ubuntu 64-bit (Linux Mint is a Ubuntu Derivative)
* Name Virtual Machine: Linux Mint 15 Cinnamon 64-bit
* Location: C:\VMware Virtual Machines\Linux Mint 15 Cinnamon 64-bit
* 100 GB Max disk size, store virtual disk as a single file
* Customize Hardware …
* 2 GB memory, 1 processor; 1 core
* Let system boot by “playing” the virtual machine
* Choose the Install Linux Mint Icon on Desktop – Double Click
  + English Language
  + Verify space and internet connection - Continue
  + Erase disk and install Linux Mint – Install Now
  + Chicago Time Zone - Continue
  + Keyboard: English (US) – English (US) - Continue
  + Identify
    - Computer Name: TbbLinuxMint15
    - Username: tbb etc.
    - Continue
  + Installation Complete 🡪 Restart Now 🡪 NO 🡪 Continue Testing
  + Return to user interface and choose Menu 🡪 Shutdown
  + “Please remove installation media and close the tray (if any) then press ENTER:
  + Now is when you can make the ISO no longer be a CD/DVD drive attached to the virtual computer (after it is shutdown)
  + Choose Player 🡪 Power 🡪 Shut Down Guest
  + Edit Virtual Machine Settings
  + CD/DVD (IDE) 🡪 Use Physical drive: 🡪 OK
  + Play virtual machine
* Log in

## Installing VMware Tools

* If VMware Tools “dialog” is showing at bottom of screen, then choose the “Install Tools” button and follow instructions.
* If not showing, choose Player 🡪 Manage 🡪 Install VMware Tools… from menus
  + Double click on VMwareTools…tar.gz file and extract contents into directory in your home directory (drag and drop into home directory)
  + In terminal window, cd into vmware-tools-distrib directory and execute vmware-install.pl
    - % sudo perl vmware-install.pl
    - Accept all default answers
  + Wait until all the printers from the Windows system are added and then
  + Reboot after installation is done
    - Menu 🡪 Quit 🡪Restart
  + Screen should now resize to fit and handle multiple monitors via VMware

## A Little Configuration and Updating

* Right click in the panel at the bottom of the screen, choose “Add Applets to Panel”, find Workspace Switcher and add it.
  + Use Ctr-Alt-Up Arrow to get to interactive Workspace Switcher
* Look for shield with “I” in it in lower right hand corner near date
  + Install updates by clicking on shield icon, entering password, and running Update Manager
  + Choose to install updates – this will take a while to download and install the updates – you may have to choose “install updates” twice and approve cascading updates (updates that are triggered by the updates you’ve said to go ahead and do)

## Setting up file sharing with Windows

* Shutdown Linux Mint in the VMware Player
* Open VMware Player, select virtual machine, and Edit virtual machine settings 🡪 Options 🡪 Shared Folders 🡪 Always enabled 🡪 Add
* Next 🡪 Browse 🡪 Select Host path: C:\ and Name WindowsC
* Next
* Enable this share (checked); Read-only (not Checked)
* Finish
* Repeat for E: Drive, Host path: E:\ and Name WindowsE
* OK
* Play Virtual Machine
* Shared folder should be at: /mnt/hgfs/

## Setting up Gedit

* Run gedit from command line
* Choose Edit 🡪 Preferences 🡪 Editor
* Tab width: 4
* Check Insert spaces instead of tabs
* Check Create backup copy of files before saving
* Close preferences
* Exit from gedit

## Creating symbolic links in home directory to shared directories

* cd /mnt/hgfs/WindowsE/projects/LinuxSetup
* . SetupSymLinkDirs.sh

## Setting up Java

* cd ~/projects/LinuxSetup
* . SetupJava.sh
* Enter password for su to root to install software
* Enter default answers

## Setting up account/login setup files

* cd ~/projects/LinuxSetup
* . SetupDotFiles.sh

## Setting up Eclipse

* . SetupEclipse.sh

## Setting up Emacs

* . SetupEmacs.sh

## Setting up Git

* . SetupGit.sh

## Installing LaTeX

* . SetupLaTeX.sh
* sudo apt-get install texlive-full
* <https://help.ubuntu.com/community/LaTeX>
* Getting to grips with LaTeX – <http://www.andy-roberts.net/misc/latex>

## Unedited below here

## Setting up Subversion and Accessing Software Projects

* % sudo apt-get install subversion
  + Pretty good tutorial at [www.jaredrichardson.net/articles/svn-cheat-sheet.html](http://www.jaredrichardson.net/articles/svn-cheat-sheet.html)
* Make sure everything is checked-in in the “old” Linux/Ubuntu
* Run backupToWindows command script from “old” Ubuntu
* ici
* Move thirdparty directory from shared folder to new Ubuntu
* Move projects directory from shared folder to new Ubuntu
* Remove projects subdirectories that are in repository
  + rm –rf ~/projects/bbassoftware
  + rm –rf ~/projects/wustlsoftware
* Checkout subversion repositories
  + cd ~/projects
  + svn checkout <https://biobehavior.svn.cvsdude.com/bbassoftware>
    - get password from KeyPass
  + svn checkout <https://biobehavior.svn.cvsdude.com/wustlsoftware>

### Some Notes

* What’s in the repository and what’s not
  + bbassoftware
    - Contains BBAS owned software
    - FromSubversion Repository – <https://biobehavior.svn.cvsdude.com/bbassoftware>
  + eclipse
    - Contains eclipse workspaces
    - Not in repository
  + examples
    - Contains example java code that might be useful for imitation/learning
    - Not in repository
  + oci
    - Contains Eclipse and code materials for OCI training activities
    - Not in repository
  + old
    - Contains “old” source code that probably won’t ever be used again, but I’m unwilling to delete as of yet
    - Not in repository
* Checking to see what changes you’ve made
  + svn diff
  + svn diff <file name>
  + svn diff <directory name>
* Checking in your changes
  + svn commit –m “commit comment”
  + svn commit –-message “commit comment” <file name>
  + svn commit <directory name>
* Changing the editor used to make commit messages if you don’t use -m
  + --editor-cmd
  + SVN\_EDITOR environment variable
  + editor-cmd runtime configuration option
  + VISUAL environment variable
  + EDITOR environment variable
  + Fallback
* Getting everyone else’s changes
  + svn update
  + svn update <file name>
  + svn update <directory name>
* Adding a new file
  + svn add <file name>
  + svn add <directory name>
* See a file’s history
  + svn log
  + svn log <file name>
  + svn log <directory name>

See SVN Book at:

<http://svnbook.red-bean.com/>

## Installing Latex for Eclipse (TeXlipse)

* <http://texlipse.sourceforge.net/manual/installation.html>
* Help 🡪 Install New Software
* Add…
* Name: TeXlipse  
  Location: <http://texlipse.sourceforge.net>
* OK
* Check TeXlipse and Pdf4Eclipse
* Next, Next, … Finish

## Setting up and testing VPN software

This section needs to be tested and rewritten at WUSTL

* sudo apt-get install vpnc
  + That’s the VPN Client software
* sudo apt-get install vinagre
* Test the following scripts
  + startNilVpn
    - May have to update password in ~/bin/NIL.conf file
* NOTE: The NIL VPN connection from the Ubuntu “system” only works when on the WUSTL campus. The NIL VPN connection from the other systems (i.e. Windows) works fine from anywhere I’ve tried it. I can start the VPN connection from the Windows system, then use it from the Ubuntu system.
* As of installation of Linux Mint 12, this now works from Linux Mint without starting the VPN from Windows
* Remote Desktop (Applications 🡪 Internet 🡪 Remote Desktop Viewer) to the following locations using SSH
  + Linux1.neuroimage.wustl.edu
    - Be sure to set user name to tbbrown
  + VNC and SSH
    - linux1.neuroimage.wustl.edu:59XX
    - cninds03.neuroimage.wustl.edu:59XX
    - ccpmac1.wustl.edu
  + sshToCCPMAC
  + sshxToCCPMAC
  + sshxToNil
  + stopNilVpn

## Setting up CDT in Eclipse

* See <http://max.berger.name/howto/cdt/cdt.jsp>
* Use hello world C++ project to get build to work
* Need to tell Eclipse to autosave on build
  1. Window 🡪 Preferences 🡪 General 🡪 Workspace 🡪 Save automatically before build