Building U-Boot in Eclipse Helios

Brad Stewart, Jan 31 2011

Background

Freescale application notes AN4171 and AN4173 describe how to extract u-boot from the LTIB package directory and compile them using command line make programs. This example uses Ubunt 10.04.

Appendix A describes how to set up a environment that allows you to use the popular Eclipse IDE to build u-boot. Unfortunately, the Eclipse example uses older version (Ganymede) and has a significantly different GUI from the latest version (Helios). This guide is designed to get you up and running with Helios.

For this example, I simply used the existing source tree and did not create a separate "custom" project as outlined in sections 1-7 in Chapter 3.

Using Command Line Make

It's a good idea to review and try out compiling U-Boot from the command line. This is explained in AN4171/3. This gives you a good idea of what Eclipse will try to emulate.

To compile u-boot using the command line, it is necessary to tell the system where the files and compiler resides. This is done using the commands:

```
$ export CROSS_COMPILE=arm-none-linux-gnueabi-

$ export

$ PATH=/opt/freescale/usr/local/gcc-4.1.2-glibc-2.5-nptl-3/arm-none-linux-gnueabi/bin/:$PATH
```

This works fine except that these environment values are only valid in the terminal session you define them in. Once you open another terminal window, or log back in, you have to type these commands again.

To make it 'permanent", add these to your /etc/environment file:

```
$ sudo gedit /etc/environment
```

The change will not take effect until you reboot or log out and log in. In my case, this is what my environment file looks like after making changes to /etc/environment:

```
$ brad@ubuntu:~$ cat /etc/environment
```

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/opt/freescale/usr/local/gcc-4.1.2-glibc-2.5-nptl-3/arm-none-linux-gnueabi/bin/"

CROSS_COMPILE="arm-none-linux-gnueabi-"
```

You can see what your environment is by typing

```
$ printenv
```

To see just the changes, use grep as shown:

```
$ printenv | grep CROSS
```
or
```
$ printenv | grep PATH
```

Setting up Helios to Compile U-Boot

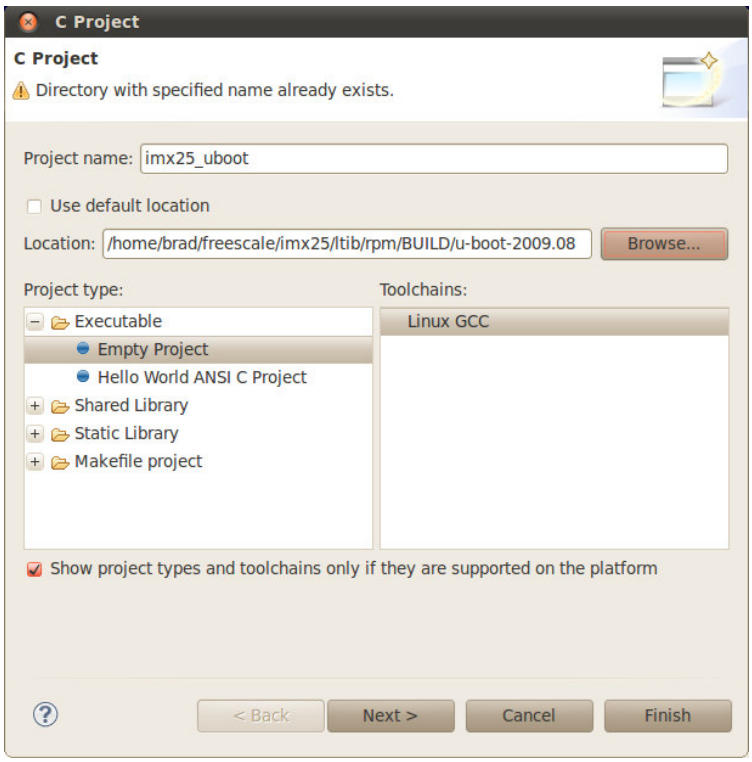
First thing is to install Eclipse. This is not difficult and is covered in another paper by the same author. Go to www.imxcommunity.org and do a search on Eclipse.

Step 1

Start by creating a new project. Click on File→New C Project

Add a project name (imx25_uboot in this example).

Uncheck the Use default location and use the Browse button to find the location of the uboot source (in this case /home/brad/freescale/imx25/ltib/rpm/BUILD/u-boot-2009.08



Click Finish

Wait until the Indexer has finished. Status is shown in the lower right side of the window. It may take a few minutes.

Step 2

Uncheck Project→Build Automatically on the top menu.

Step 3

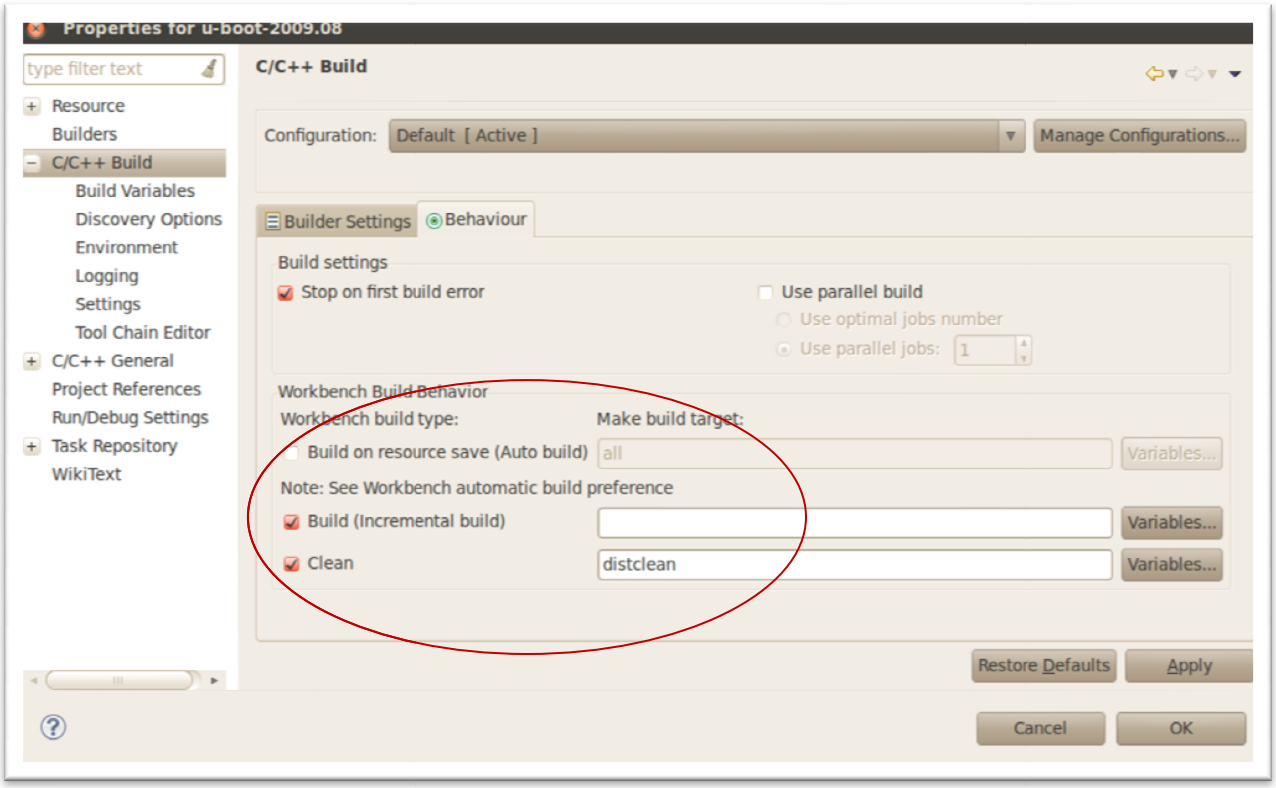
Go to Project→Properties (note that you need to high light the project in the project tree before this command becomes available).

In C/C++ Build, select the **Behaviour** tab.

Make the field under Build (Incremental build) blank. We need to do this otherwise the IDE will issue a "make all" command when you build the project.

Uncheck **Build on resource save (Auto build)**

Change clean to distclean otherwise it will issue the "make clean" command rather than the desired "make distclean".



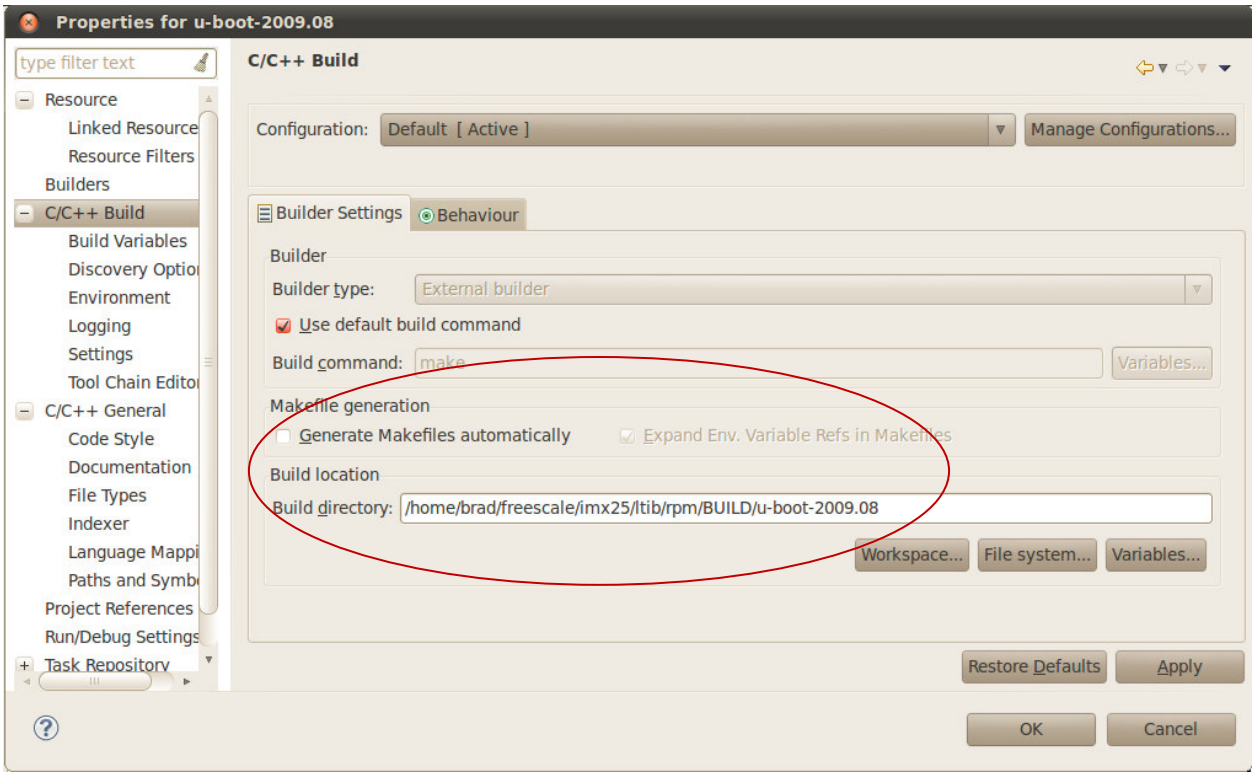
Click on the **Builder Settings** Tab

Add the Build directory pathname. In my case, I used the directory that the source file was copied to by LTIB using the command

$ ./ltib -m prep -p u-boot

This placed the source tree in /home/brad/Freescale/imx25/lbit/rpm/BUILD/u-boot-2009.08. Note that this directory may be different .

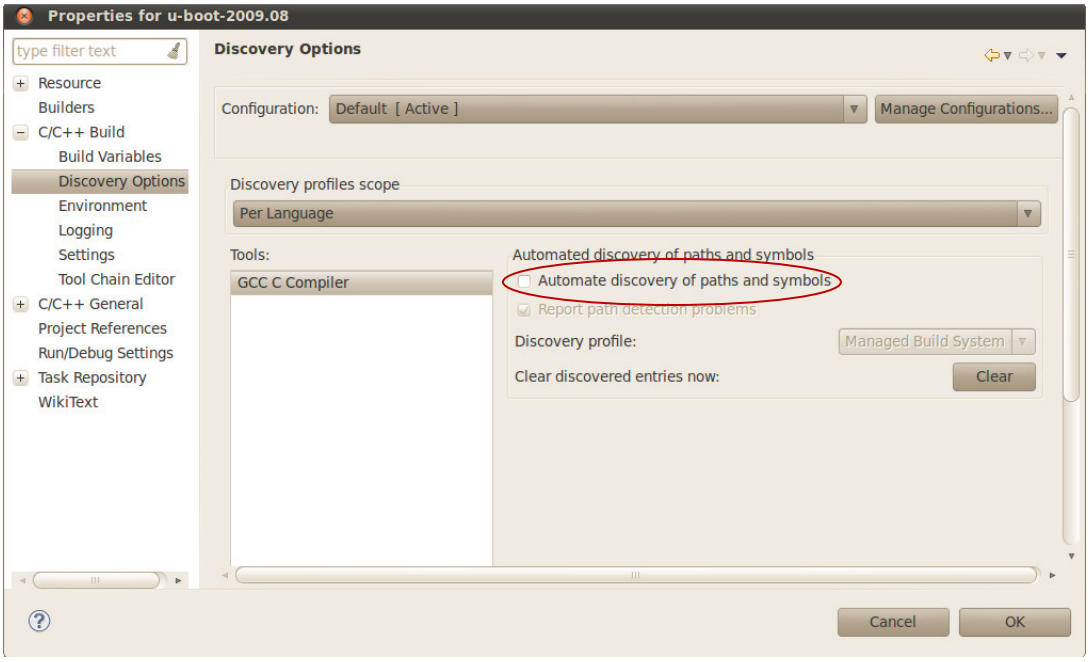
Uncheck **General Makefiles automatically** and add the build directory (click on Files system and navigate to where the source is). In my case, it's "/home/brad/Freescale/imx25/lbit/rpm/BUILD/u-boot-2009.08" and will probably be different from your setup.



Click on Apply

Step 4

Select Discover Options and uncheck **Automate discovery of paths and symbols** as shown below.



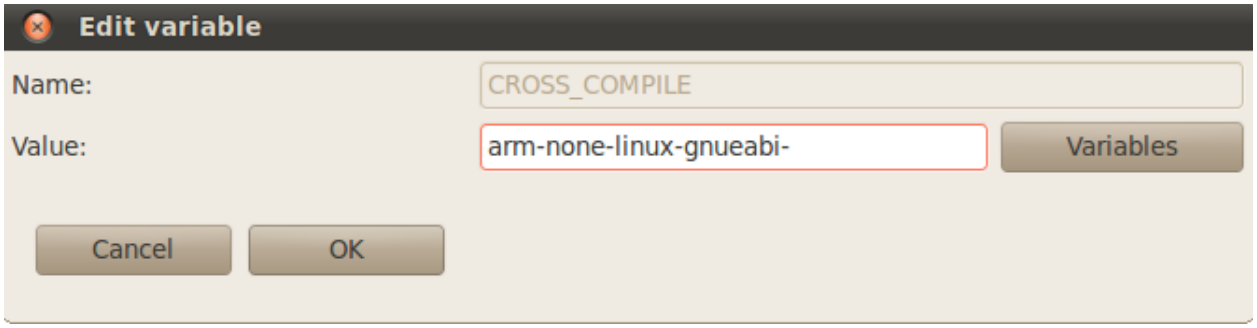
Next, we want to define environment variables that are passed to the make configureation file. These include PATH (to tell Eclipse where the files are) and CROSS_COMPILE (to tell Eclipse where the compiler is).

Step 5

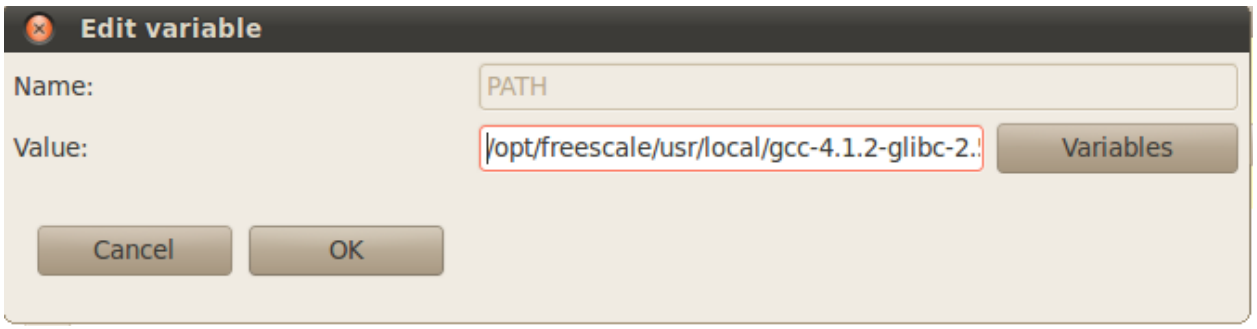
Click on Environment,

Undefine CWD and PWD. Add two new variables by clicking on Add…

Add the name CROSS_COMPILE and "arm-non-linux-gnueabi-" for the variable (don't use quotes).

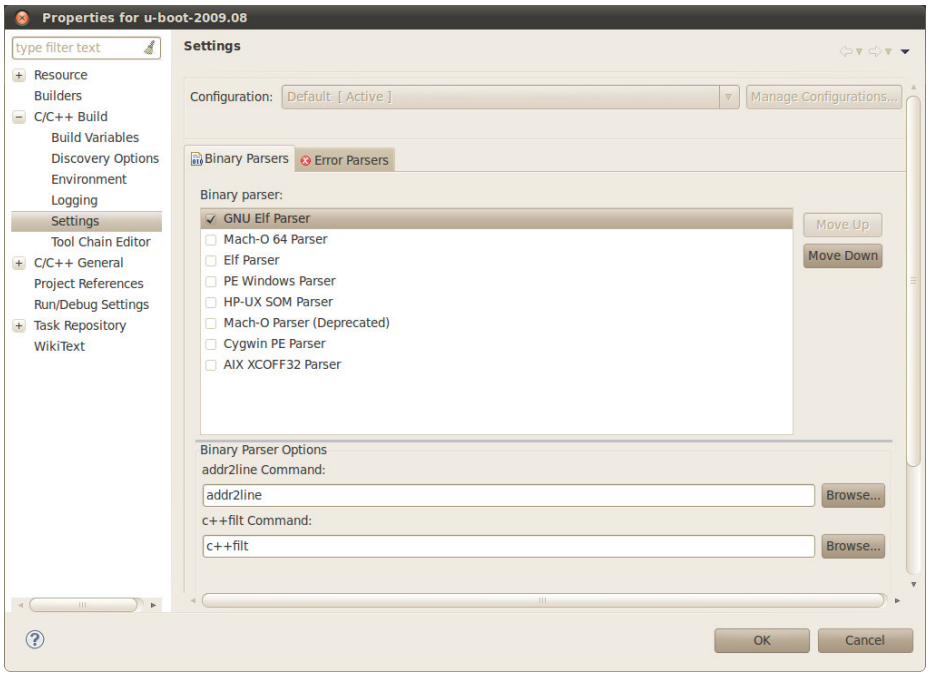


And define PATH with the following value:
"/opt/freescale/usr/local/gcc-4.1.2-glibc-2.5-nptl-3/arm-none-linux-gnueabi/bin/".
Note that this uses gcc-4.1.2. If you installed a later version of LTIB for iMX5x, then this line might read "opt/Freescale/usr/local/gcc-4.4.4-glibc-2.11.1-multilib-1.0/bin"



Step 6

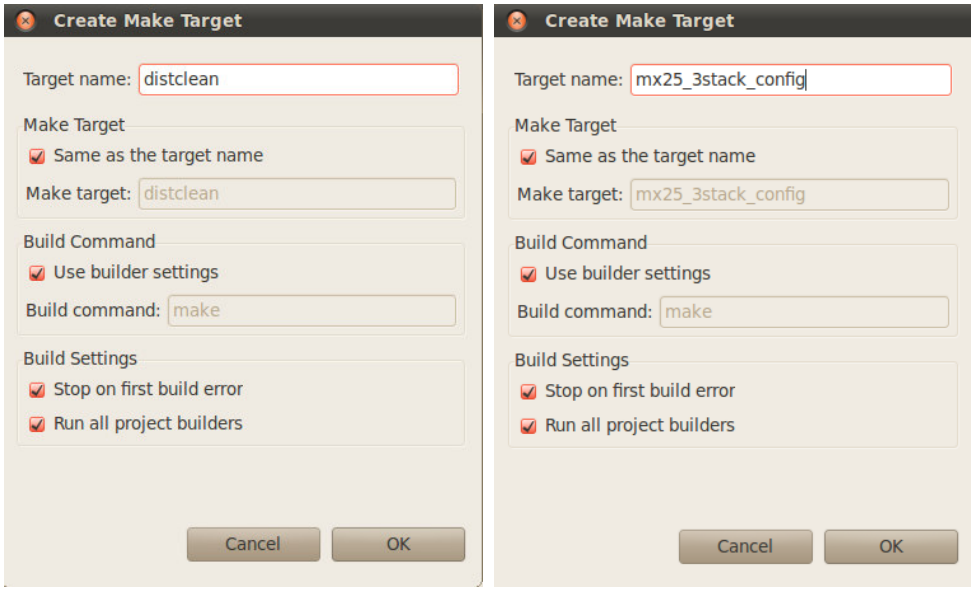
In Settings, slect GNU Elf Parser. You should see addr2line and c++filt



Click OK to get back to the C/C++ Perspective.

Step 7

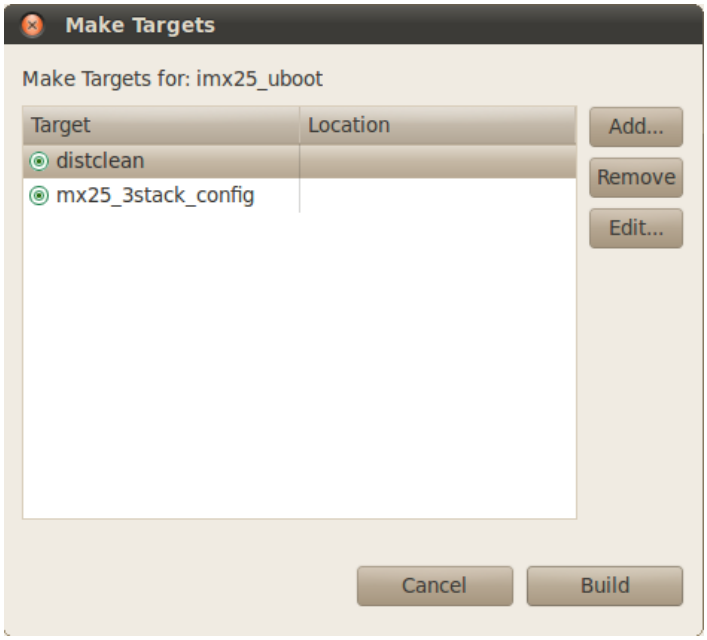
Create the Make targets. Click on Project→Make Target→Create or Project→Make Target→Build→Add…



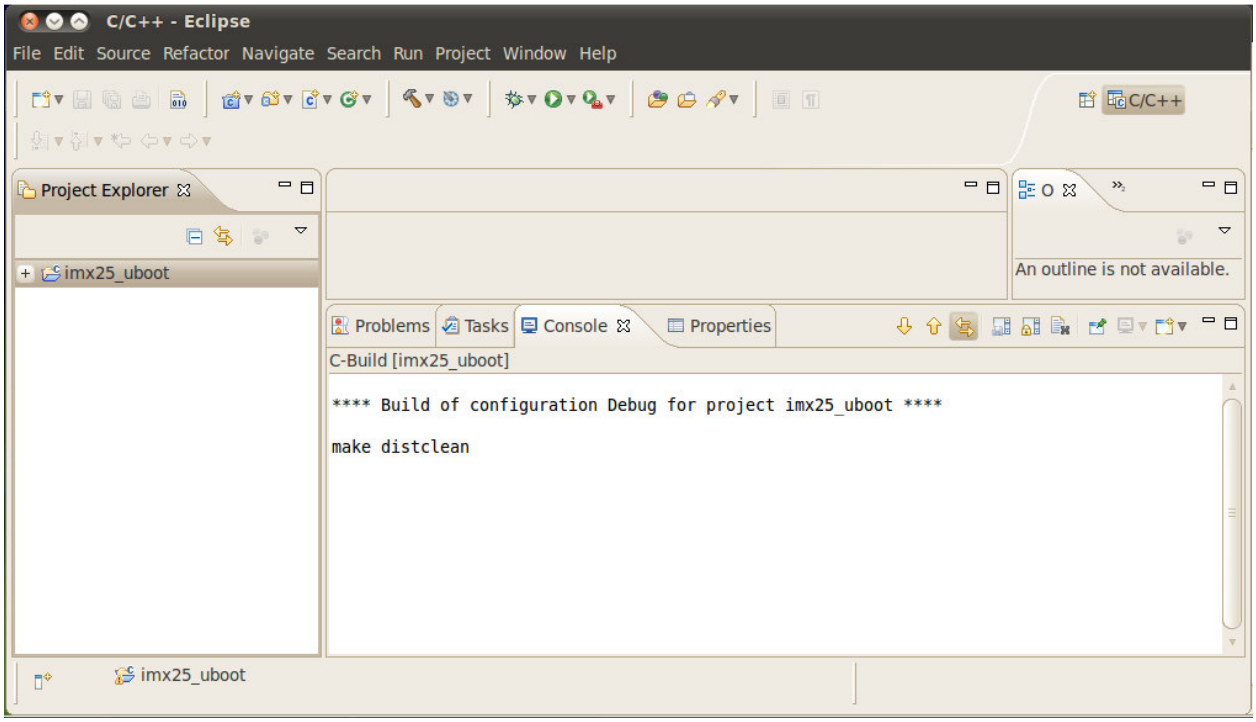
You'll need to do this twice for **distclean** and **mx25_3stack_config**.

Step 8

If the environment variables were done correctly as shown in Step 7, then if you click on Project→Make Target→Build, you should see the following window:

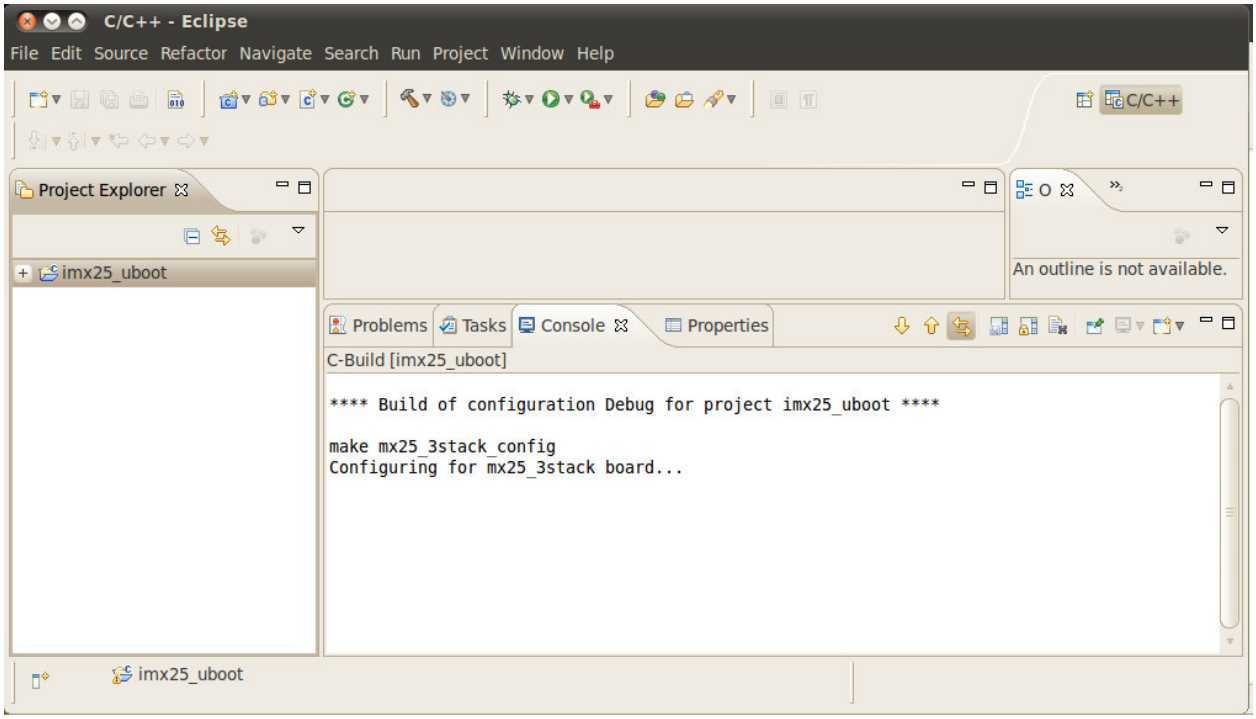


Highlight **distclean** and click on Build. You should see the following in the Console window:



This will clean the project and remove the u-boot binary and map files.

Repeat but this time, highlight **mx25_3stack_config** and click on Build. You should see the following in the Console window:



Step 9

Finally, build the project. Click on Project→Build Project. This may take a few minuites.
The result is shown below. Note the bin and map files are added to the project tree.