- Eclipse 3.7 Indigo
  - Integrated Development Environment (IDE)
  - has a plugin architecture to add features
    - support for C development is via a plugin, **CDT**
  - can use the gcc compiler and gdb debugger
    - and Makefiles
  - Requires a Java Runtime Environment

  http://www.eclipse.org/downloads
    - Eclipse IDE for C/C++ Linux Developers

  http://www.eclipse.org/cdt/

DANGER!

- Does Eclipse run on Windows?
  - Yes
- Can I write C code on Windows?
  - Yes, with the Cygwin suite installed
- Can I write C code on Windows for this class?
  - No

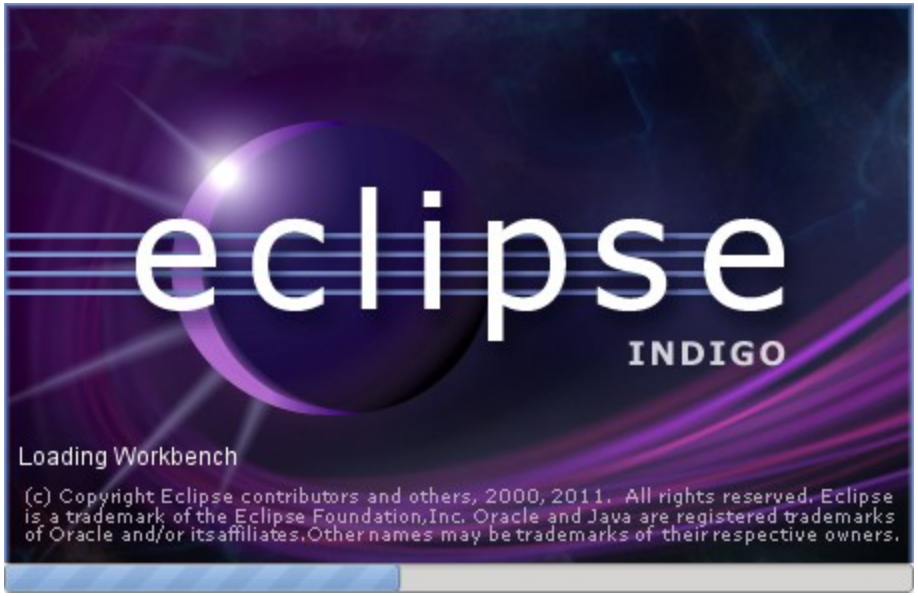
Let's make an Icon

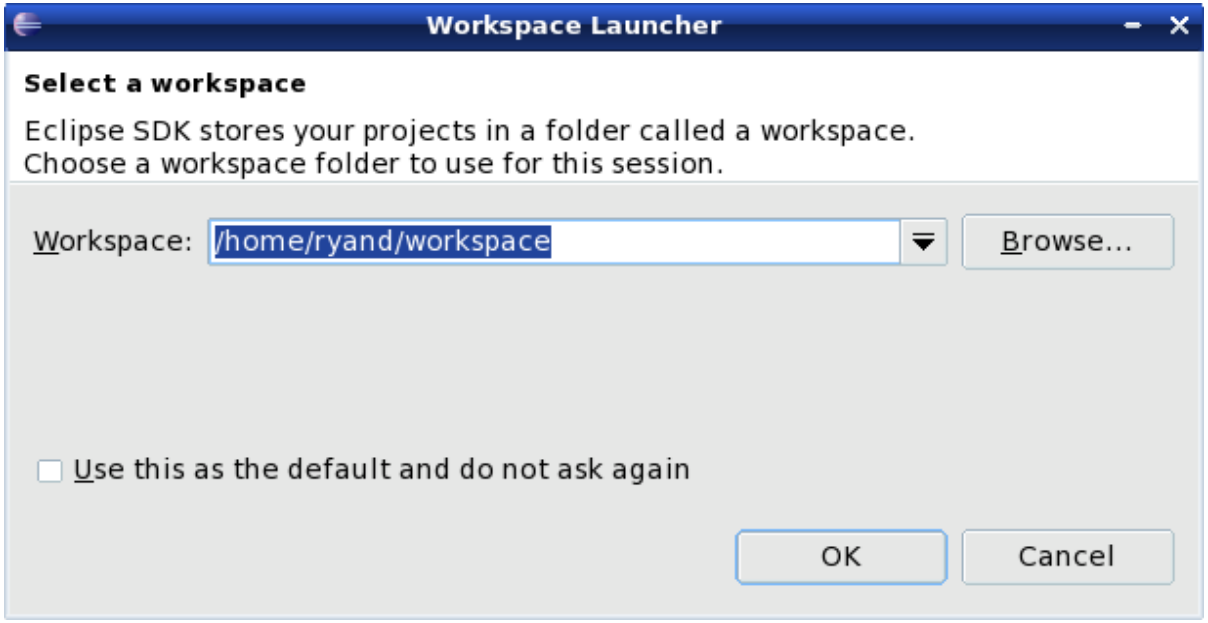
- Where is the executable eclipse?
  /usr/local/share/eclipse/eclipse

  not in your $PATH

- And then start Eclipse

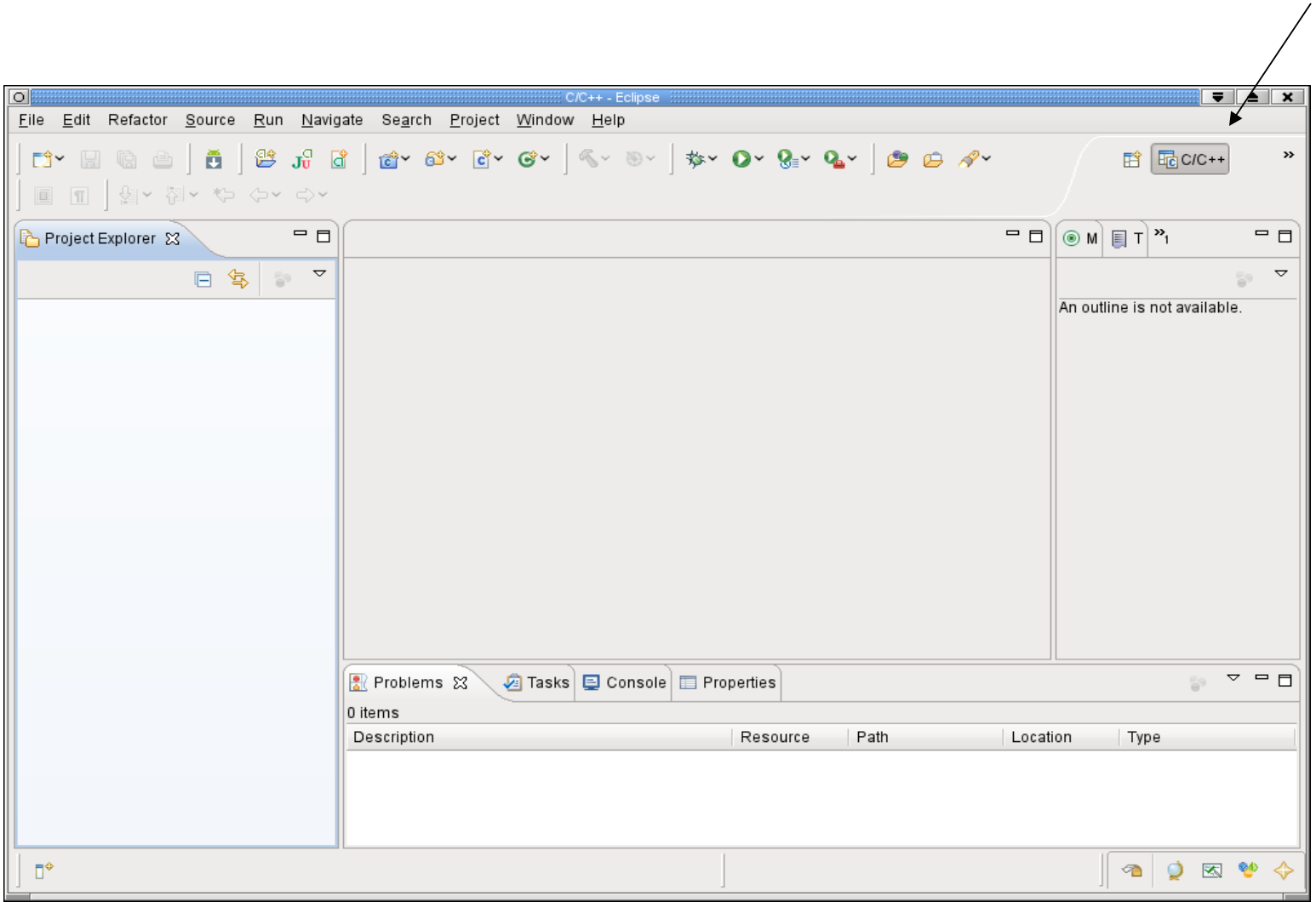
Workspace Launcher





Select the perspective for coding

Make sure the perspective is C/C++ not Java



Create a new H**elloWorld** project

File → New → C Project

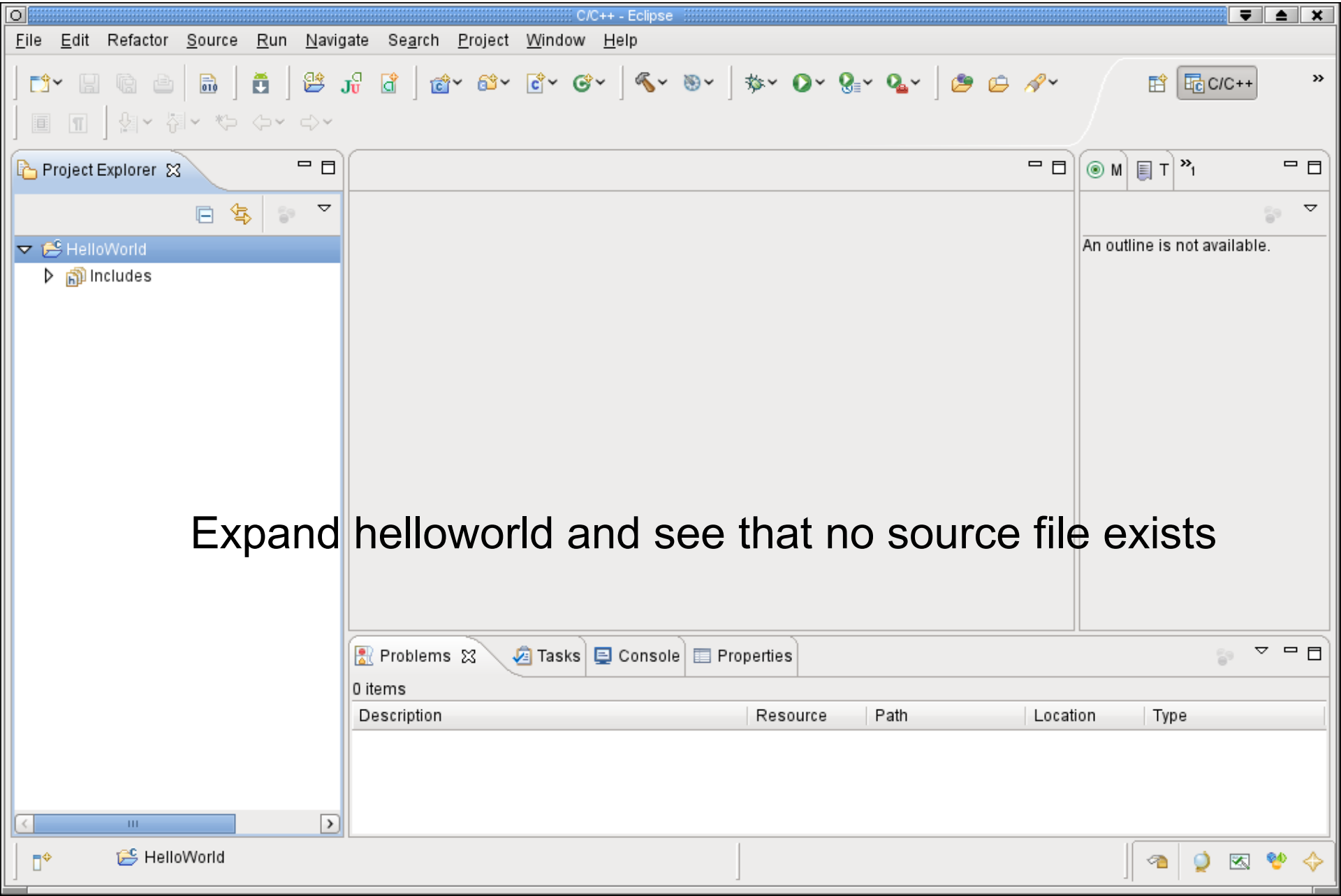
Makefile Project → Empty Project → Linux GCC

Then click Next

Advanced Settings → C/C++ Build → Uncheck “Generate Makefiles Automatically” → OK

Then Finish

A HelloWorld project



Expand helloworld and see that no source file exists

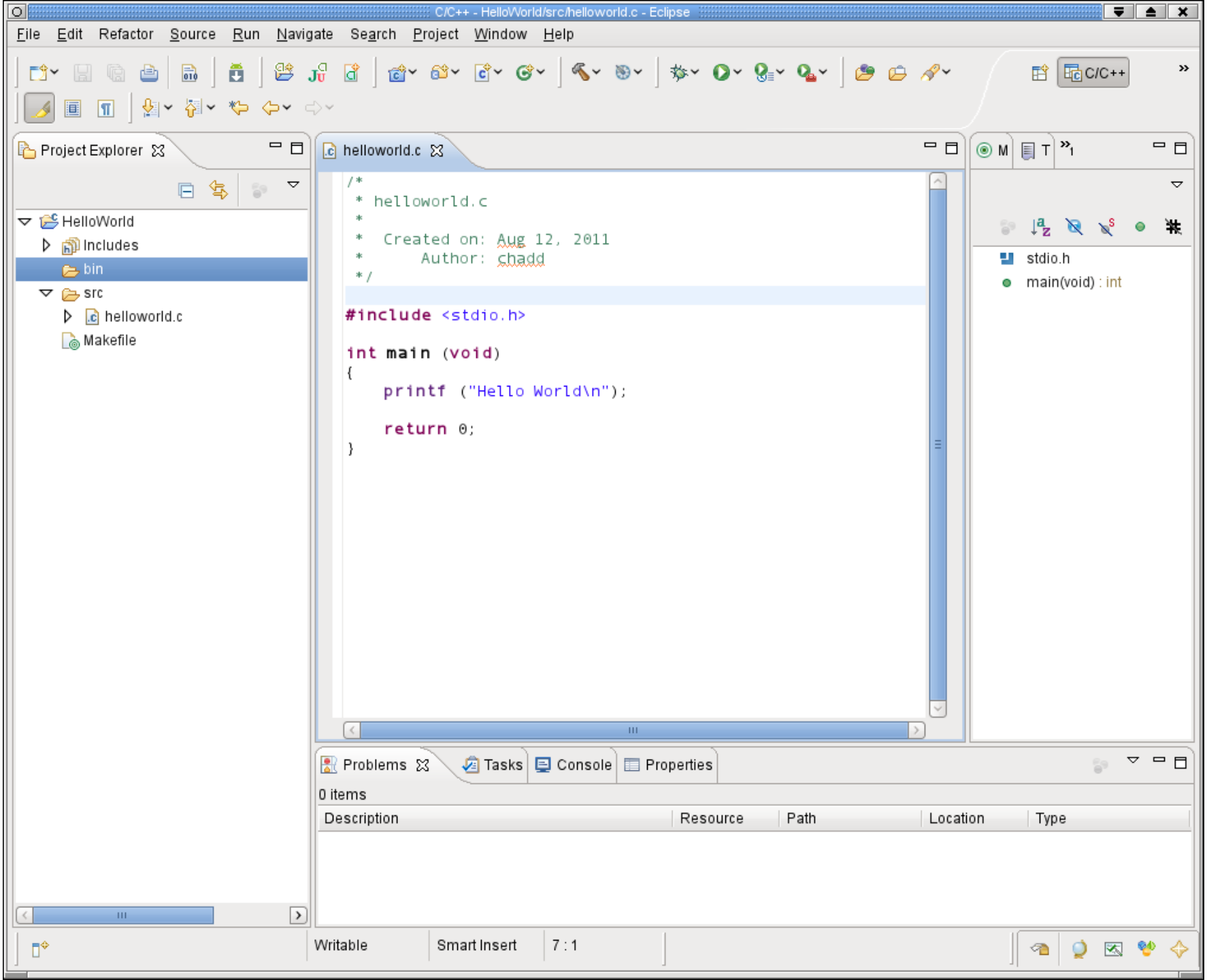
Create helloworld.c?

In HelloWorld project, create a folder called **src**
Create a folder called **bin**
Then create a new Source File called **helloworld.c** in the source folder

```
#include <stdio.h>

int main (void)
{
   printf ("Hello World\n");

   return 0;
}
```


C/C++ - HelloWorld/src/helloworld.c - Eclipse
File Edit Refactor Source Run Navigate Search Project Window Help
C/C++
Project Explorer
HelloWorld
Includes
bin
src
helloworld.c
Makefile
helloworld.c
/*
* helloworld.c
*
* Created on: Aug 12, 2011
* Author: chadd
*/
#include <stdio.h>
int main (void)
{
printf ("Hello World\n");
return 0;
}
M
T
stdio.h
main(void) : int
Problems
Tasks
Console
Properties
0 items
Description
Resource
Path
Location
Type
Writable
Smart Insert
7 : 1

Create a Makefile

You need to create a file called Makefile in the helloworld folder. A Makefile specifies rules of how the executable is to be created.

Right Click HelloWorld → New → File
Call the file **Makefile**

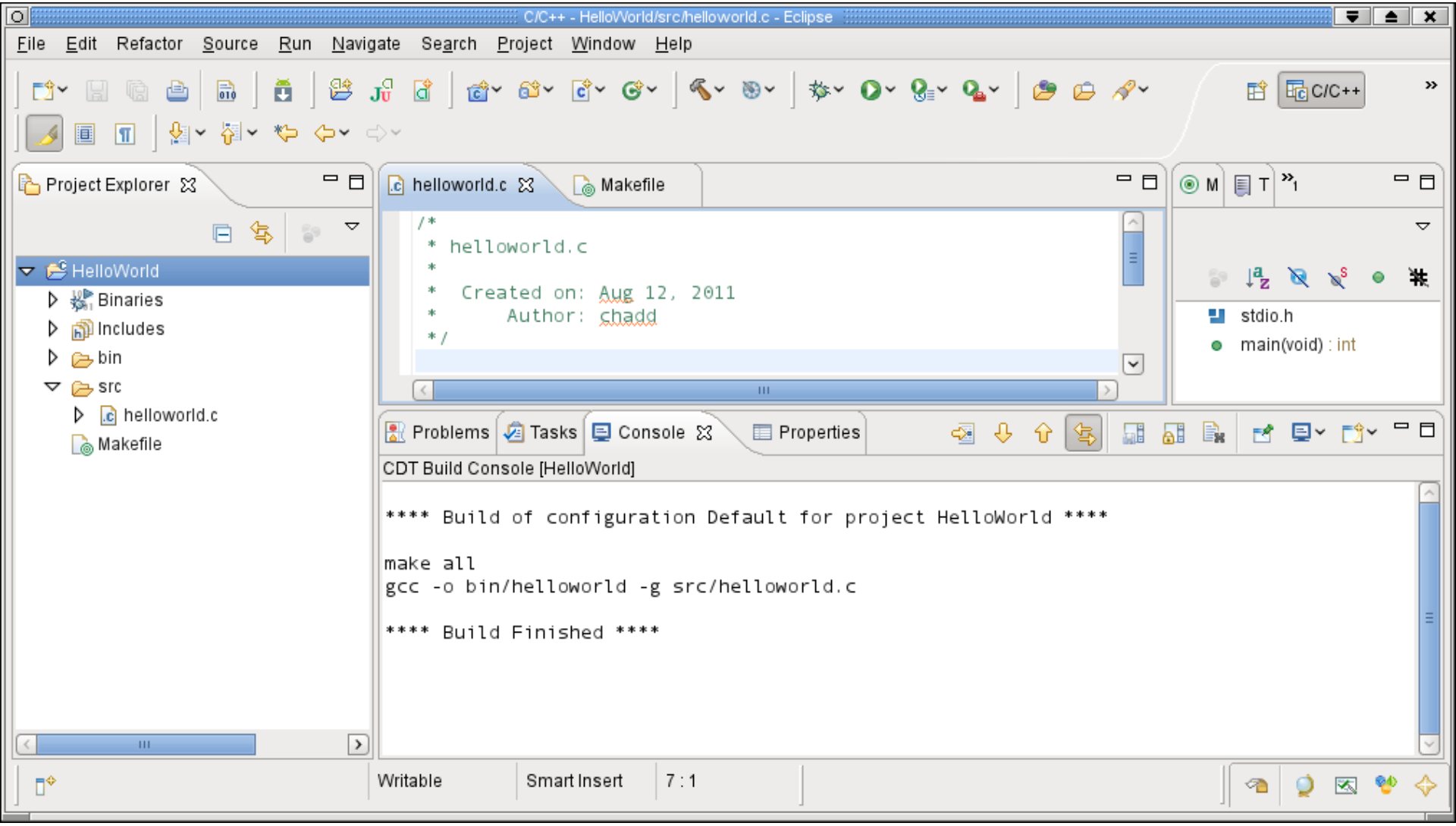
The makefile text must be (a single tab character precedes gcc):

```
all: bin/helloworld

bin/helloworld: bin/helloworld.o
   gcc -o bin/helloworld bin/helloworld.o ${CFLAGS}

bin/helloworld.o: src/helloworld.c
   gcc -o bin/helloworld.o -c src/helloworld.c ${CFLAGS}
```

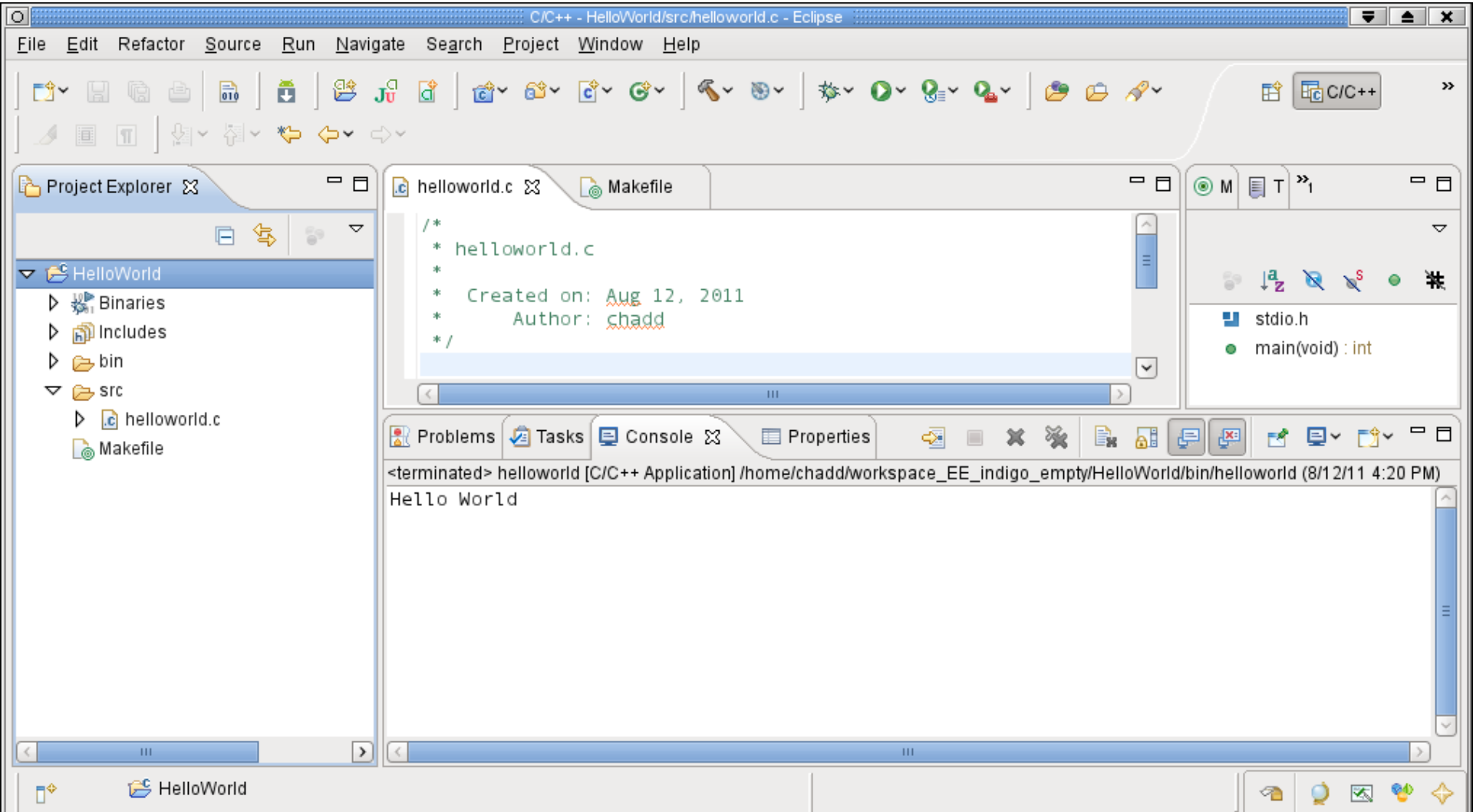
How to build your project ?



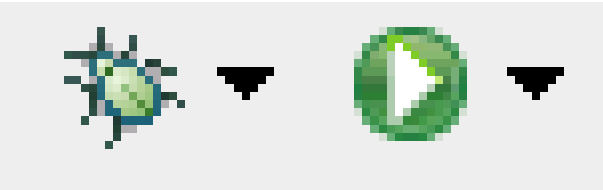
Click on HelloWorld, then Project → Build Project

How to run your program?

Then right click on helloworld and Run As → Local C/C++ Application choose gdb/mi if given the option



Run versus Debug



Debug Run

Printing

- Windows | Preferences
- General | Appearance | Colors and Fonts
- C/C++ | Editor | C/C++ Editor Text Font
  - Use Courier 10 Pitch, Size 8
- This changes the font on the screen!
  - You may want to change back after printing
- Print doubled sided!

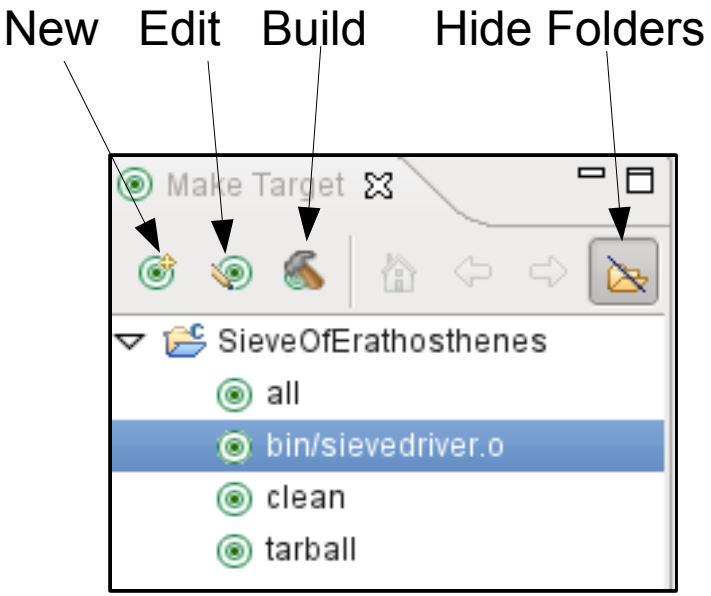
Coding Standards

- Download CodingStandardsProfile
  - untar them!
  - CS300PrefsF11.xml
  - sets tabs, newlines, spacing
  - Does not fix everything!
- Window | Preferences | C/C++ | Code Style | Import
- Open your .c file
  - Source | Format
  - Shift+Control+F

Add Make Target

- Open Makefile
- Open Make Target panel
  - Window | View | Make Target
- New Make Target
  - type name of existing target in the Makefile

This is just for you convenience.



Helpful Commands

- F3 while cursor on function call
  - go to that function
- Control-L
  - go to line
- Control-A
  - select all
- Control-I
  - correct indentation

Be sure to look through the Source and Navigate menu!

C Topics

- #ifdef / #ifndef
- #define
- static
- array
- include ""

```
#ifndef _EXAMPLE_
#define _EXAMPLE_

#include “localHdr.h”
#define ARRAYSIZE 1024
#define TRUE 1

static int value;
int bigArray[ARRAYSIZE];
#endif
```

C Topics

```
void foo(int arr[], int len, char *str)
{
   int index = 0;
   for( ; index < len ; index ++)
   {
      printf("%d\t", arr[index]);
   }
    printf("%s\n", str);
}
// the function call
foo(array, ARRAY_SIZE, "the message");
```

Other tips

- Window | Preferences
  - search for template to setup .c and .h file templates
    - you can add the file comment header automatically!
  - search for margin
    - set the print margin column to 80!
  - search for name style
    - to set naming conventions
  - search for code analysis
    - setup error/warnings in code style

Source dir vs dir