How to Build/Debug GridLAB-D for Windows and Linux

# Windows OS:

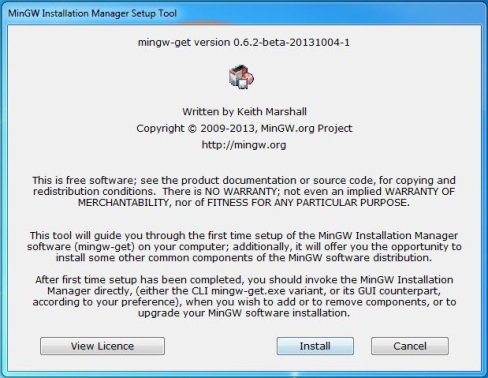
Needed Software:

* Trunk of GridLAB-D.
* Eclipse.
* MinGW.
* MSYS.
* Win-Builds.
* DLFCN(third party software provided in the ticket)
* Xerces (third party software provided in the ticket)

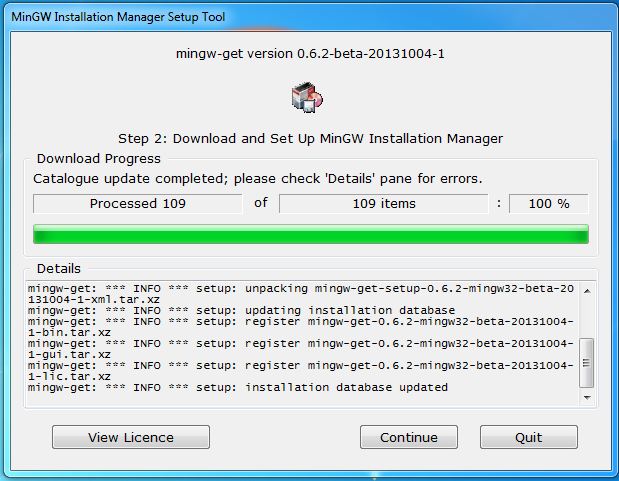
# Installation:

MinGW and MSYS:

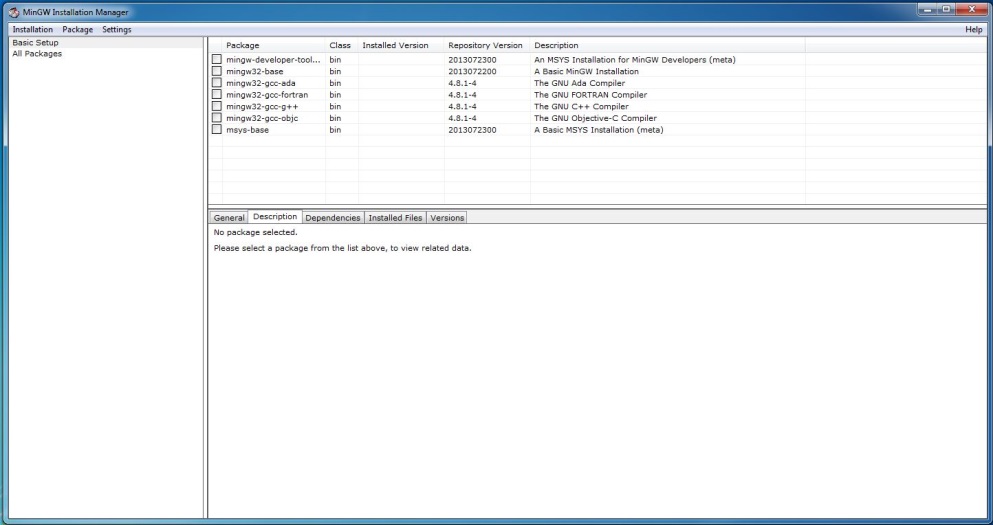
Download and run mingw-get-setup.exe from <http://sourceforge.net/projects/mingw/>. This will download the MinGW package installer. Click install, continue, and continue a second time.



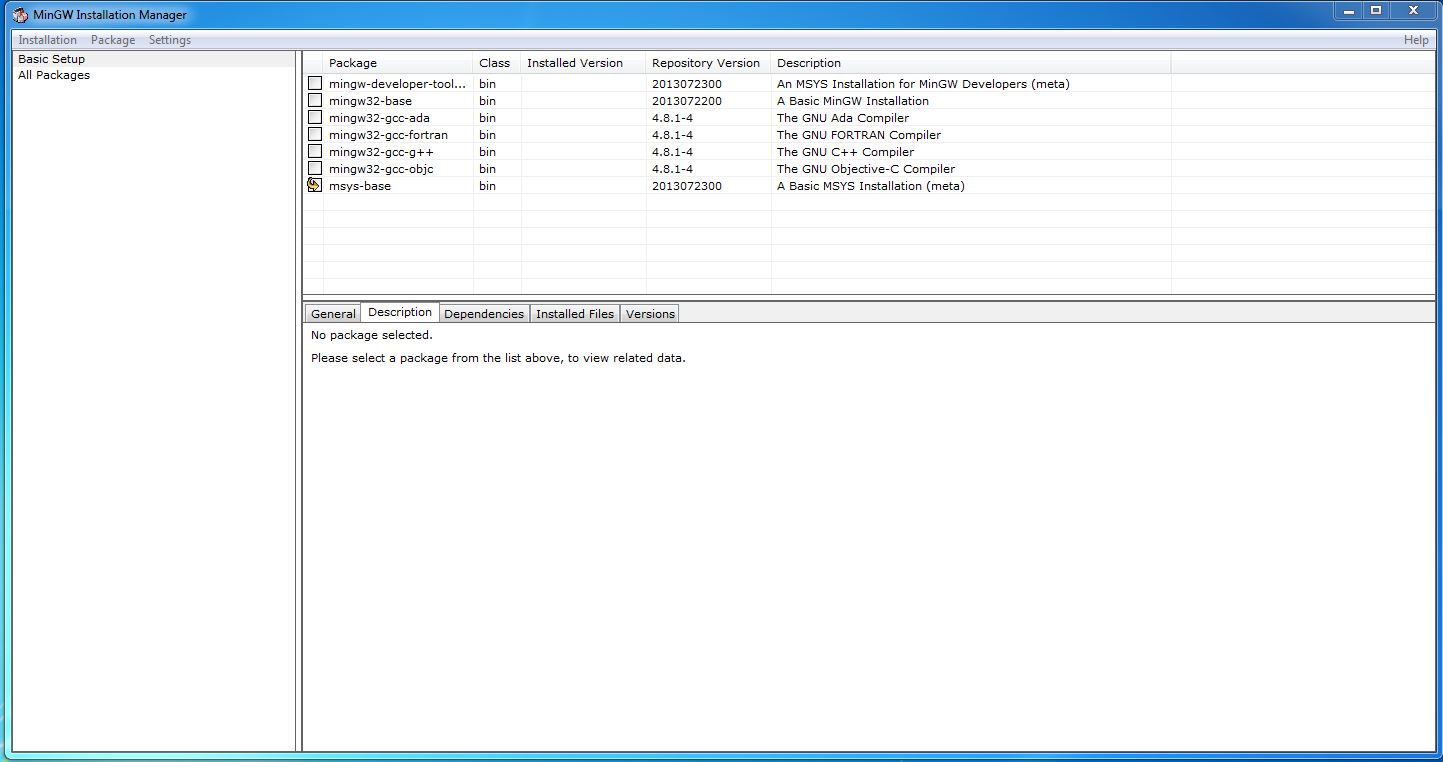




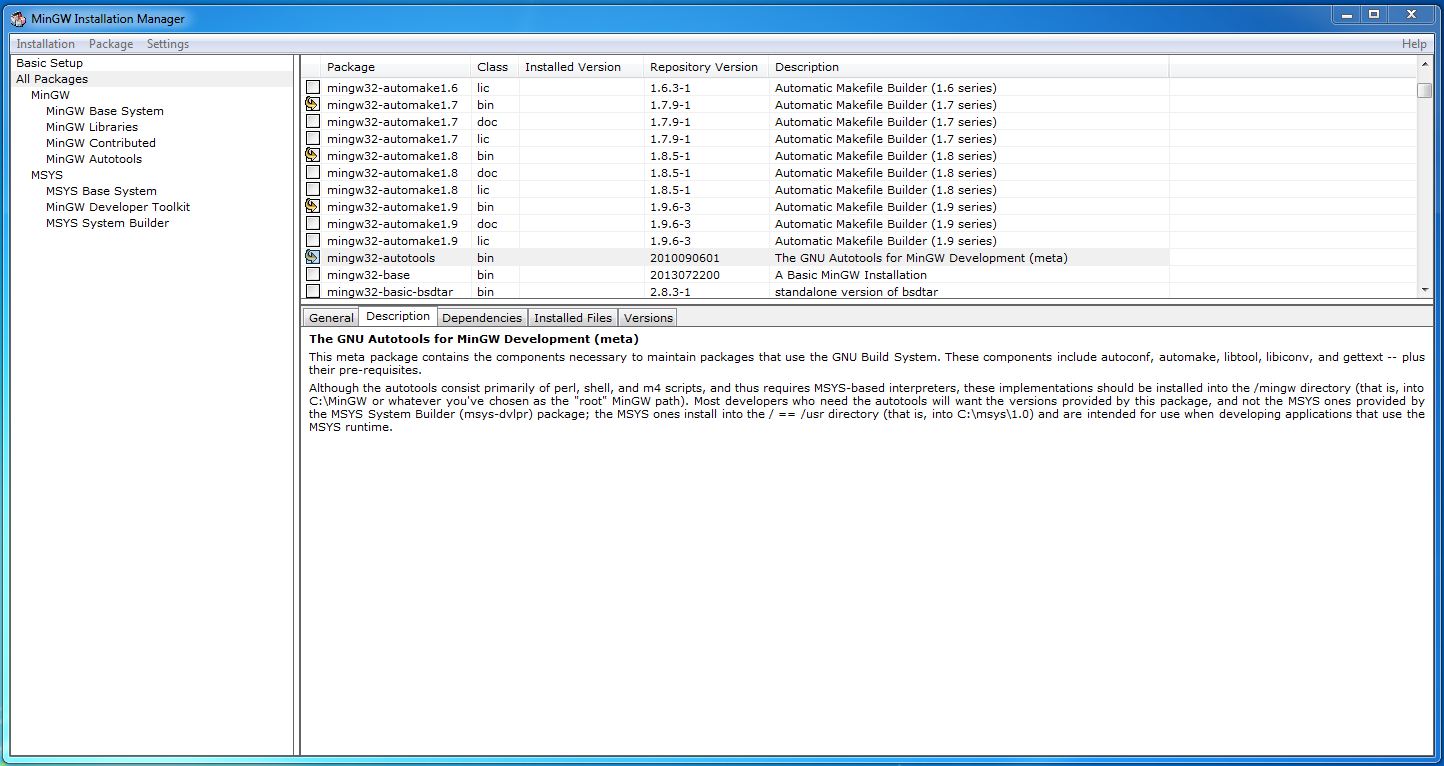
After clicking Continue the MinGW installer window should popup. From here The correct packages MinGW and msys will be chosen and installed on the computer.



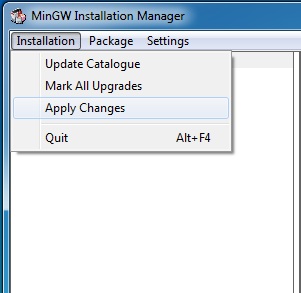
Click on Basic Setup in the left section of the window. In the top right section of the window click on the msys-base package and select Mark for Installation.



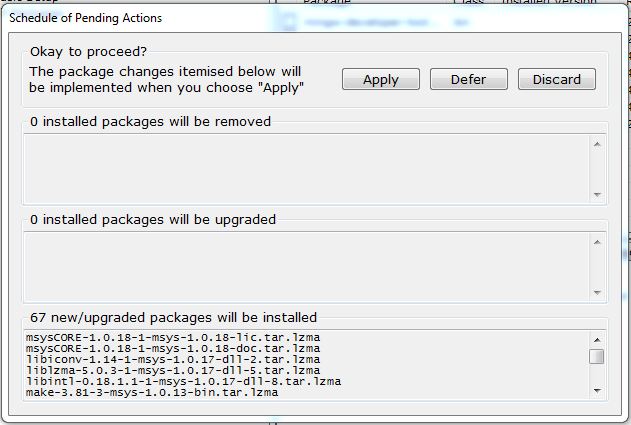
Now Click on All Packages in the left section of the window. In the Top Right section of the window scroll down and click on mingw32-autotools and select Mark for Installation. This will automatically select multiple packages.



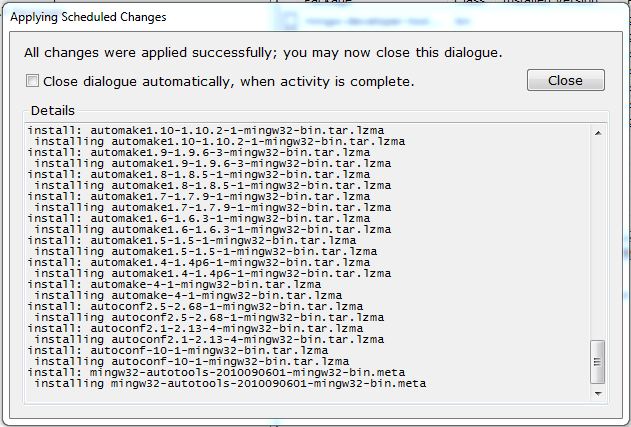
Now click on Installation and select Apply Changes.



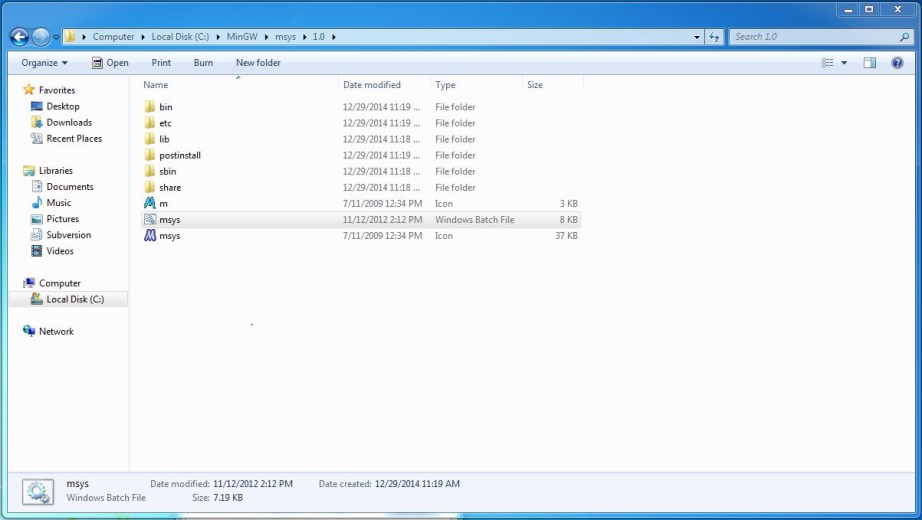
Then Click Apply.



Once the packages have been installed click Close.

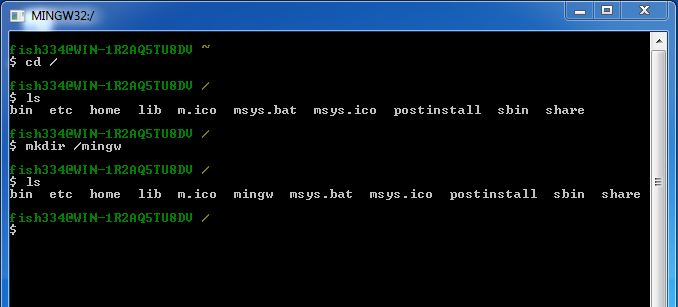


Now There should be a C:\MinGW directory. Open Explorer and navigate to C:\MinGW\msys\1.0 and run msys.bat.



A Linux terminal window should now be open. This is where GridLAB-D will be built from using autotools. Make a directory in the root called mingw. Type

* cd /
* mkdir mingw

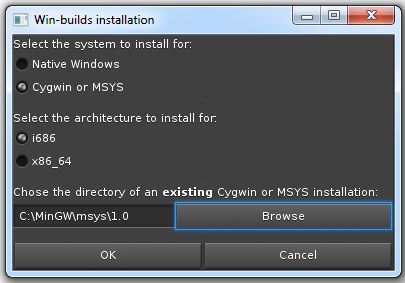


Close the terminal. Navigate to C:\MinGW\msys\1.0\etc. Copy fstab.sample. Paste it in the same directory and rename it fstab (remove the extension).

Win-Builds:

Once MinGW and MSYS have been installed, Win-Builds can be installed. Go to http://win-builds.org and download and run the package manager (win-builds-1.5.exe).

You need to run this executable twice in order to get both 32bit and x64 build capability for MinGW. The 32bit architecture looks like this.

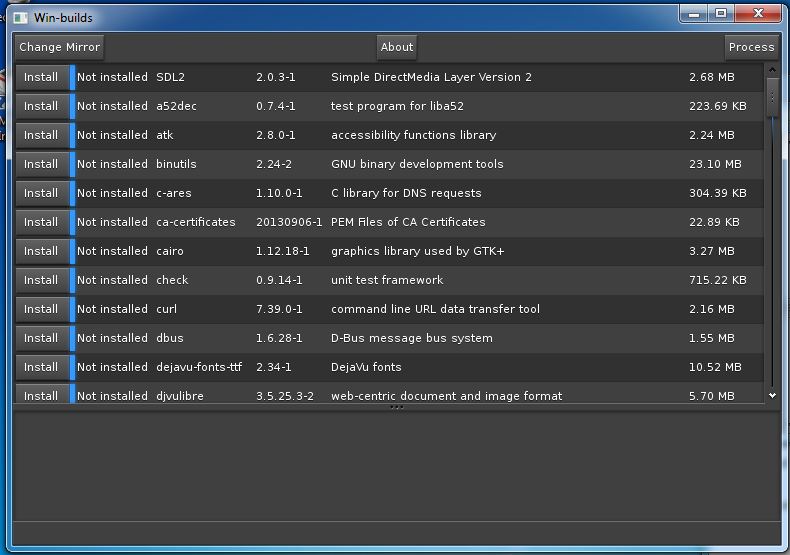


And the 64 bits architecture looks like this.

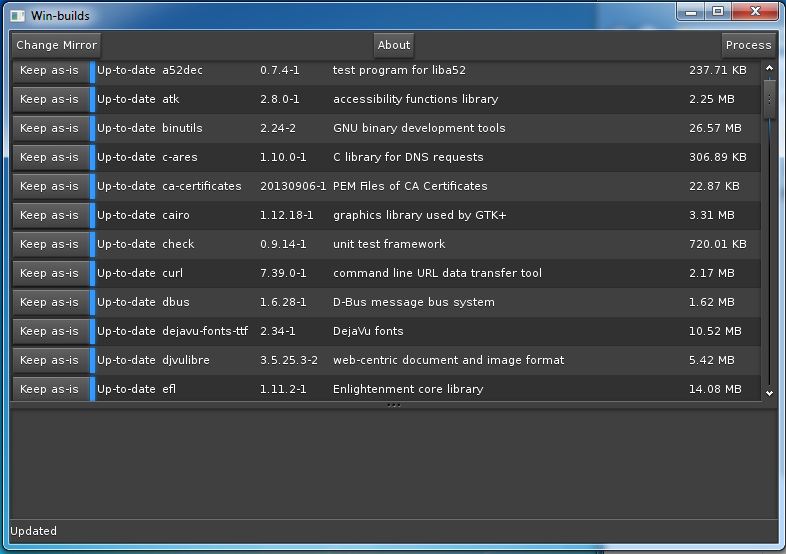


Make sure Cygwin or MSYS is selected in either case. The directory of the existing MSYS installation should be C:\MinGW\msys\1.0. Once the desired architecture is selected, click OK.

This window should pop up. Click on Process and that will install all the necessary win-build packages to C:\MinGW\msys\1.0\opt\windows\_32 or C:\MinGW\msys\1.0\opt\windows\_64.



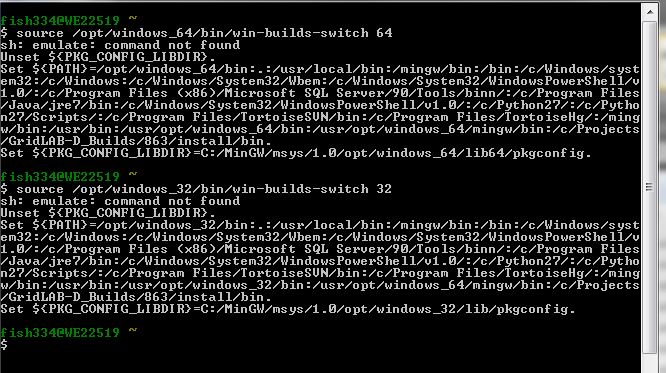
When it is finished processing the window will look like this. It is ok to close the window at this point.



Switch Between 32 and 64 Bit Build Architectures:

Every time the msys.bat file is run the build architecture needs to be set. To select the architecture desired type the following

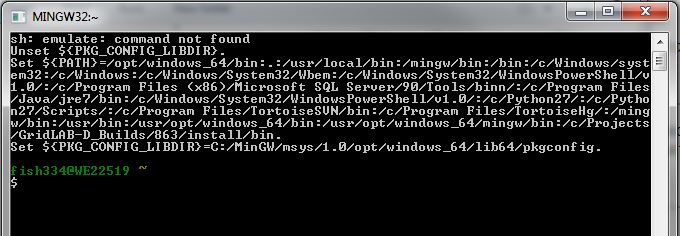
* source /opt/windows\_32/bin/win-builds-switch. (32 bit architecture)
* source /opt/windows\_64/bin/win-builds-switch. (64 bit architecture)



A default build architecture can be set by adding it to the user profile demonstrated below.

* echo ‘source /opt/windows\_32/bin/win-builds-switch’ >> ~/.profile. (32 bit default)
* echo ‘source /opt/windows\_64/bin/win-builds-switch’ >> ~/.profile. (64 bit default)

Now whenever msys.bat is run the window should show.



Eclipse:

This instruction set will be done using 64 bit Eclipse. If you use 64 bit Eclipse you must make sure you have a 64 bit version of Java installed. If you don’t have 64 bit version of Java you can go to <http://java.com/en/download/manual.jsp> and under windows select Windows Offline (64-bit) and follow the installation process.

Go to <http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/lunar> and click on the Windows 64-bit link under Download Links. Download eclipse-cpp-luna-R-win32-x86\_64.zip. Notice that this is a zip folder so there is no installation wizard. Once it’s downloaded simply extract the folder named eclipse from the zip folder to the desired location of your choosing and Eclipse is installed.

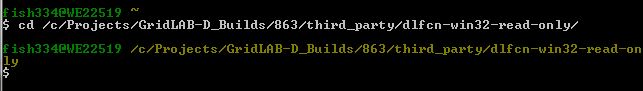
ATTENTION: Before installing dlfcn, xerces, and GridLAB-D severall things need to be added to the PATH environment variable. The following paths are:

* C:\MinGW\msys\1.0\bin
* For 32 Bit Builds
  + C:\MinGW\msys\1.0\opt\windows\_32\bin
  + C:\MinGW\msys\1.0\opt\windows\_32\mingw\bin
* For 64 Bit Builds
  + C:\MinGW\bin
  + C:\MinGW\msys\1.0\opt\windows\_64\bin
  + C:\MinGW\msys\1.0\opt\windows\_64\mingw\bin

These can be added to PATH in Windows 7 by navigating to Control Panel > System and Security > System, clicking on Advanced system settings, and clicking on Environment Variables. Then path can be edited from there.

DLFCN:

Download ticket 863 to a location of your choice here by referenced as TICKET\_DIR. It is required that TICKET\_DIR contain no spaces. Run msys.bat. From here cd to TICKET\_DIR/third\_party/dlfcn-win32-read-only. You can specify the C drive by using /c/<rest of the path> as illustrated below.



To install type the following:

* (32 bit) ./configure --prefix=/opt/windows\_32/mingw --enable-shared
* (64 bit) ./configure --prefix=/opt/windows\_64/mingw --enable-shared
* make
* make install

XERCES:

From the linux shell cd to TICKET\_DIR/third\_party. Unzip xerces-c-3.1.1.tar.gz.

* tar -xvf xerces-c-3.1.1.targ.gz

Now cd to xerces-c-3.1.1 and run the autotools make process as follows.

* (32 bit) ./configure --build=i686-w64-mingw32 --prefix=/opt/windows\_32/mingw
* (64 bit) ./configure --build=x86\_64-w64-mingw32 --prefix=/opt/windows\_64/mingw
* make
* make install

GridLAB-D:

This instruction set installs GridLAB-D to a folder inside TICKET\_DIR called install32 or install64. There is no need to make the directory before installing. If you wish to change where GridLAB-D gets installed simply supply a different path to --prefix at the configure step.

The install steps are as follows. cd to TICKET\_DIR.

* autoreconf -if
* (32 bit) ./configure --build=i686-w64-mingw32 --prefix=$PWD/install32 --with-xerces=/opt/windows\_32/mingw/bin --with-xerces-inc=/opt/windows\_32/mingw/include --with-xerces-lib=/opt/windows\_32/mingw/lib
* (64 bit) ./configure --build=x86\_64-w64-mingw32 --prefix=$PWD/install64 --with-xerces=/opt/windows\_64/mingw/bin --with-xerces-inc=/opt/windows\_64/mingw/include --with-xerces-lib=/opt/windows\_64/mingw/lib
* make (Note: In order to debug the code edits to the makefile created by ./configure before executing make. Edit lines 678 and 690 and comment out -O2. ‘#’ is the comment character.)
* make install

Linking with MATLAB:

In order to build GridLAB-D with MATLAB you must set the option --with-matlab=<path to MATLAB install> at the ./configure step of the installation. When running a simulation with MATLAB %PATH% must contain the path the MATLAB dlls, <MATLAB install>\bin\win64 or <MATLAB install>\bin\win32. It is important that no space be contained in the paths. If there are spaces in the paths you can find the short name windows uses for the directory by opening up a command window and typing dir /X one directory above the directory with the spaces. An example of what --with-matlab might look like is below.

* Install path on the windows explorer: C:\Program Files\MATLAB\R2012a.
* Option set with short names: --with-matlab=/c/PROGRA~1/MATLAB/R2012a.

Linking with MySQL:

In order to build GridLAB-D with MySQL you must set the option --with-mysql=<path to the mysql connector folder> at the ./configure step of the installation. When running a simulation with MySQL %PATH% must contain the path the MySQL dlls, <MySQL install>\lib. It is important that no space be contained in the paths. If there are spaces in the paths you can find the short name windows uses for the directory by opening up a command window and typing dir /X one directory above the directory with the spaces.

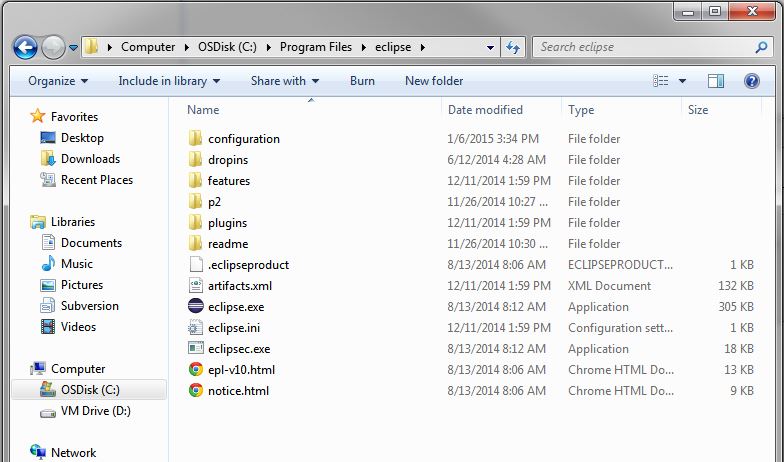
* Install path on the windows explorer: C:\Program Files\MySQL\MySQL Connector C.
* Option set with short names: --with-mysql=/c/PROGRA~1/MySQL/MYSQLC~1.

# Running GridLAB-D:

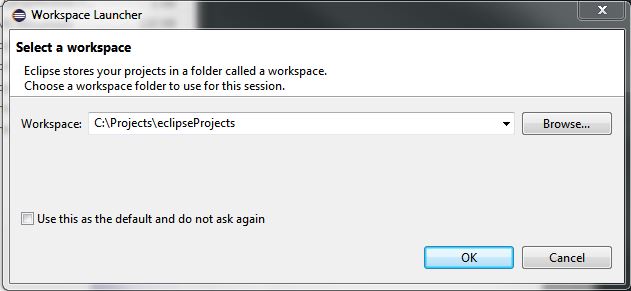
Eclipse:

Before opening Eclipse, the path to gridlabd.bin.exe must be added to the PATH environment variable. If you followed the instruction above then gridlabd.bin.exe is located in TICKET\_DIR\install\_32 or install\_64\bin.

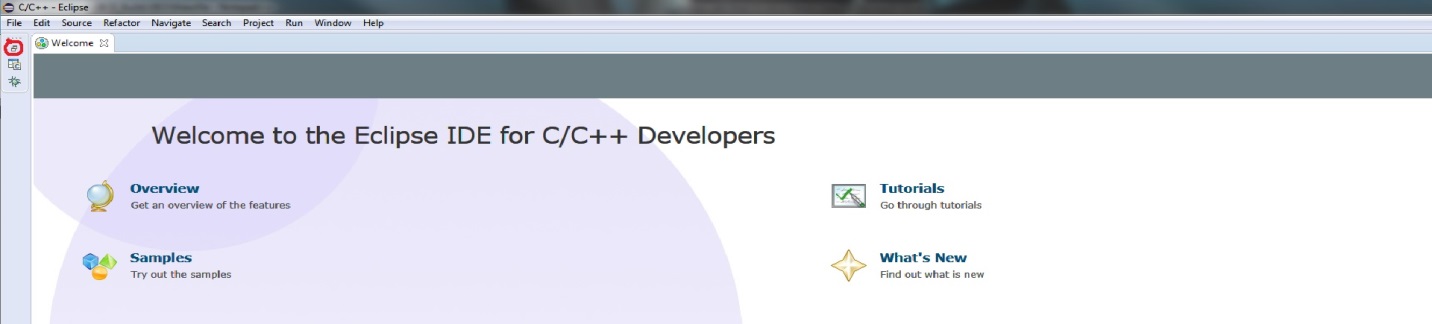
In the explorer navigate to where the eclipse folder was extracted. Inside the eclipse folder, double click eclipse.exe



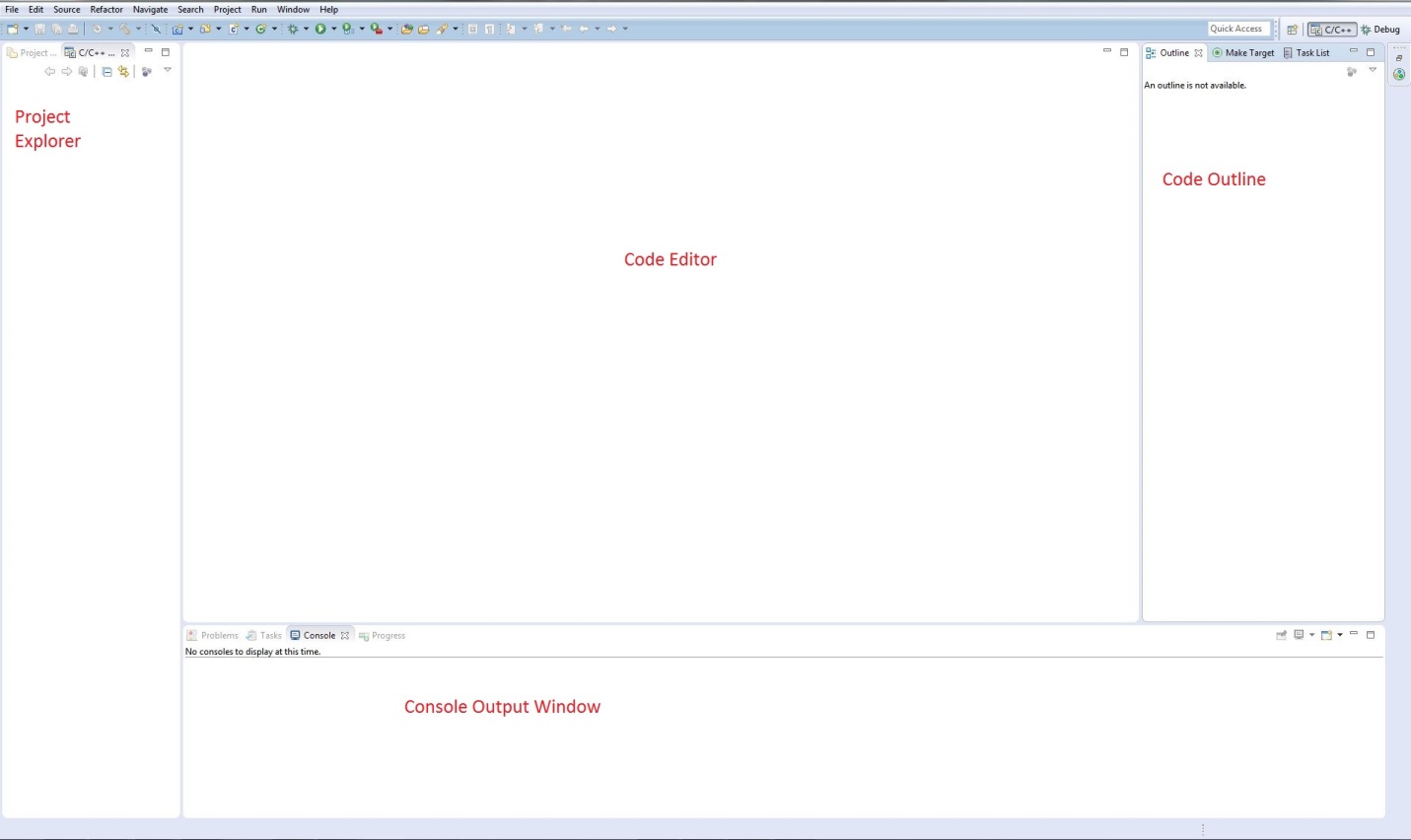
It will ask for a path to use for projects space. Choose your desired path and click OK.



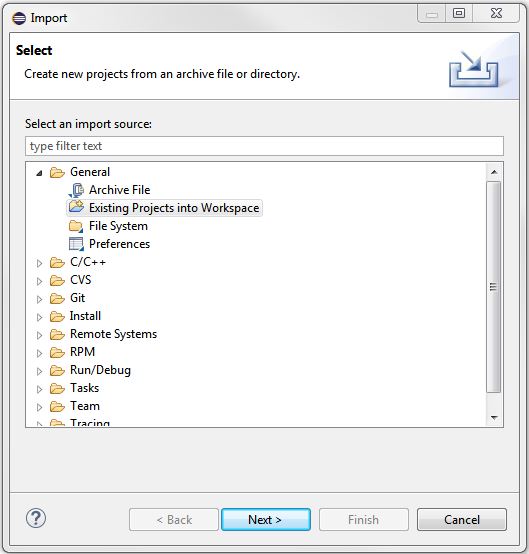
Since you are opening Eclipse for the first time the welcome view is displayed. To get to c++ view select the circled icon.



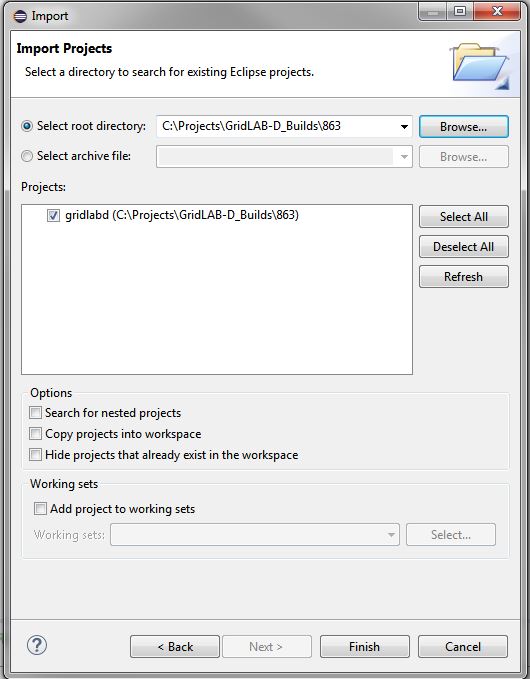
If so desired you can minimize the welcome view altogether. The picture below shows the layout of the C/C++ View.



Before loading and running GridLAB-D. It is a good Idea to check for updates. This can be done by selecting Help > Check for Updates. Once that is finished Lets go ahead an import the ticket project into Eclipse. Select File > Import and the Import window will pop up. Expand General and select Existing Projects into Workspace and click Next.

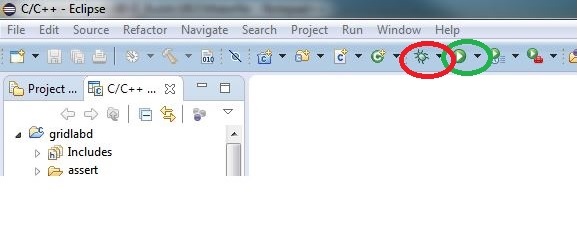


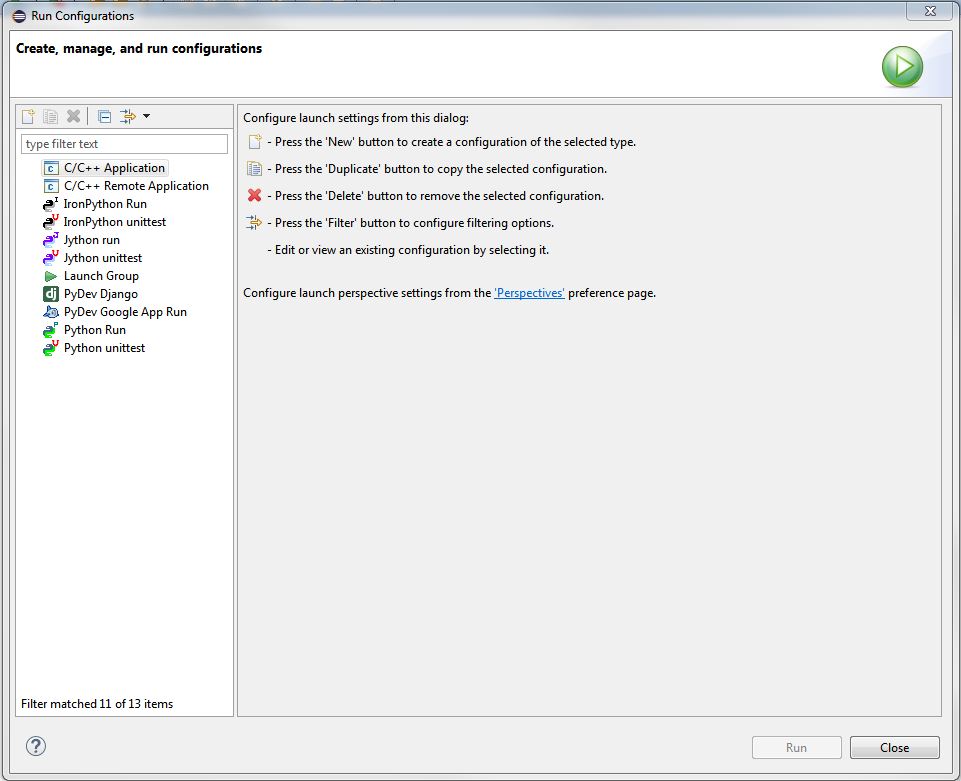
Provide the Path to the TICKET\_DIR and select Finish.

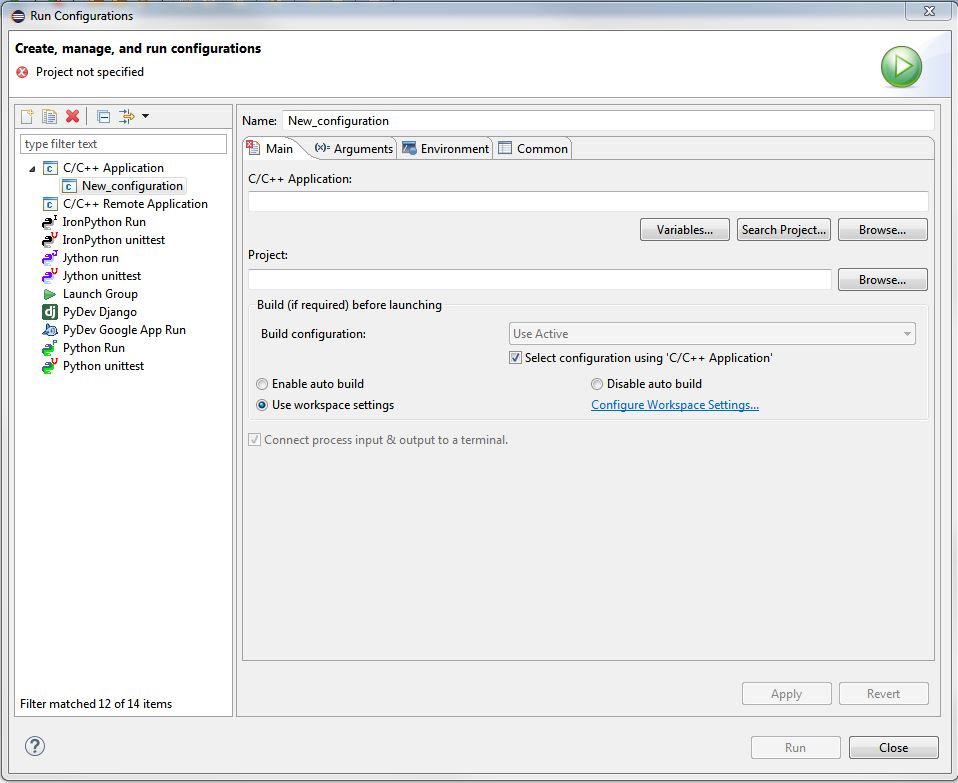


You should now see a project called gridlabd in the project explorer. Expanding this will give you the ability to see all the files in TICKET\_DIR.

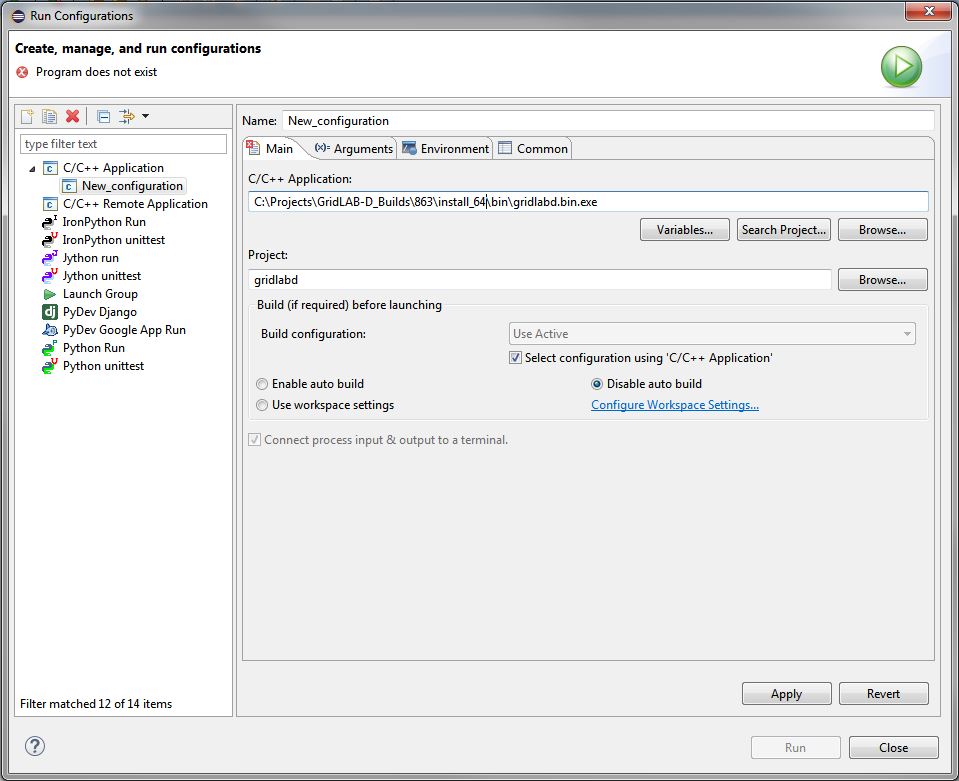
In order to run a GridLAB-D model, the run icon should be used. The run icon is circled in green in the picture below. To run the same file to debug code the debug icon should be used. The debug icon is circled in red below.



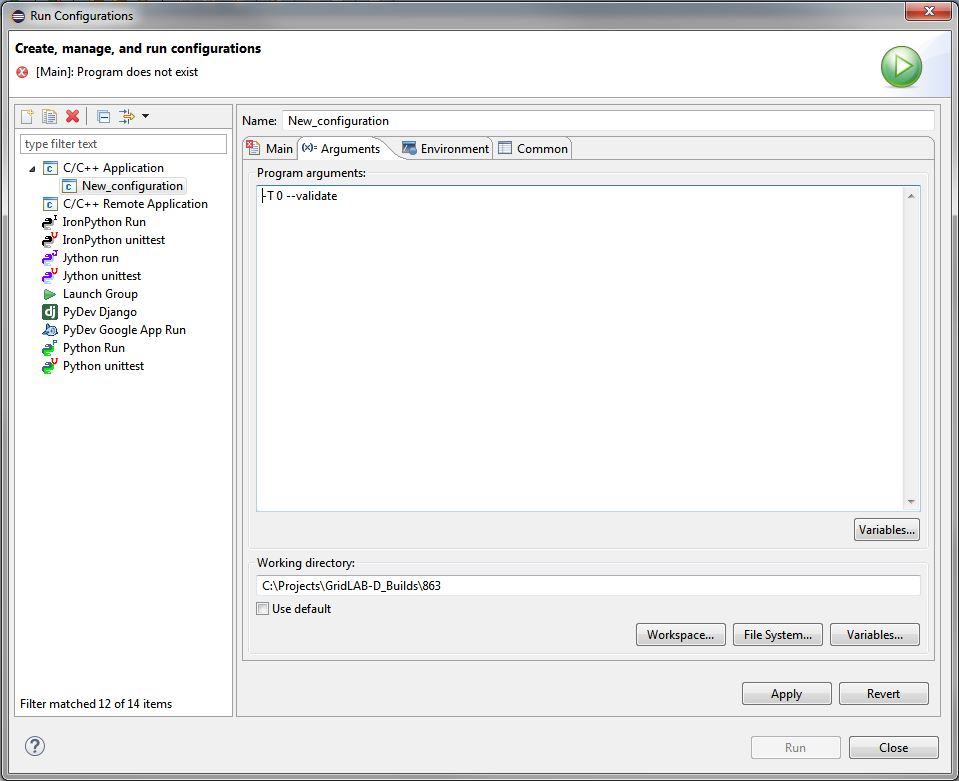
However, we need to configure the Run and Debug first. Clicking on the down arrows will give you the access to Run Configurations and Debug Configurations. Let’s set up a run configuration that will run the validation script so select Run Configurations to open the Run Configurations window. 

Double click C/C++ Application to create a new Run configuration.

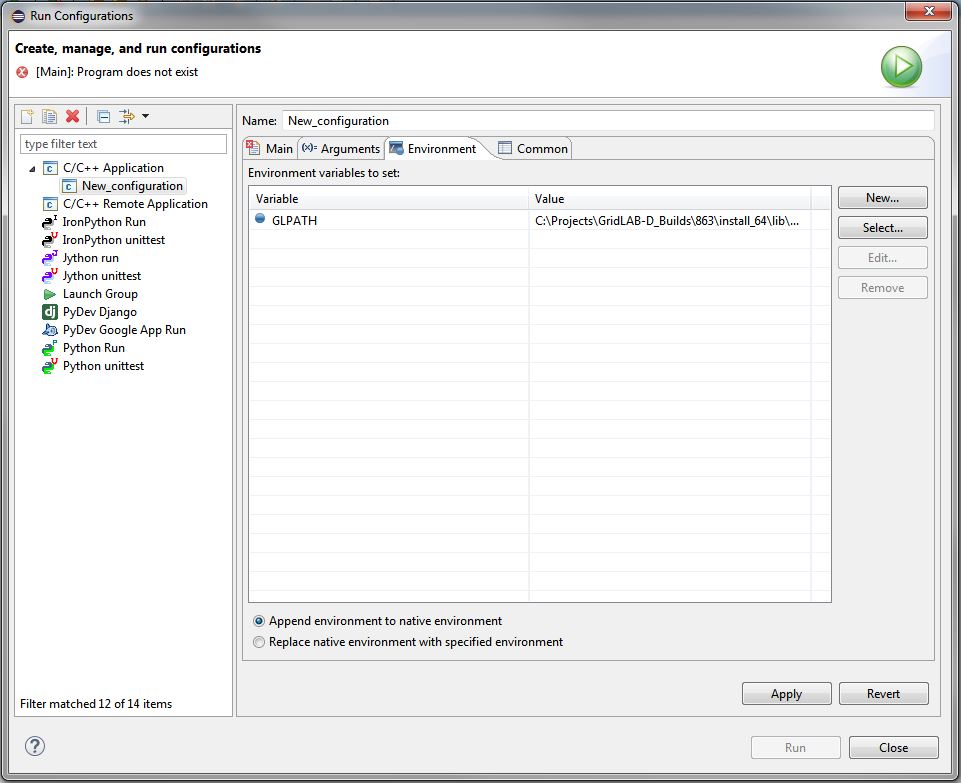
Give the configuration a useful name. Under Main > C/C++ Application Browse for the gridlabd.bin.exe that you build. Make sure the Project is the project in your project explorer. Make sure Disable auto build is selected as well.



Under Arguments place the valide commands to run the validation script. And make sure the working directory is the TICKET\_DIR. Regardless, This is where gridlabd commands would be placed for running various .glms.



Finally, under Environment, A new variable called GLPATH needs to be added. It needs to contain the paths to <GridLAB-D install path>\lib\gridlabd and <GridLAB-D install path>\share\gridlabd. Once that’s done you can select Apply and Run and the validation script should start running and the Console should start displaying the output.



The exact same process needs to be applied to the Debug Configurations. You can set breakpoints through the code editor window. When the Debug run begins it will ask if you want to switch to the Debug view. Click yes. The debug run always stop at the beginning of main.c. This ok just simply tell it to continue to the next breakpoint and you are on your way to debugging code in Eclipse.

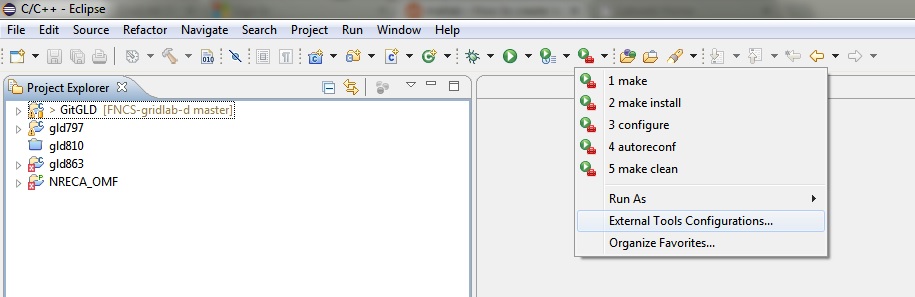
Remember: Once you make edits you need to go back to msys.bat to rebuild GridLAB-D.

# Linux OS:

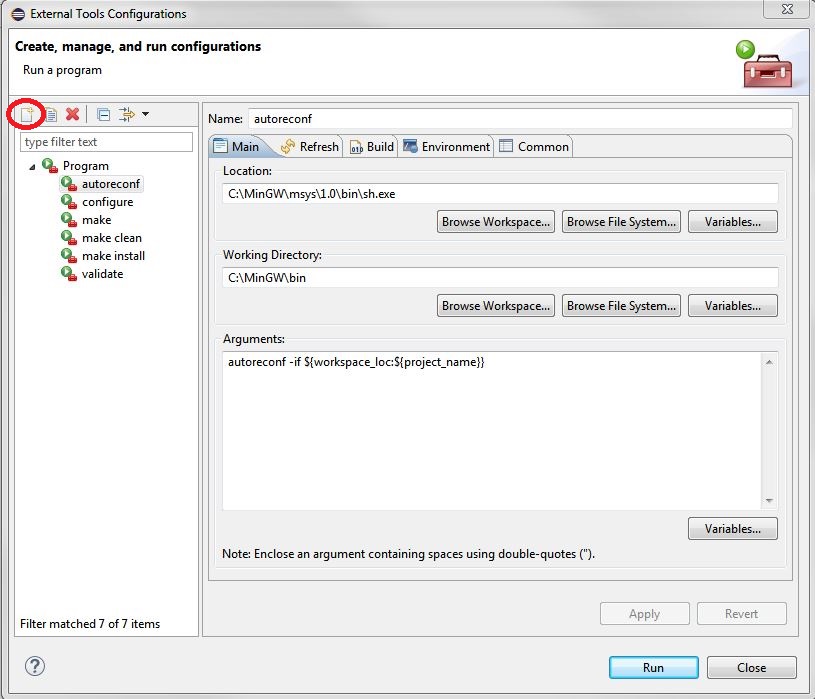
The Linux OS is much simpler. All you need is ticket 863, Eclipse, and Xerces. No fancy prefixes are needed for installing exerces, a simple ./configure, make, and make install is all that is needed. Eclipse needs to be extracted from a tar file instead of a zip file. And the installation for GridLAB-D remains the same as it has always been. Although, I do recommend installing to a custom prefix. The steps for running GridLAB-D from Eclipse remain the same as stated previously above.

# Creating Custom Autotool commands in Eclipse:

This section will cover creating custom autotool commands in Eclipse using the external configuration tool shown below.

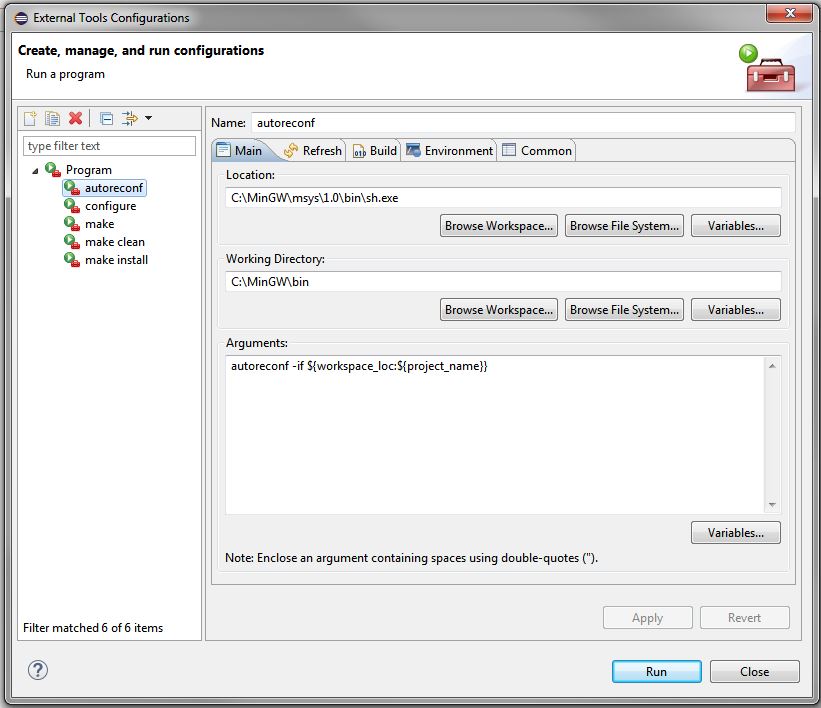


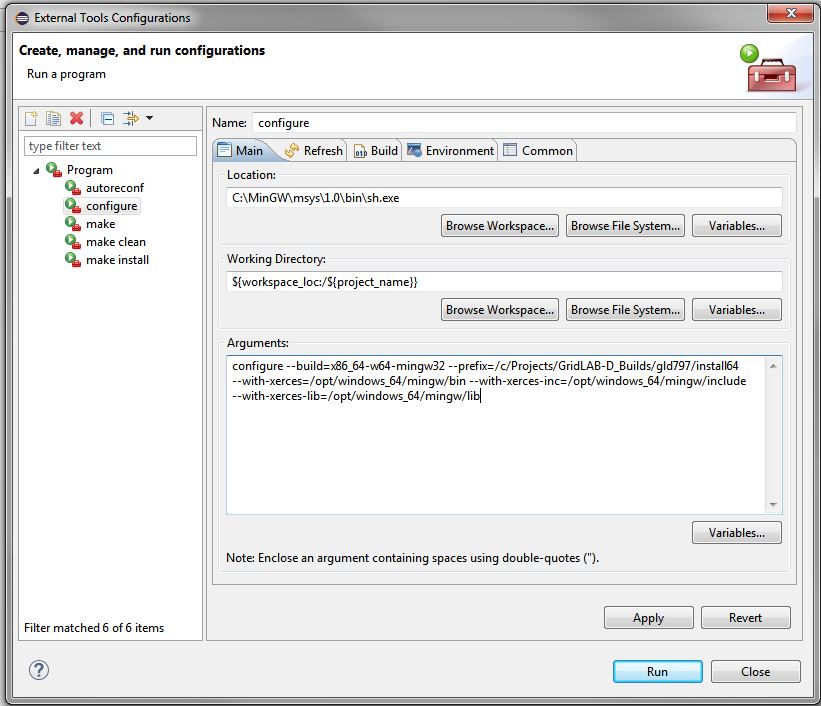
The external configuration tool can be used to create autotool commands for building GridLAB-D. an Autoreconf, configure, make, make install, and make clean tool can be created by clicking on the new icon shown below.

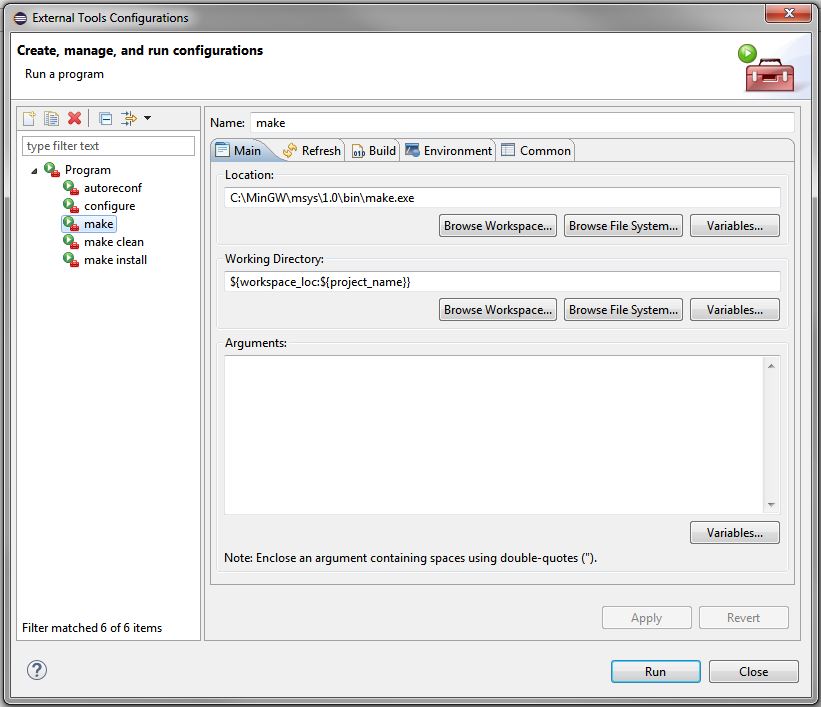


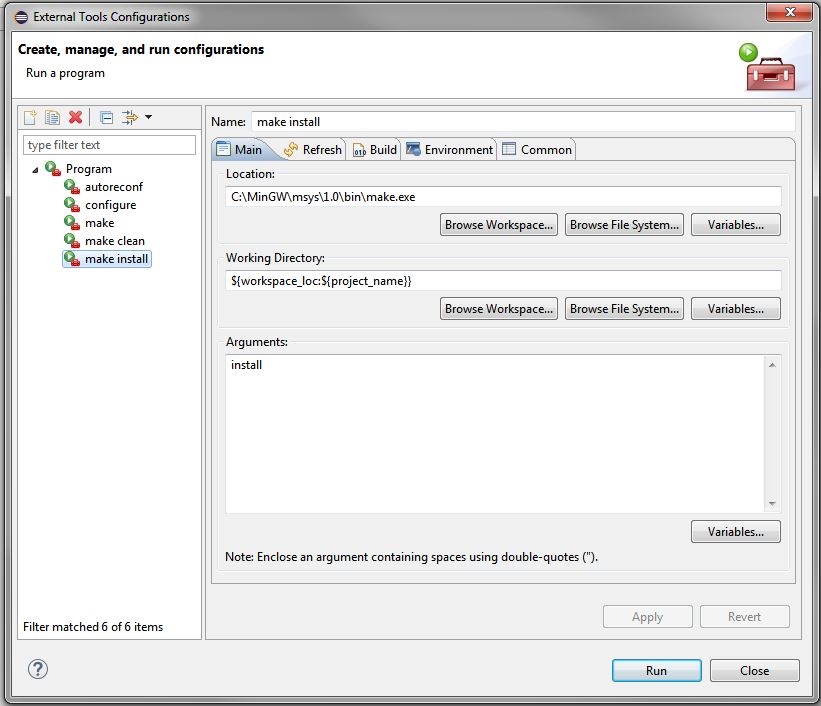
The pictures shown below show how to set up each tool.

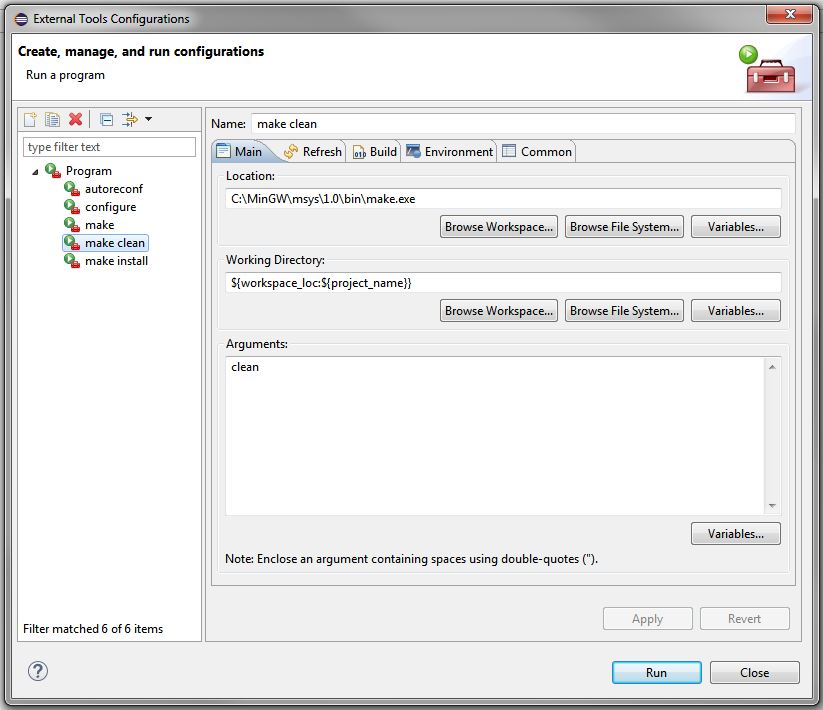
On a Windows + MinGW system.











Please note that the paths shown may vary depending on where you installed MinGW on your system.

On Linux Systems.

