* Configuring OpenStudio Build Environments

Contents

[Downloading Source Code 1](#_Toc362936902)

[Windows Configuration 2](#_Toc362936903)

[Universal 2](#_Toc362936904)

[Windows 7/XP, Visual Studio 2008 Express 3](#_Toc362936905)

[Windows 7/XP, Visual Studio 2008 Professional 3](#_Toc362936906)

[Windows 7, Visual Studio 2010 Professional 3](#_Toc362936907)

[Linux Configuration 5](#_Toc362936908)

[Ubuntu 12.04 (x86 and x64) 5](#_Toc362936909)

[Ubuntu 13.04 (x86 and x64) 6](#_Toc362936910)

[Fedora 19 (x86 and x64) 6](#_Toc362936911)

[RHEL 5 (x86 and x64) 7](#_Toc362936912)

[Red Mesa (RHEL x64) 10](#_Toc362936913)

[Mac Configuration 15](#_Toc362936914)

[Mac OS X 10.7.x 15](#_Toc362936915)

[Mac OS X 10.8.x 16](#_Toc362936916)

[Windows Build 19](#_Toc362936917)

[Linux/Mac OS X Build 19](#_Toc362936918)

# Downloading Source Code

1. [Create a GitHub account](https://github.com/signup/free) (if you don’t already have one).
2. You will now be able to clone the repository: <https://github.com/NREL/OpenStudio>
3. If you need help cloning the source code refer to the document [Using OpenStudio with Git and GitHub](https://github.com/NREL/OpenStudio/wiki/Using-OpenStudio-with-Git-and-GitHub) instructions.

**Archive Extraction**

[7-Zip](http://www.7-zip.org/download.html) is highly recommended (particularly for Windows XP users) to extract the Ruby, Boost, and Qt libraries to their destination directories.

# Windows Configuration

### Universal

Extract [Ruby](http://developer.nrel.gov/downloads/buildings/openstudio/src/ruby-1.8.6-msvc.zip) to C:\Ruby

(Version 1.8.6 MSVC tested and working)

* + Install [Boost](http://boostpro.com/download/boost_1_47_setup.exe) for the Visual C++ 9.0 compiler with the Multithreaded and Multithreaded Debug variants

(1.47.0 MSVC installer tested and working)

* Install [Qt Libraries](http://download.qt-project.org/official_releases/qt/4.8/4.8.5/qt-win-opensource-4.8.5-vs2008.exe)

(Latest 4.8.5 VS 2008 tested and working)

Extract [SWIG](http://sourceforge.net/projects/swig/files/swigwin/swigwin-2.0.10/swigwin-2.0.10.zip/download) to C:\swig\swigwin-2.0.10

(Latest 2.0.10 tested and working)

* Install EnergyPlus 8.0 ([32-bit](http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-win.exe)) ([64-bit](http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-win-64.exe)) using the password `SFNpXW88q`
  + Install [CMake](http://www.cmake.org/files/v2.8/cmake-2.8.11.2-win32-x86.exe) with the option of adding CMake to the system PATH for all users

(Latest 2.8.11.2 tested and working)

* + Install [OpenSSL](http://slproweb.com/download/Win32OpenSSL-1_0_1e.exe), ignoring the warning regarding Visual C++ 2008 Redistributables, with the option of copying OpenSSL DLLs to the OpenSSL binaries (/bin) directory

(Latest Win32 1.0.1e tested and working)

* + Append `C:\Ruby\bin;C:\Qt\4.8.5\bin;C:\swig\swigwin-2.0.10;C:\Program Files (x86)\boost\boost\_1\_47` to the System `Path` variable

***For Building Documentation***

* Install [Doxygen](http://ftp.stack.nl/pub/users/dimitri/doxygen-1.8.4-setup.exe)

(Latest 1.8.4 tested and working)

* Install [Graphviz](http://www.graphviz.org/pub/graphviz/stable/windows/graphviz-2.30.1.msi)

(Latest 2.30.1 tested and working)

***For Building C# Bindings***

* Install [Sandcastle](http://sandcastle.codeplex.com/releases/view/47665?DownloadId=128770)

(Latest 2.6.1062.1 tested and working) - XP Users: Requires .NET Framework 2.0 or higher, so if you get a warning, come back to this step after installing .NET Framework 4.0 as part of the Visual Studio installation steps

***For Using Radiance***

* Install [Radiance](https://openstudio.nrel.gov/accept/file/1043) using the installer

***For Using DAKOTA***

Extract [DAKOTA 5.3.1](http://dakota.sandia.gov/distributions/dakota/5.3.1/dakota-5.3.1-public-CYGWIN.i686.zip) to C:\dakota-5.3.1-public-CYGWIN.i686

***For Building Installer Packages***

* Install [NSIS](http://prdownloads.sourceforge.net/nsis/nsis-2.46-setup.exe?download)

(2.46 tested and working)

### Windows 7/XP, Visual Studio 2008 Express

***Visual Studio (order is important)***

* Install [Visual C++ 2008 Express SP1](http://go.microsoft.com/?linkid=7729279) (the Silverlight Runtime and SQL Server 2008 Express options are unnecessary and may be unchecked)
* Install [Windows SDK 6.1](http://www.microsoft.com/download/en/confirmation.aspx?id=11310)
* Install [.NET Framework 4](http://www.microsoft.com/download/en/confirmation.aspx?id=17851)
* Install [Windows SDK 7.1](http://www.microsoft.com/download/en/confirmation.aspx?id=8279), and launch the Windows SDK Configuration Tool to make 7.1 the current SDK (Start Menu->Microsoft Windows SDK v7.1->Visual Studio Registration-> Windows SDK Configuration Tool)

VERY IMPORTANT: If your SDK 7.1 installation fails, uninstall *all* 2010 Redistributables from Control Panel->Programs and Features

* Add `C:\Program Files\Microsoft SDKs\Windows\v7.1\Bin` to the System `Path` variable

***To Build CSHARP\_BINDINGS***

* Install [Visual C# 2008 Express SP1](http://go.microsoft.com/?linkid=7729278) (the Silverlight Runtime and SQL Server 2008 Express options are unnecessary and may be unchecked)

***Special Notes***

It is recommended that all important Windows Updates be installed.

### Windows 7/XP, Visual Studio 2008 Professional

***Visual Studio***

* Install Visual Studio 2008 Professional with the Complete option is recommended, or at least with Visual C++ and Visual C#
* Install [Windows SDK 6.1](http://www.microsoft.com/download/en/confirmation.aspx?id=11310)
* Install [.NET Framework 4](http://www.microsoft.com/download/en/confirmation.aspx?id=17851)
* Install [Windows SDK 7.1](http://www.microsoft.com/download/en/confirmation.aspx?id=8279), and launch the Windows SDK Configuration Tool to make 7.1 the current SDK (Start Menu->Microsoft Windows SDK v7.1->Visual Studio Registration-> Windows SDK Configuration Tool)
* Add `C:\Program Files\Microsoft SDKs\Windows\v7.1\Bin` to the System `Path` variable

***Special Notes***

Windows SDK 6.0A that comes with Visual Studio 2008 Professional is sufficient, but it is recommended that the Windows SDK 6.1, .NET Framework 4, Windows SDK 7.1, and all relevant Windows Updates be installed, with 7.1 set as the current SDK and added to the System Path.

### Windows 7, Visual Studio 2010 Professional

* Install Visual Studio 2010 Professional (Uncheck everything except Visual C++ and Visual C#

Follow Windows Universal Instructions apart from the following:

* + Install [Boost](http://boostpro.com/download/boost_1_47_setup.exe) for the Visual C++ 10.0 compiler with the Multithreaded and Multithreaded Debug variants

(1.47.0 MSVC installer tested and working)

* Install [Qt Libraries](http://download.qt-project.org/official_releases/qt/4.8/4.8.5/qt-win-opensource-4.8.5-vs2010.exe)

(Latest 4.8.5 VS 2010 tested and working)

* Add `C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\Bin` to the System `Path` variable

# Linux Configuration

### Ubuntu 12.04 (x86 and x64)

***All Dependencies (except EnergyPlus):***

sudo apt-get install dpkg-dev subversion cmake-curses-gui libqt4-dev libboost-all-dev ruby-dev ruby swig libxt-dev doxygen graphviz

***EnergyPlus***

Install EnergyPlus 8.0 ([32-bit](http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-lin.sh)) ([64-bit](http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-lin-64.sh)) using the password `Xc9XYtEBq`

#Change this depending on architecture

wget http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-lin-64.sh

sudo sh SetEPlusV800008-lin-64.sh

rm SetEPlusV800008-lin-64.sh

***CMake***

sudo apt-get install libncurses-dev

wget http://www.cmake.org/files/v2.8/cmake-2.8.11.2.tar.gz

tar -xzf cmake-2.8.11.2.tar.gz

rm cmake-2.8.11.2.tar.gz

cd cmake-2.8.11.2

./configure

make

sudo make install

cd ..  
rm -rf cmake-2.8.11.2

***For Using DAKOTA***

Install gfortran, ATLAS, and LAPACK

sudo apt-get install gfortran libatlas-base-dev liblapack-dev

Download and extract the DAKOTA 5.3.1 source

wget http://dakota.sandia.gov/distributions/dakota/5.3.1/dakota-5.3.1-public-src.tar.gz

tar -xzf dakota-5.3.1-public-src.tar.gz

rm dakota-5.3.1-public-src.tar.gz

cd dakota-5.3.1.src

mkdir build

cd build

ccmake ..

#Set CMAKE\_BUILD\_TYPE to ‘Release’

#Set CMAKE\_INSTALL\_PREFIX to `/usr/local/dakota-5.3.1`

#Ensure HAVE\_X\_GRAPHICS is set to OFF

export F77=gfortran

make

sudo make install

echo 'export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/usr/local/dakota-5.3.1/lib' >> ~/.bashrc

cd ../..

rm -rf dakota-5.3.1.src

### Ubuntu 13.04 (x86 and x64)

***All Dependencies (except EnergyPlus):***

sudo apt-get install dpkg-dev subversion cmake-curses-gui libqt4-dev libboost1.49-all-dev ruby1.8-dev ruby1.8 swig libssl-dev libxt-dev doxygen graphviz

sudo ln -fs /usr/bin/ruby1.8 /usr/bin/ruby

***EnergyPlus***

Install EnergyPlus 8.0 ([32-bit](http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-lin.sh)) ([64-bit](http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-lin-64.sh)) using the password `Xc9XYtEBq`

#Change this depending on architecture

wget http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-lin-64.sh

sudo sh SetEPlusV800008-lin-64.sh

rm SetEPlusV800008-lin-64.sh

***For Using DAKOTA***

Dakota 5.3.1 is untested with Ubuntu 13

### Fedora 19 (x86 and x64)

***All Dependencies (except EnergyPlus):***

sudo yum groupinstall development-libs development-tools

sudo yum install gcc-c++ cmake swig patch qt-devel qtwebkit-devel ruby ruby-devel graphviz

sudo yum remove boost-devel

curl -O http://downloads.sourceforge.net/project/boost/boost/1.47.0/boost\_1\_47\_0.tar.gz

tar -xzf boost\_1\_47\_0.tar.gz

rm boost\_1\_47\_0.tar.gz

cd boost\_1\_47\_0

# Apply patch <https://svn.boost.org/trac/boost/ticket/6165>

# Replace all instances of “TIME\_UTC” with “TIME\_UTC\_” in boost/thread/xtime.hpp and libs/thread/src/pthread/timeconv.inl

sh ./bootstrap.sh

./b2

sudo ./b2 install --prefix=/usr/local

cd ..

rm -rf boost\_1\_47\_0

***EnergyPlus***

Install EnergyPlus 8.0 ([32-bit](http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-lin.sh)) ([64-bit](http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-lin-64.sh)) using the password `Xc9XYtEBq`

#Change this depending on architecture

curl -O http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-lin-64.sh

sudo sh ./SetEPlusV800008-lin-64.sh

rm SetEPlusV800008-lin-64.sh

### RHEL 5 (x86 and x64)

sudo yum groupinstall development-libs development-tools

sudo yum remove boost-devel

##############

# Ruby 1.8.7 #

##############

sudo yum remove ruby ruby-devel

wget [http://ftp.ruby-lang.org/pub/ruby/1.8/ruby-1.8.7-p374.tar.gz](http://ftp.ruby-lang.org/pub/ruby/1.8/ruby-1.8.7-p358.tar.gz)

tar -xzf ruby-1.8.7-p374.tar.gz

rm ruby-1.8.7-p374.tar.gz

cd ruby-1.8.7-p374

./configure

make

sudo make install

cd ..

rm -rf ruby-1.8.7-p374

wget http://production.cf.rubygems.org/rubygems/rubygems-2.0.4.tgz

tar -xzf rubygems-2.0.4.tgz

rm rubygems-2.0.4.tgz

cd rubygems-2.0.4

sudo ruby setup.rb

cd ..

rm -rf rubygems-2.0.4

gem install rake

##############

# SWIG 2.0.10 #

##############

wget http://prdownloads.sourceforge.net/swig/swig-2.0.10.tar.gz

tar -xzf swig-2.0.10.tar.gz

rm swig-2.0.10.tar.gz

cd swig-2.0.10

sudo yum install pcre-devel

./configure

make

sudo make install

cd ..

rm -rf swig-2.0.10

################

# Boost 1.47.0 #

################

wget http://downloads.sourceforge.net/project/boost/boost/1.47.0/boost\_1\_47\_0.tar.gz

tar -xzf boost\_1\_47\_0.tar.gz

rm boost\_1\_47\_0.tar.gz

cd boost\_1\_47\_0

./bootstrap.sh

./bjam

sudo ./bjam install

cd ..

rm -rf boost\_1\_47\_0

############

# Qt 4.8.5 #

############

wget http://download.qt-project.org/official\_releases/qt/4.8/4.8.5/qt-everywhere-opensource-src-4.8.5.tar.gz

tar -xzf qt-everywhere-opensource-src-4.8.5.tar.gz

rm qt-everywhere-opensource-src-4.8.5.tar.gz

cd qt-everywhere-opensource-src-4.8.5

./configure -debug-and-release -opensource -qt-sql-sqlite -plugin-sql-sqlite -no-qt3support -nomake examples -nomake demos -nomake docs -confirm-license

gmake

***#Go through the Makefiles in the 3rd party directory and remove the -Werror compiler flags if build failed (This step may need to be repeated)***

#find src/3rdparty/webkit/ -type f -name Makefile -exec sed -i 's/-Werror //g' {} \;

#find src/3rdparty/webkit/ -type f -name Makefile.\* -exec sed -i 's/-Werror //g' {} \;

#gmake

sudo gmake install

cd ..

rm -rf qt-everywhere-opensource-src-4.8.5

###############

# CMake 2.8.11.2 #

###############

wget http://www.cmake.org/files/v2.8/cmake-2.8.11.2.tar.gz

tar -xzf cmake-2.8.11.2.tar.gz

rm cmake-2.8.11.2.tar.gz

cd cmake-2.8.11.2

./configure

gmake

sudo gmake install

cd ..  
rm -rf cmake-2.8.11.2

##############

# Dakota 5.3.1 #

##############

sudo yum -y install gcc-gfortran blas-devel lapack-devel

wget http://dakota.sandia.gov/distributions/dakota/5.3.1/dakota-5.3.1-public-src.tar.gz

tar -xzf dakota-5.3.1-public-src.tar.gz

rm dakota-5.3.1-public-src.tar.gz

cd dakota-5.3.1.src

mkdir build

cd build

ccmake ..

#Set CMAKE\_BUILD\_TYPE to ‘Release’

#Set CMAKE\_INSTALL\_PREFIX to `/usr/local/dakota-5.3.1`

#Ensure HAVE\_X\_GRAPHICS is set to OFF

export F77=gfortran

make

sudo make install

cd ../..

rm -rf dakota-5.3.1.src

***EnergyPlus***

RHEL will require a special EnergyPlus build for 8.0 due to glibc incompatibility

Install EnergyPlus 8.0 ([32-bit](http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-lin-RHEL5.sh)) ([64-bit](http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-lin-64-RHEL5.sh)) using the password `Xc9XYtEBq`

#Change this depending on architecture

curl -O http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-lin-64-RHEL5.sh

sudo sh ./SetEPlusV800008-lin-64-RHEL5.sh

rm SetEPlusV800008-lin-64-RHEL5.sh

### Red Mesa (RHEL x64)

***All-In-One:***

mkdir ~/scratch/deps

###############

# CMake 2.8.11.2 #

###############

cd ~/scratch/deps

wget http://www.cmake.org/files/v2.8/cmake-2.8.11.2.tar.gz

tar -xzf cmake-2.8.11.2.tar.gz

rm cmake-2.8.11.2.tar.gz

cd cmake-2.8.11.2

./configure --prefix=/projects/nrel/apps/eplus/deps/cmake

gmake

gmake install

cd ..

rm -rf cmake-2.8.11.2

##############

# SWIG 2.0.10 #

##############

cd ~/scratch/deps

wget http://prdownloads.sourceforge.net/swig/swig-2.0.10.tar.gz

tar -xzf swig-2.0.10.tar.gz

rm swig-2.0.10.tar.gz

cd swig-2.0.10

./configure --prefix=/projects/nrel/apps/eplus/deps/swig

make

make install

cd ..

rm swig-2.0.10

################

# EnergyPlus 8 #

################

cd ~/scratch/deps

wget http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-lin-64-RHEL5.sh

sh ./SetEPlusV800008-lin-64-RHEL5.sh

#/projects/nrel/apps/eplus/deps

#n

#Xc9XYtEBq

#Ctrl+C at sudo prompt

rm SetEPlusV800008-lin-64-RHEL5.sh

################

# Boost 1.47.0 #

################

cd ~/scratch/deps

wget http://downloads.sourceforge.net/project/boost/boost/1.47.0/boost\_1\_47\_0.tar.gz

tar -xzf boost\_1\_47\_0.tar.gz

rm boost\_1\_47\_0.tar.gz

cd boost\_1\_47\_0

./bootstrap.sh --prefix=/projects/nrel/apps/eplus/deps/boost

srun -p inter --time=3:00:00 --account=BTP000 ./bjam --prefix=/projects/nrel/apps/eplus/deps/boost

./bjam install --prefix=/projects/nrel/apps/eplus/deps/boost

cd ..

rm -rf boost\_1\_47\_0

############

# Qt 4.8.5 #

############

cd ~/scratch/deps

wget http://download.qt-project.org/official\_releases/qt/4.8/4.8.5/qt-everywhere-opensource-src-4.8.5.tar.gz

tar -xzf qt-everywhere-opensource-src-4.8.5.tar.gz

rm qt-everywhere-opensource-src-4.8.5.tar.gz

cd qt-everywhere-opensource-src-4.8.5

srun -p inter --time=1:00:00 --account=BTP000 ./configure -release -opensource -qt-sql-sqlite -plugin-sql-sqlite -no-qt3support -nomake examples -nomake demos -nomake docs -confirm-license -prefix /projects/nrel/apps/eplus/deps/qt

srun -p inter --time=3:00:00 --account=BTP000 gmake -j8

cp -r \* /projects/nrel/apps/eplus/deps/qt

cd ..

rm -rf qt-everywhere-opensource-src-4.8.5

##############

# Ruby 1.8.7 #

##############

cd ~/scratch/deps

wget [http://ftp.ruby-lang.org/pub/ruby/1.8/ruby-1.8.7-p374.tar.gz](http://ftp.ruby-lang.org/pub/ruby/1.8/ruby-1.8.7-p358.tar.gz)

tar -xzf ruby-1.8.7-p374.tar.gz

rm ruby-1.8.7-p374.tar.gz

cd ruby-1.8.7-p374

./configure --prefix=/projects/nrel/apps/eplus/deps/ruby --disable-pthread

srun -p inter --time=1:00:00 --account=BTP000 make -j8

make install

cd ..

rm -rf ruby-1.8.7-p374

Set .bash\_profile at this step to ensure that the ruby path is correct

##############

# Rake 0.8.7 #

##############

cd ~/scratch/deps

wget http://rubyforge.org/frs/download.php/56872/rake-0.8.7.tgz

tar -xzf rake-0.8.7.tgz

rm rake-0.8.7.tgz

cd rake-0.8.7

ruby install.rb

cd ..

rm -rf rake-0.8.7

##############

# Dakota 5.3.1 #

##############

cd ~/scratch/deps

wget http://dakota.sandia.gov/distributions/dakota/5.3.1/dakota-5.3.1-public-src.tar.gz

tar -xzf dakota-5.3.1-public-src.tar.gz

rm dakota-5.3.1-public-src.tar.gz

cd dakota-5.3.1.src

mkdir build

cd build

ccmake ..

#Set CMAKE\_BUILD\_TYPE to ‘Release’

#Set CMAKE\_INSTALL\_PREFIX to /projects/nrel/apps/eplus/deps/dakota-5.3.1`

#Ensure HAVE\_X\_GRAPHICS is set to OFF

srun -p inter --time=1:00:00 --account=BTP000 make -j8

make install

cd ../..

rm -rf dakota-5.3.1.src

#########

# D-Bus #

#########

# This may be necessary to use the BCL

cd ~/scratch/deps

wget http://dbus.freedesktop.org/releases/dbus/dbus-1.7.4.tar.gz

tar -xzf dbus-1.7.4.tar.gz

rm dbus-1.7.4.tar.gz

cd dbus-1.7.4

./configure --prefix=$HOME/scratch/deps/dbus-1.7.4/build

make

make install

cp build/lib/libdbus-1.so.3.7.2 /projects/nrel/apps/eplus/deps/qt/lib

ln -s /projects/nrel/apps/eplus/deps/qt/lib/libdbus-1.so.3.7.2 /projects/nrel/apps/eplus/deps/qt/lib/libdbus-1.so

ln -s /projects/nrel/apps/eplus/deps/qt/lib/libdbus-1.so.3.7.2 /projects/nrel/apps/eplus/deps/qt/lib/libdbus-1.so.3

cd ..

rm -rf dbus-1.7.4

##########################

# OpenStudio v1.0.0 #

##########################

cd /projects/nrel/apps/eplus/

git clone -b v1.0.1 git@github.com:NREL/OpenStudio.git openstudio-1.0.0

cd openstudio-1.0.0

mkdir build

cd build

ccmake ../

#Configure & Generate with Package, Testing, OpenSSL, and PCH on, in Release mode, with install prefix erased

srun -p inter --time=3:00:00 -c 8 --account=BTP000 make package -j8

Note: If building OpenStudio <v0.6.3, replace find\_energyplus.rb with the latest version from ./openstudiocore/ruby/openstudio/energyplus/find\_energyplus.rb

.bash\_profile:

EPLUS=/projects/nrel/apps/eplus

export RUBYLIB=$EPLUS/openstudio/Ruby:$EPLUS/openstudio/share/openstudio/Ruby:$RUBYLIB

export PATH=$EPLUS/deps/cmake/bin:$EPLUS/deps/boost:$EPLUS/deps/qt/bin:$EPLUS/deps/swig/bin:$EPLUS/deps/EnergyPlus-8-0-0/bin:$EPLUS/deps/ruby/bin:$PATH

export LD\_LIBRARY\_PATH=$EPLUS/openstudio/lib:$EPLUS/deps/boost/lib:$EPLUS/deps/qt/lib:$EPLUS/deps/dakota/lib:$LD\_LIBRARY\_PATH

export ENERGYPLUSDIR=$EPLUS/deps/EnergyPlus-8-0-0

export Boost\_DIR=/projects/nrel/apps/eplus/deps/boost

export DAKOTADIR=/projects/nrel/apps/eplus/deps/Dakota

umask 0007

# Mac Configuration

### Mac OS X 10.7.x

Install EnergyPlus 8.0 ([64-bit](http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-mac-64.dmg)) using the password `q3NDwCqrU`

Install [CMake](http://www.cmake.org/files/v2.8/cmake-2.8.11.2-Darwin64-universal.dmg) with the option of creating symbolic links in /usr/bin

Install Xcode from the App Store

#Xcode->Preferences->Downloads->Components->Command Line Tools

sudo /usr/bin/xcode-select -switch /Applications/Xcode.app/Contents/Developer

#Xcode->Open Developer Tool->More Developer Tools…

#Download the latest Auxiliary Tools for Xcode

#Drag PackageMaker.app to Applications or to the Xcode package Applications directory

Extract, configure, and build [Boost](http://downloads.sourceforge.net/project/boost/boost/1.47.0/boost_1_47_0.tar.gz)

curl -O http://downloads.sourceforge.net/project/boost/boost/1.47.0/boost\_1\_47\_0.tar.gz

tar -xzf boost\_1\_47\_0.tar.gz

rm boost\_1\_47\_0.tar.gz

cd boost\_1\_47\_0

#Apply patch: https://svn.boost.org/trac/boost/attachment/ticket/6686/xcode\_43.diff

sh ./bootstrap.sh

./b2 variant=release address-model=32\_64 architecture=x86 debug-symbols=on pch=off macosx-version=10.7 threading=multi install --prefix=build -j2

cd build

sudo cp -r \* /usr/local/

cd ../..

rm -rf boost\_1\_47\_0

Extract, configure, and build [Qt](http://download.qt-project.org/official_releases/qt/4.8/4.8.5/qt-everywhere-opensource-src-4.8.5.tar.gz)

curl -O http://download.qt-project.org/official\_releases/qt/4.8/4.8.5/qt-everywhere-opensource-src-4.8.5.tar.gz

tar -xzf qt-everywhere-opensource-src-4.8.5.tar.gz

rm qt-everywhere-opensource-src-4.8.5.tar.gz

cd qt-everywhere-opensource-src-4.8.5

./configure -sdk /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.7.sdk -debug-and-release -opensource -arch x86 -arch x86\_64 -qt-sql-sqlite -plugin-sql-sqlite -nomake examples -nomake demos -nomake docs -no-qt3support -confirm-license

make

sudo make install

cd ..

rm -rf qt-everywhere-opensource-src-4.8.5

#sudo nano /etc/paths

##Add "/usr/local/Trolltech/Qt-4.8.5/bin"

Extract, configure, build [SWIG](http://prdownloads.sourceforge.net/swig/swig-2.0.10.tar.gz) with [PCRE](ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.33.tar.gz) in /usr/bin.

curl -O ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.33.tar.gz

tar -xzf pcre-8.33.tar.gz

rm pcre-8.33.tar.gz

cd pcre-8.33

./configure

make

sudo make install

cd ..

rm -rf pcre-8.33

curl -O http://prdownloads.sourceforge.net/swig/swig-2.0.10.tar.gz

tar -xzf swig-2.0.10.tar.gz

rm swig-2.0.10.tar.gz

cd swig-2.0.10

./configure

make

sudo make install

cd ..

rm -rf swig-2.0.10

***For Building Documentation***

Download [Doxygen](http://ftp.stack.nl/pub/users/dimitri/Doxygen-1.8.4.dmg) and drag it to Applications

Install Graphviz ([Lion](http://www.graphviz.org/pub/graphviz/stable/macos/lion/graphviz-2.30.1.pkg))

### Mac OS X 10.8.x

Install EnergyPlus 8.0 ([64-bit](http://apps1.eere.energy.gov/buildings/energyplus/download/SetEPlusV800008-mac-64.dmg)) using the password `q3NDwCqrU`

Install [CMake](http://www.cmake.org/files/v2.8/cmake-2.8.11.2-Darwin64-universal.dmg) with the option of creating symbolic links

Install Xcode from the App Store

#Xcode->Preferences->Downloads->Components->Command Line Tools

sudo /usr/bin/xcode-select -switch /Applications/Xcode.app/Contents/Developer

#Xcode->Open Developer Tool->More Developer Tools…

#Download the latest Auxiliary Tools for Xcode

#Drag PackageMaker.app to Applications or to the Xcode package Applications directory

Extract, configure, and build [Boost](http://downloads.sourceforge.net/project/boost/boost/1.47.0/boost_1_47_0.tar.gz)

curl -O http://downloads.sourceforge.net/project/boost/boost/1.47.0/boost\_1\_47\_0.tar.gz

tar -xzf boost\_1\_47\_0.tar.gz

rm boost\_1\_47\_0.tar.gz

cd boost\_1\_47\_0

#Apply patch: https://svn.boost.org/trac/boost/attachment/ticket/6686/xcode\_43.diff

sh ./bootstrap.sh

./b2 variant=release variant=debug address-model=32\_64 architecture=x86 --layout=tagged macosx-version=10.7 --without-python --without-math install --prefix=build -j2

cd build

sudo cp -r \* /usr/local/

cd ../..

rm -rf boost\_1\_47\_0

Extract, configure, and build [Qt](http://download.qt-project.org/official_releases/qt/4.8/4.8.5/qt-everywhere-opensource-src-4.8.5.tar.gz)

curl -O http://download.qt-project.org/official\_releases/qt/4.8/4.8.5/qt-everywhere-opensource-src-4.8.5.tar.gz

tar -xzf qt-everywhere-opensource-src-4.8.5.tar.gz

rm qt-everywhere-opensource-src-4.8.5.tar.gz

cd qt-everywhere-opensource-src-4.8.5

cd src/3rdparty/webkit/WebKitLibraries

mv libWebKitSystemInterfaceMountainLion.a libWebKitSystemInterfaceMountainLion.a.bak

cp libWebKitSystemInterfaceLion.a libWebKitSystemInterfaceMountainLion.a

cd ../../../..

./configure -sdk /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.7.sdk -debug-and-release -opensource -arch x86 -arch x86\_64 -qt-sql-sqlite -plugin-sql-sqlite -nomake examples -nomake demos -nomake docs -no-qt3support -confirm-license

make

sudo make install

cd ..

rm -rf qt-everywhere-opensource-src-4.8.5

#sudo nano /etc/paths

##Add "/usr/local/Trolltech/Qt-4.8.5/bin"

Extract, configure, build [SWIG](http://prdownloads.sourceforge.net/swig/swig-2.0.10.tar.gz) with [PCRE](ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.33.tar.gz) in /usr/bin.

curl -O ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.33.tar.gz

tar -xzf pcre-8.33.tar.gz

rm pcre-8.33.tar.gz

cd pcre-8.33

./configure

make

sudo make install

cd ..

rm -rf pcre-8.33

curl -O http://prdownloads.sourceforge.net/swig/swig-2.0.10.tar.gz

tar -xzf swig-2.0.10.tar.gz

rm swig-2.0.10.tar.gz

cd swig-2.0.10

./configure

make

sudo make install

cd ..

rm -rf swig-2.0.10

***For Building Documentation***

Download [Doxygen](http://ftp.stack.nl/pub/users/dimitri/Doxygen-1.8.4.dmg) and drag it to Applications

Install Graphviz ([Mountain Lion](http://www.graphviz.org/pub/graphviz/stable/macos/mountainlion/graphviz-2.30.1.pkg))

# Windows Build

1. Create a directory to store the source
2. Clone the OpenStudio Git repository.
3. Launch CMake-GUI.
4. Under "Where is the source code:” browse to the OpenStudio checkout on your computer (e.g., `C:\projects\OpenStudio`).
5. Under "Where to build the binaries:" select a build directory (e.g., `C:\projects\OpenStudio\build`).
6. Click the configure button.
7. Select Visual Studio 9 2008 when prompted.
8. After it configures, some items in the main box will be in red. This includes the BUILD\_ options, such as BUILD\_DOCUMENTATION, and BUILD\_TESTING. Choose your options.
9. Click the configure button again.
10. If you are using Visual Studio Express Edition, check the MSVC\_IS\_EXPRESS option if it becomes available.
11. Repeat until there are no more red items (More options are available in Advanced mode).
12. Finally, click the generate button.

The Visual Studio Solution file (OpenStudio.sln) will be in the Build Directory. Open with Visual Studio and build your desired targets.

# Linux/Mac OS X Build

1. Clone the OpenStudio Git repository.

git clone -b iteration git@github.com:NREL/OpenStudio.git OpenStudio

1. Create a build directory

cd OpenStudio; mkdir build; cd build

1. Launch CMake

ccmake ..

1. Press `c` to configure, and change the configuration as necessary.
2. Repeat until the generate option becomes available (More options are available in Advanced mode, `t`).
   1. For Xcode 4.3 or later, set CMAKE\_OSX\_ARCHITECTURES to `i386;x86\_64`, set CMAKE\_OSX\_DEPLOYMENT\_TARGET to `10.7`, and set CMAKE\_OSX\_SYSROOT to `/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.7.sdk`
   2. For older versions of Xcode, set CMAKE\_OSX\_ARCHITECTURES to `i386;x86\_64`, set CMAKE\_OSX\_DEPLOYMENT\_TARGET to `10.6`, and set CMAKE\_OSX\_SYSROOT to `/Developer/SDKs/MacOSX10.6.sdk`
3. Press `g` to generate the makefiles in the build directory.
4. To build the complete OpenStudio project, simply type `**make**`. If you have a multi-core machine, you can build in parallel using the -j option. For example, `make -j8`, to run make with up to 8 threads.