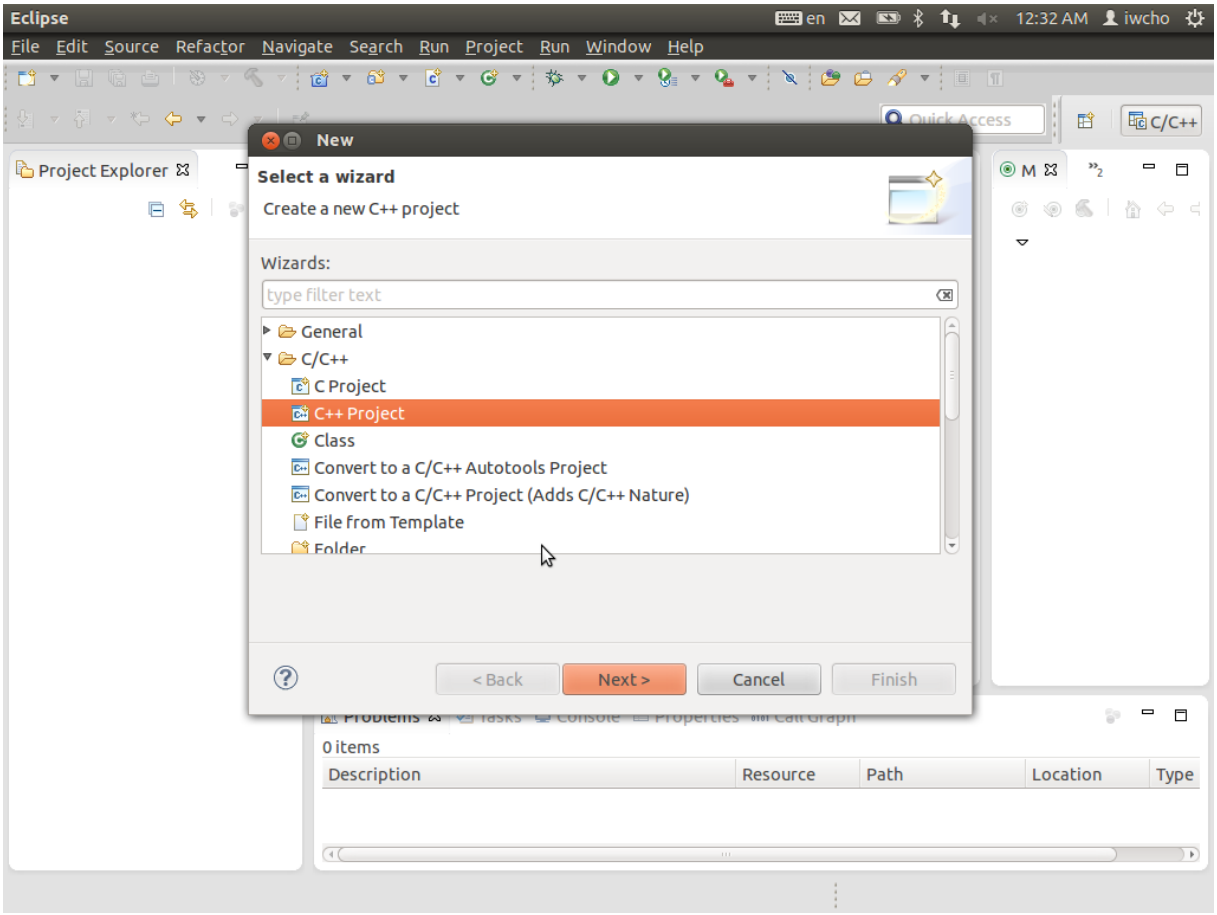
TUTORIAL ON MANUALLY CREATING AN ECLIPSE PROJECT AROUND BRL-CAD

Made by Iwan Manew

BRL-CAD build system uses CMake and so I will show you how to build a project from cross-platform CMake on Eclipse. The CMake 2.8 project generator for Eclipse does not work, so you must create the project and configure it to build with GNU Make. Here's how to do it on Linux.

Create an Eclipse Project

Create an Eclipse CDT (C/C++ Development Tooling) project using the *File > New > C++ Project* command for your C++ project, or *File > New > C Project* for a C project.
Do not create the project using the CMake Eclipse project generator.

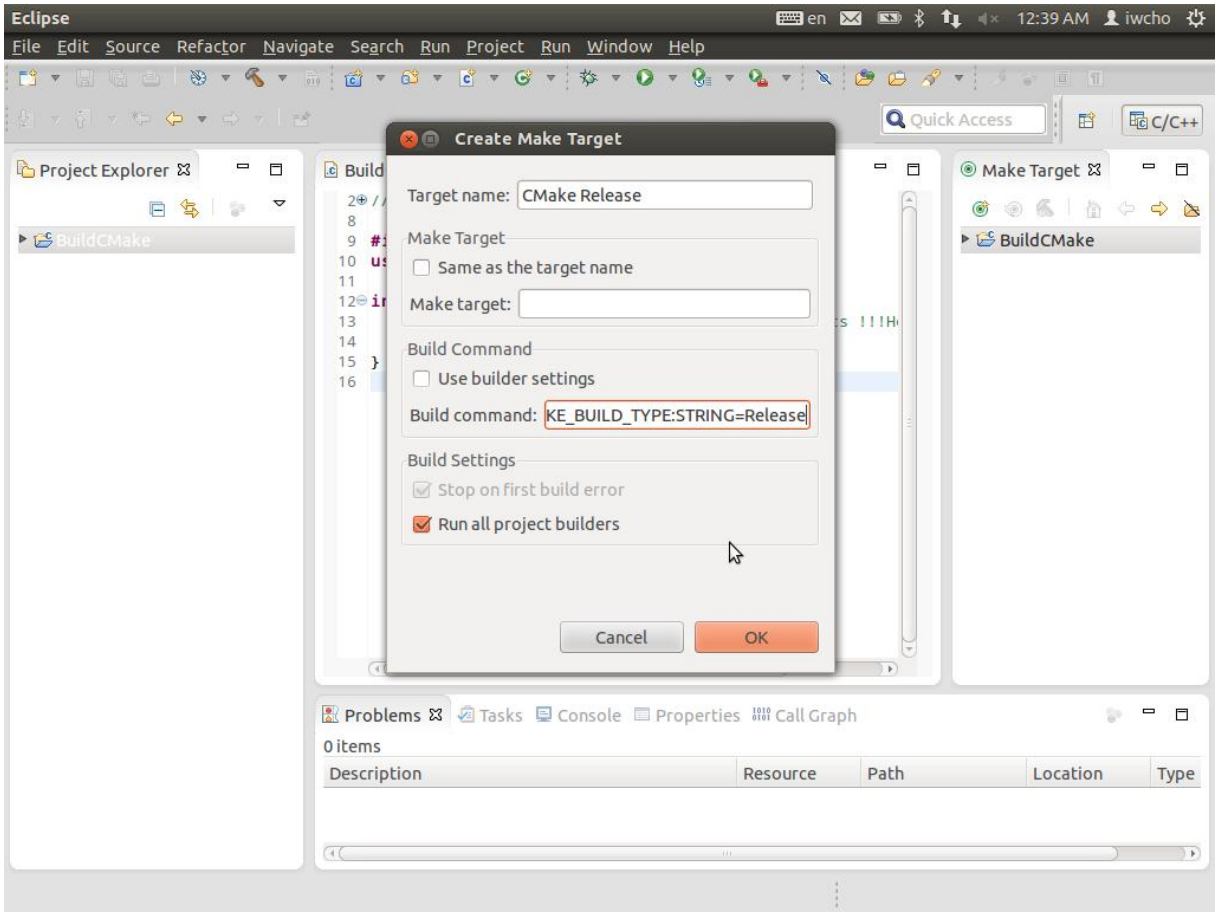


Create Eclipse Make Targets

The conventional approach to using CMake with Eclipse is to create an *external tool* in Eclipse. However, a *Make Target* is simpler, and because it is stored in the Eclipse .project file, you can check it into your version control system and it will work in every one of your working copies, on every computer.

Create a Make Target for each configuration that you want to build. Here I assume that you have the usual *Release* and *Debug*configurations:

- Display the *Make Target* window using the*Window* > *Show View* > *Make Target* menu command. It should appear on the right, with the *Outline* window.
- Select the folder for the project for which you want to add CMake. CMake will run with this folder as its working directory.
- Right click on the folder and select *New* from the context menu. The *Create Make Target* dialog will appear.
  - Type *Target name* **CMake Release**
  - In *Make target*, deselect *Same as the target name*, and make sure that the*Make target* field is empty
  - In *Build Command*, deselect *Use builder settings* and set the *Build command* to **cmake -E chdir Release/ cmake -G "Unix Makefiles" ../ -DCMAKE_BUILD_TYPE:STRING=Release**
  - Click *OK*



- Repeat, this time for *Target name* CMake Debug, and *Build command*:

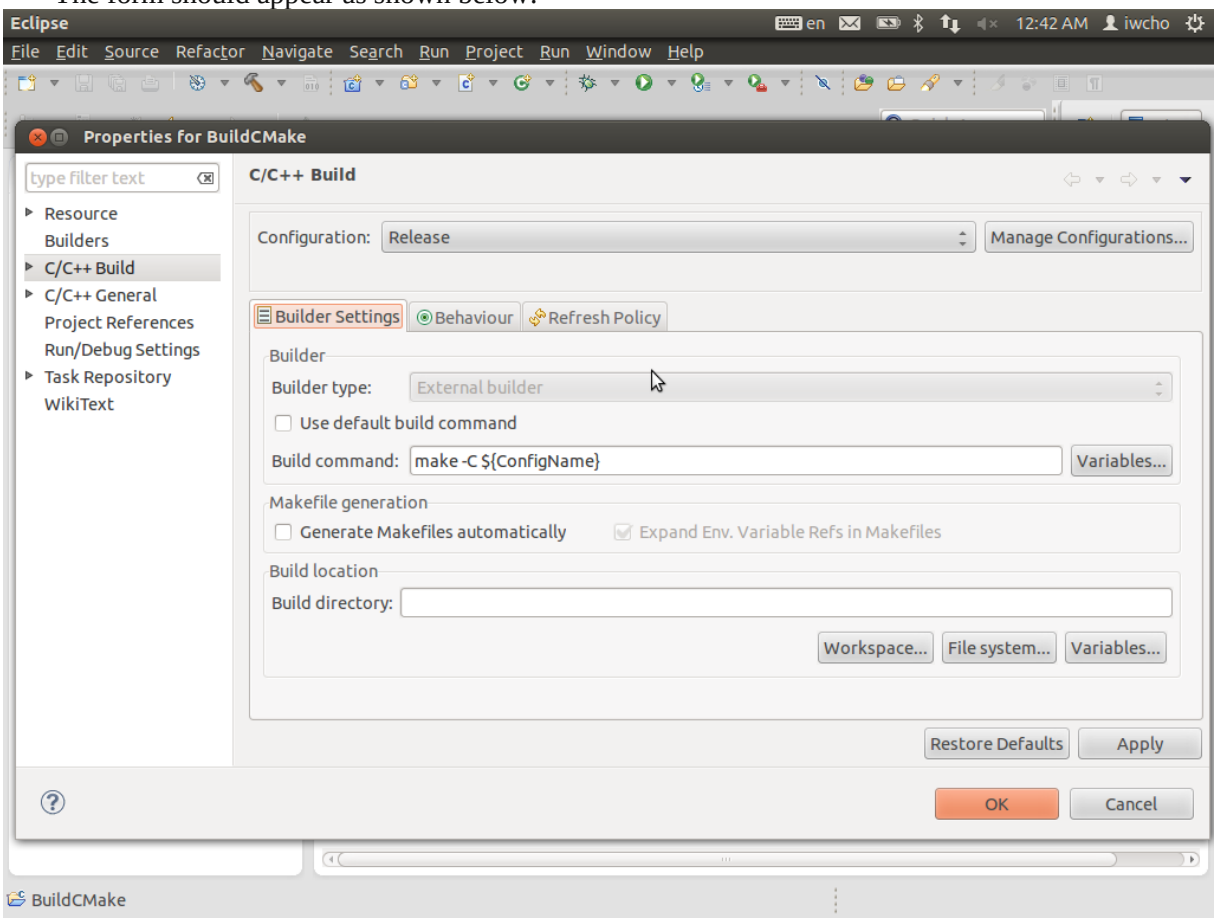
cmake -E chdir Debug/ cmake -G "Unix Makefiles" ../ -DCMAKE_BUILD_TYPE:STRING=Debug

- Create the **Release/** and **Debug/** directories
  mkdir Release Debug

Set Up the Eclipse CDT Builder

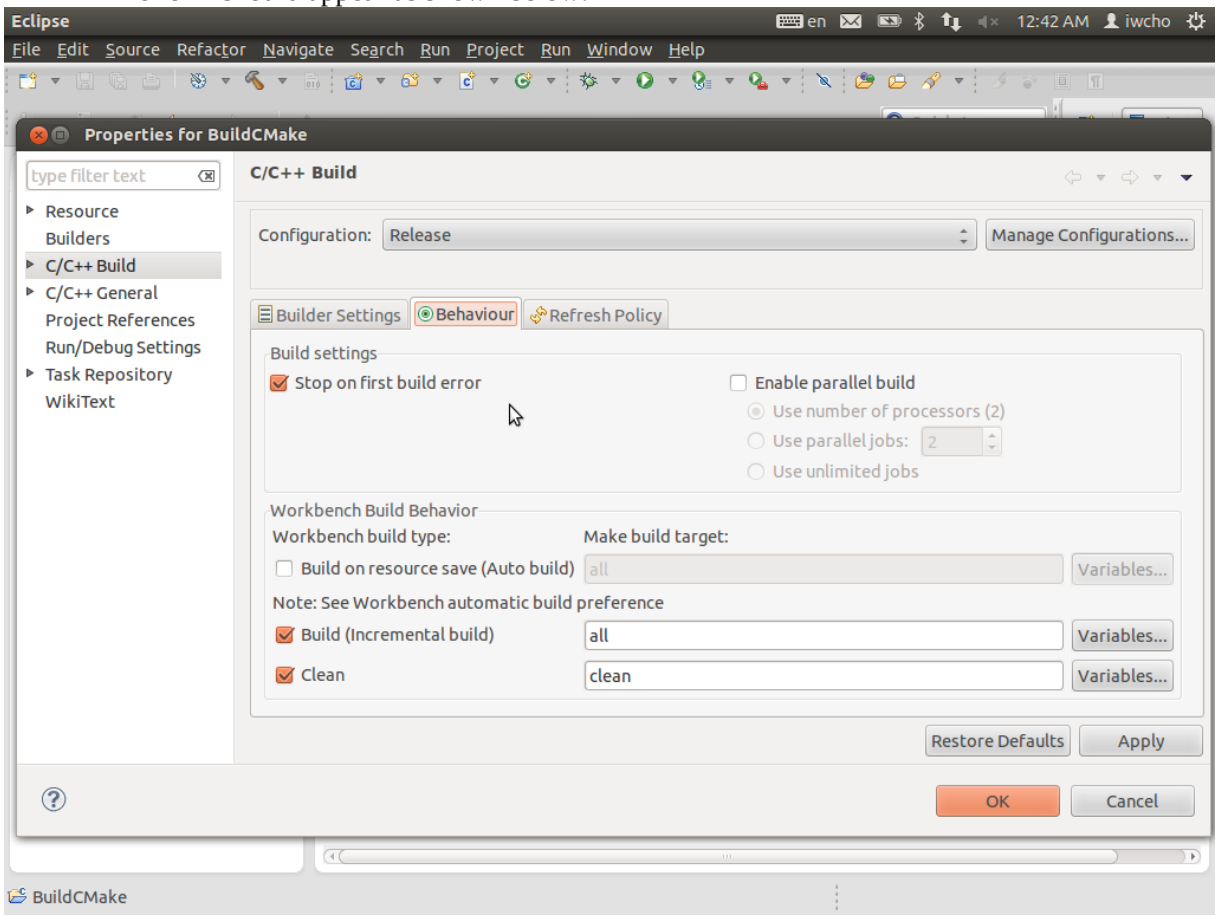
Next, set up the CDT builder to run the Makefiles that CMake builds.

- Right click on a CDT project. In the context menu, select *Properties*.
- On the left, select *C/C++ Build*
  - Set *Configuration* to *Release*
    - Choose the *Builder Settings* tab
      - Deselect *Use default build command*
      - Specify the *Build command*:
        make -C ${ConfigName}
      - Deselect *Generate Makefiles automatically*
      - Make the *Build directory* field blank
      - The form should appear as shown below:

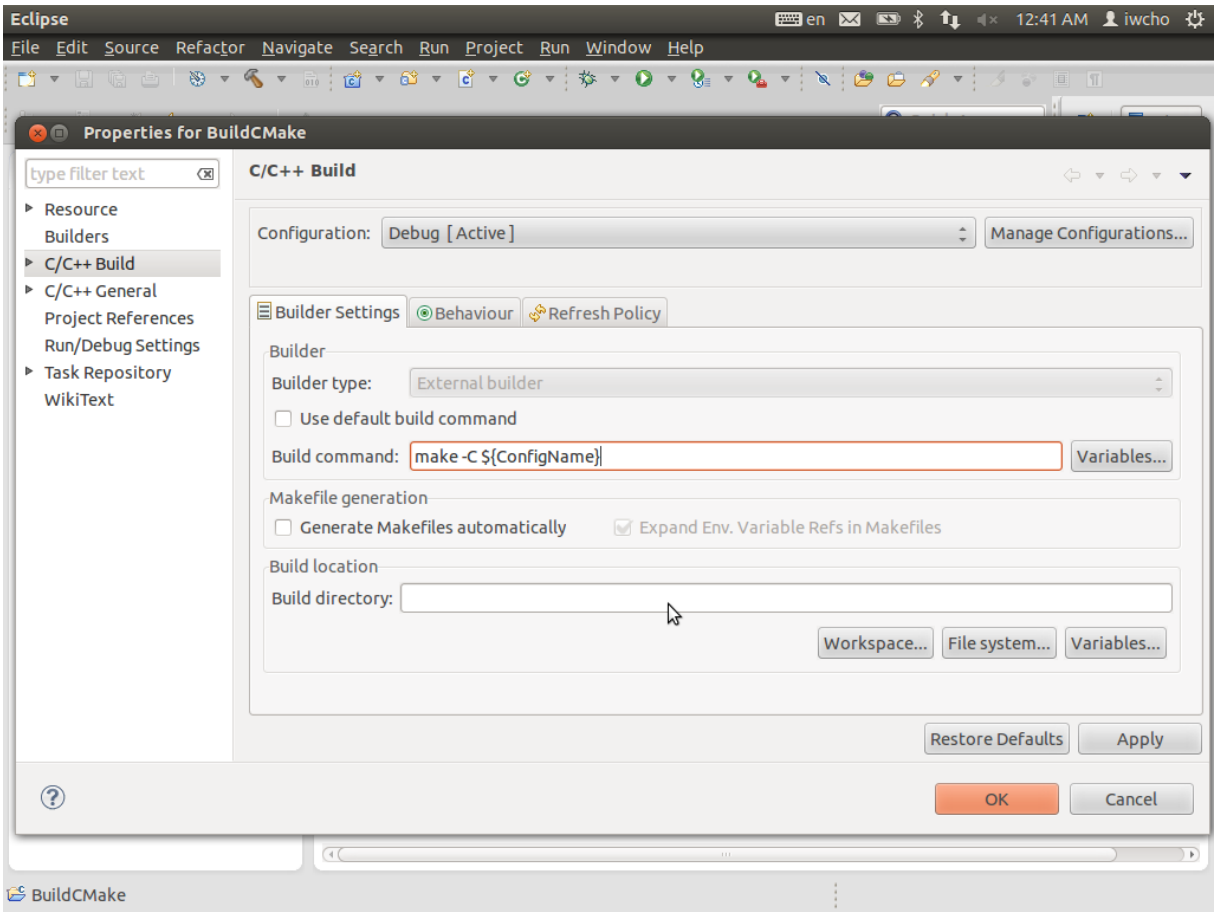


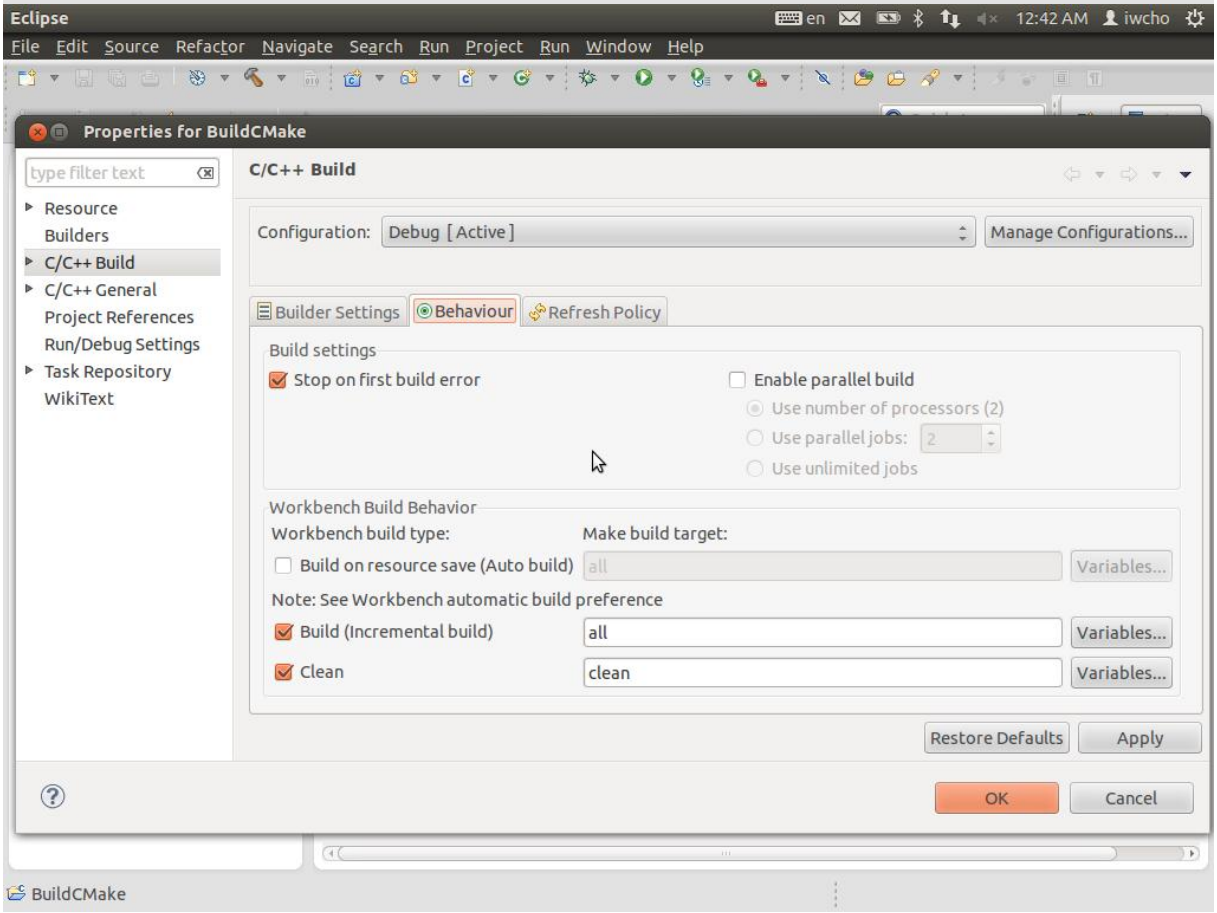
    - Choose the *Behavior* tab
      - Select *Build (Incremental build)* and specify the target name *all*

- Select *Clean* and specify the target name *clean*
- The form should appear as shown below:



- Set *Configuration* to *Debug*
  - Choose the *Builder Settings* tab
    - Set all values exactly the same as the *Release* configuration
  - Choose the *Behavior* tab
    - Set all values exactly the same as the *Release* configuration


Eclipse
en 12:41 AM iwcho
File Edit Source Refactor Navigate Search Run Project Run Window Help
Properties for BuildCMake
type filter text
Resource
Builders
C/C++ Build
C/C++ General
Project References
Run/Debug Settings
Task Repository
WikiText
C/C++ Build
Configuration: Debug [Active]
Manage Configurations...
Builder Settings
Behaviour
Refresh Policy
Builder
Builder type: External builder
Use default build command
Build command: make -C ${ConfigName}
Variables...
Makefile generation
Generate Makefiles automatically
Expand Env. Variable Refs in Makefiles
Build location
Build directory:
Workspace...
File system...
Variables...
Restore Defaults
Apply
OK
Cancel
BuildCMake

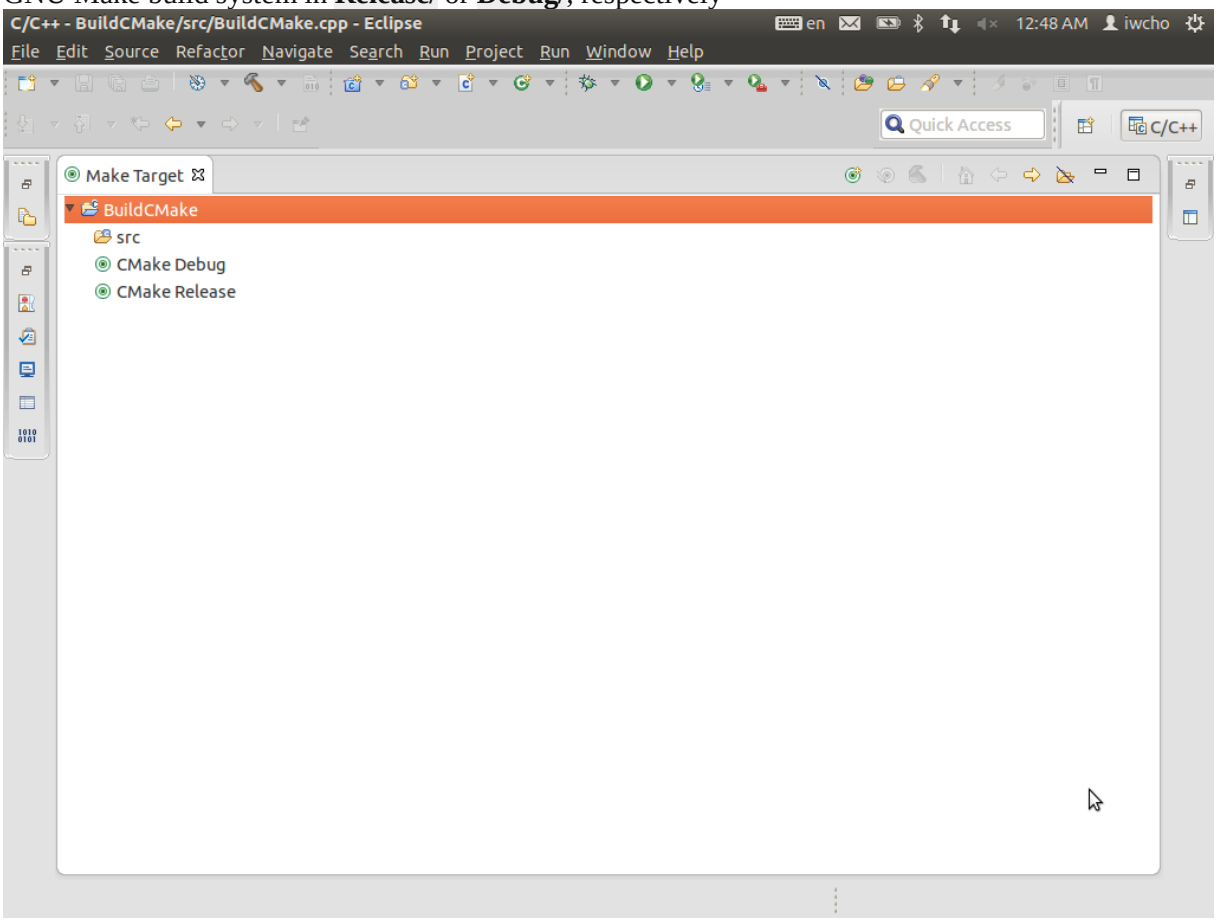

Eclipse
en 12:42 AM iwcho
File Edit Source Refactor Navigate Search Run Project Run Window Help
Properties for BuildCMake
type filter text
Resource
Builders
C/C++ Build
C/C++ General
Project References
Run/Debug Settings
Task Repository
WikiText
C/C++ Build
Configuration: Debug [Active]
Manage Configurations...
Builder Settings
Behaviour
Refresh Policy
Build settings
Stop on first build error
Enable parallel build
Use number of processors (2)
Use parallel jobs: 2
Use unlimited jobs
Workbench Build Behavior
Workbench build type:
Make build target:
Build on resource save (Auto build)
all
Variables...
Note: See Workbench automatic build preference
Build (Incremental build)
all
Variables...
Clean
clean
Variables...
Restore Defaults
Apply
OK
Cancel
BuildCMake

- Click *OK*

Build the Project

Use CMake to generate an out-of-source GNU Make build system:

- In the *Make Targets* window, double click on *CMake Release* or *CMake Debug* to generate the GNU Make build system in **Release/** or **Debug/**, respectively



- If necessary, edit your **CMakeLists.txt** control files
- Delete the contents of the corresponding build directory. For example:
  rm -r Release/*
  and repeat.

  Actually, for minor edits to your **CMakeLists.txt** control files, you need not delete the build directory. However, I cannot tell you exactly what the threshold for "minor edits" is.

Now, build the project the usual way with Eclipse:

- Select the configuration to build (*Release* of *Debug*) with the *Project* > *Build Configurations* > *Set Active* command
- Build with the *Project* > *Build Project* command
- Edit your source code files, and repeat

END