# Install CentOS 8 for WaveQ3D on VirtualBox

An outline of the steps needed to setup a WaveQ3D Linux development environment for the first time.

# Install CentOS 8 in VirtualBox 6.1

* Download **CentOS-8.2.2004-x86\_64-dvd1.iso** from x86\_64 link under CentOS Linux on <https://www.centos.org/download/>
  + The DVD1 ISO image contains the installer as well as a set of all packages that can be installed during an interactive installation. Also sometimes referred to as the "Binary DVD" or "Binary ISO image". This is the recommended download for most users.
* Download and install **VirtualBox 6.1.14** from <https://www.virtualbox.org/wiki/Downloads>
  + Start this application and install extensions
* Create a new Virtual Machine in Virtual Box.
  + Name = CentOS 8
  + Type = Linux
  + Version = Red Hat (64-bit)
  + Memory Size = 8192 MB
  + VDI Hard Drive = 64 GB dynamically allocated
* Update Virtual Box settings
  + General > Advanced > Shared clipboard = Bidirectional
  + System > Motherboard > Boot Order move Hard Disk to top of list, disable Floppy
  + System > Processor > Processors = 4
  + Display > Screen > Acceleration > Video Memory = 64 Mb, Graphics Controller=VMSVGA (the default), enable 3D Acceleration = On
  + Shared Folders (optional)
  + User Interface > Mini Toolbar = Show in Full-screen = On & Show at Top of Screen = On
* Start this virtual machine to install CentOS
  + Select startup disk = CentOS-8.2.2004-x86\_64-dvd1.iso
  + Wait for automatic boot or hit I <Enter> to start
  + Language = English
  + Software Selection = Workstation with Development, Graphical Administration, and System Tools
  + Installation Destination = Automatically configure partitioning
  + Network and Host Name > Host Name = [yourname.bbn.com](http://yourname.bbn.com), Enthernet = On
  + Hit button to Begin Installation
  + Set Root Password
  + User Creation = yourname, not an administrator
  + Installation starts automatically
  + Hit Reboot button when complete
* Wait for automatic boot or hit I <Enter> to start
  + Licensing information = I accept the license agreement
  + Hit FINISH CONFIGURATION button.

# Setup Gnome 3 Desktop

* Login as yourname
* Language = English
* Typing = English (US)
* Location Services = off
* Connect Your Online Accounts = Skip
* Press the "Start Using CentOS Linux" button
* Close “Getting Started” window using “x” button upper, right
* Update all packages using Activities > Software

# Setup Dynamic Kernel Module Support

* Without this, Virtual Box Guest Additions need to be re-installed any time there is a kernel update.
* Open a terminal window and use “su” command to get root access.
* dnf install epel-release
* dnf install dkms
* exit

# Setup Virtual Box Guest Additions

* In the Virtual Box menu bar, select Devices > Insert Guest Additions CD image …
* Select “Run” to automatically start Guest Additions installation.
* Wait for it to compile kernel additions, this may take a while
* Restart machine.  If VM screen can be resized, Guest Additions have been installed.
* Eject Guest Additions CD image

# User Account Tweaks

* Activities > Settings > Power > Power Savings > Never
* Fix icon size: gsettings set org.gnome.nautilus.icon-view default-zoom-level small from terminal window
* Add /usr/local/lib to LD\_LIBRARY\_PATH in ~/.bashrc
* Add [Interesting Bash Scripts](https://collab.bbn.com/confluence/display/OAMASS/Interesting+Bash+Scripts) to ~/.bashrc
* Open a terminal window and use “su” command to get root access.
  + Add user to vboxsf group for shared folders: # usermod -aG vboxsf yourname
  + Add user to wheel group for sudo: # usermod -aG wheel yourname
  + exit
* Restart machine for changes to take effect

# Add dnf packages for development tools

* Open a terminal window and use “su” command to get root access.
* dnf groupinstall debugging
* dnf install git-gui gitk (to install Git GUI tools)
* dnf install texlive doxygen graphviz (to use Latex and Doxygen with collaboration diagrams)
* dnf install cmake cmake-gui (platform independent build scripts)
* dnf install meld (graphical diff for Git)
* dnf config-manager --set-enabled PowerTools (needed for NetCDF)
* dnf install netcdf-cxx-devel netcdf-devel nco (to install NetCDF C 4.7 and C++ 4.2)
* exit

Setup /usr/local to compile source code packages

* Open a terminal window and use “su” command to get root access.
* cd /usr
* chmod a+w local
* exit

# Compile Boost from source

* Download boost\_1\_75\_0.tar.gz (or better) from [http://www.boost.org](http://www.boost.org/) to /home/yourname/Downloads
* Open a normal terminal window
* cd /usr/local
* tar xvf /home/yourname/Downloads/boost\_1\_74\_0.tar.gz
* cd boost\_1\_75\_0
* ./bootstrap.sh (to configure gcc./b2)
* ./b2 (to compile Boost libraries)
* sudo ./b2 install (to install boost libraries)
* You should see boost libraries in /usr/local/include and /usr/local/lib directories.

# Download USML from your BBN BitBucket account

* cd ~
* mkdir Projects
* cd Projects
* git clone <https://collab.bbn.com/bitbucket/scm/oam/usml.git>
* git clone [https://collab.bbn.com/bitbucket/scm/oam/usml\_usnavy.git](https://collab.bbn.com/bitbucket/scm/oam/usml.git)
* git clone [https://collab.bbn.com/bitbucket/scm/oam/usml\_private.git](https://collab.bbn.com/bitbucket/scm/oam/usml.git)

# Setup Eclipse 2020 Integrated Development Environment

* Download Eclipse 2020 installer (eclipse-inst-jre-linux64.tar.gz) from  [https://www.eclipse.org/downloads](https://www.eclipse.org/downloads/) and expand archive file.
* Run eclipse-inst from /home/yourname/Downloads/eclipse-inst-jre-linux64/eclipse-installer directory.
* Select **Eclipse IDE for Scientific Computing** toolkit. Create start menu entry and create desktop shortcut.
  + Installs into /usr/local/eclipse/parallel-2020-12; dies if you didn't make /usr/local writable above.
  + Launch when install complete.  Close Welcome window.
  + Optional: Install AnsiConsole from Eclipse Marketplace to respect ANSI escape sequences generated by Boost Test package.
* Update Preferences for USML
  + General >Editors > Text Editors: Insert spaces for tabs = on, displayed tab width = 4, show print margin
  + C/C++ > Code Style > Formatter > Import: /home/yourname/Projects/usml/config/USML\_Formatter.xml
  + C/C++ > Code Analysis: Turn on all Potential Programming Problems, Security Vulnerabilities, and Syntax and Semantic Errors.
* Optional: Perspective > Customize Perspective and turn off:
  + Eclipse User Storage
  + Terminal
  + C/C++ Element Creation
  + Launch
  + Debug
  + Editor Presentation

# Import CMake Project to Eclipse

* Create project using CMake GUI
  + Where is the source code: /home/yourname/Projects/usml
  + Where to build binaries: /home/yourname/Projects/usml\_eclipse (must be outside of source code
  + Configure: Eclipse CDT4 - Unix Makefiles
  + Generate
* Import project into Eclipse
  + Turn off Project > Build Automatically
  + File > Import > General > Existing Projects into Workspace
    - Root directory = /home/yourname/Projects
    - Projects: usml-Release@usml\_eclipse
  + Clean project and rebuild all.