Getting the raytracer up and running

Table of Contents

Microsoft Visual C++ 2005 Express Edition

Step 0: Installing the IDE

This IDE can be obtained free of charge from the Microsoft website. To install, please follow steps 1-3 as described on the following page:
http://msdn.microsoft.com/vstudio/express/visualc/usingpsdk/default.aspx

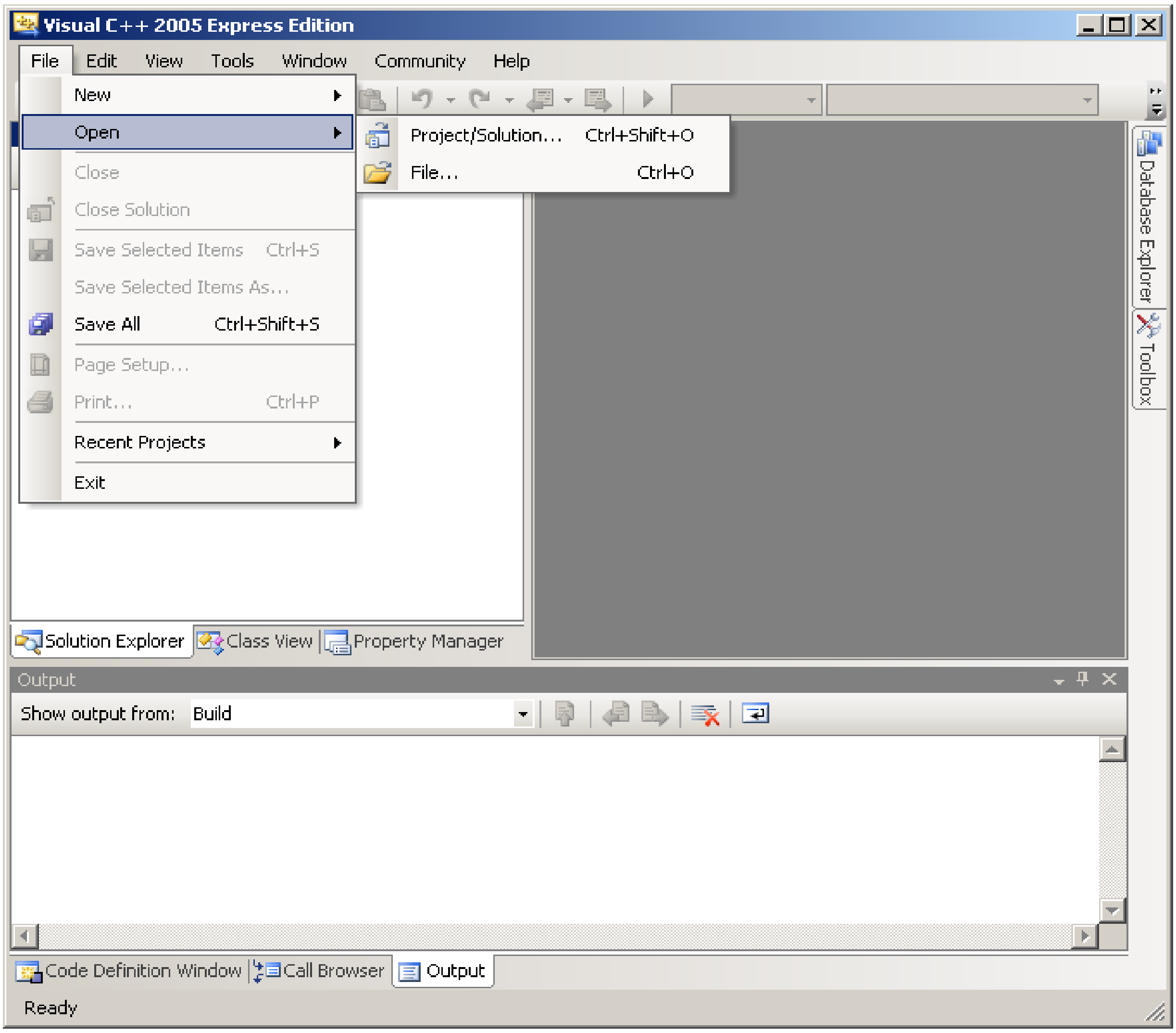
Note that the IDE along with the Microsoft Platform SDK will be ~400MB to download.

Step 1: Unzip the project files

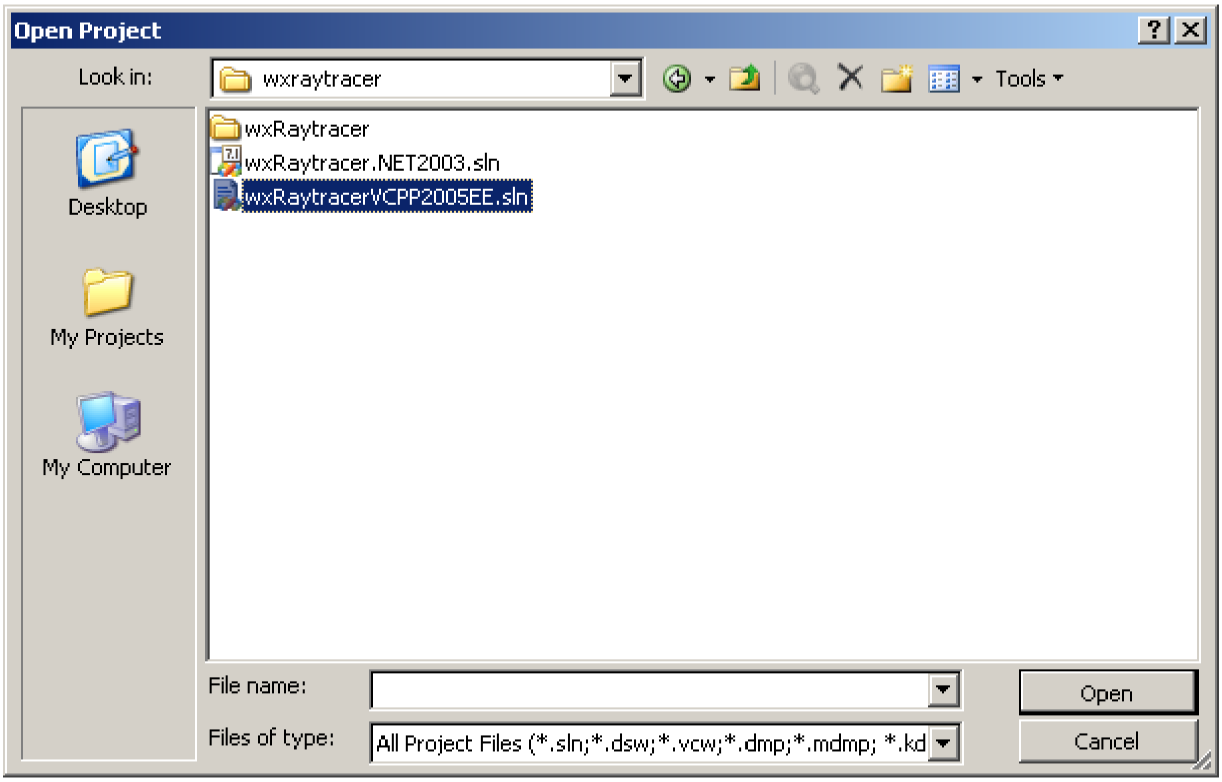
The project files are compressed as a zip archive. To decompress the files, a zip file archiver tool is required. If you don't already have one on your system, a powerful open source tool which will do the trick can be downloaded from http://www.7-zip.org

Step 2: Open the project in Visual C++

Select “File” > “Open” > “Project/Solution...” or hit Ctrl+Shift+O to open the “Open Project” dialogue.



Navigate to the folder where you unzipped the project files and open the file “wxRaytracerVCPP2005EE.sln”.

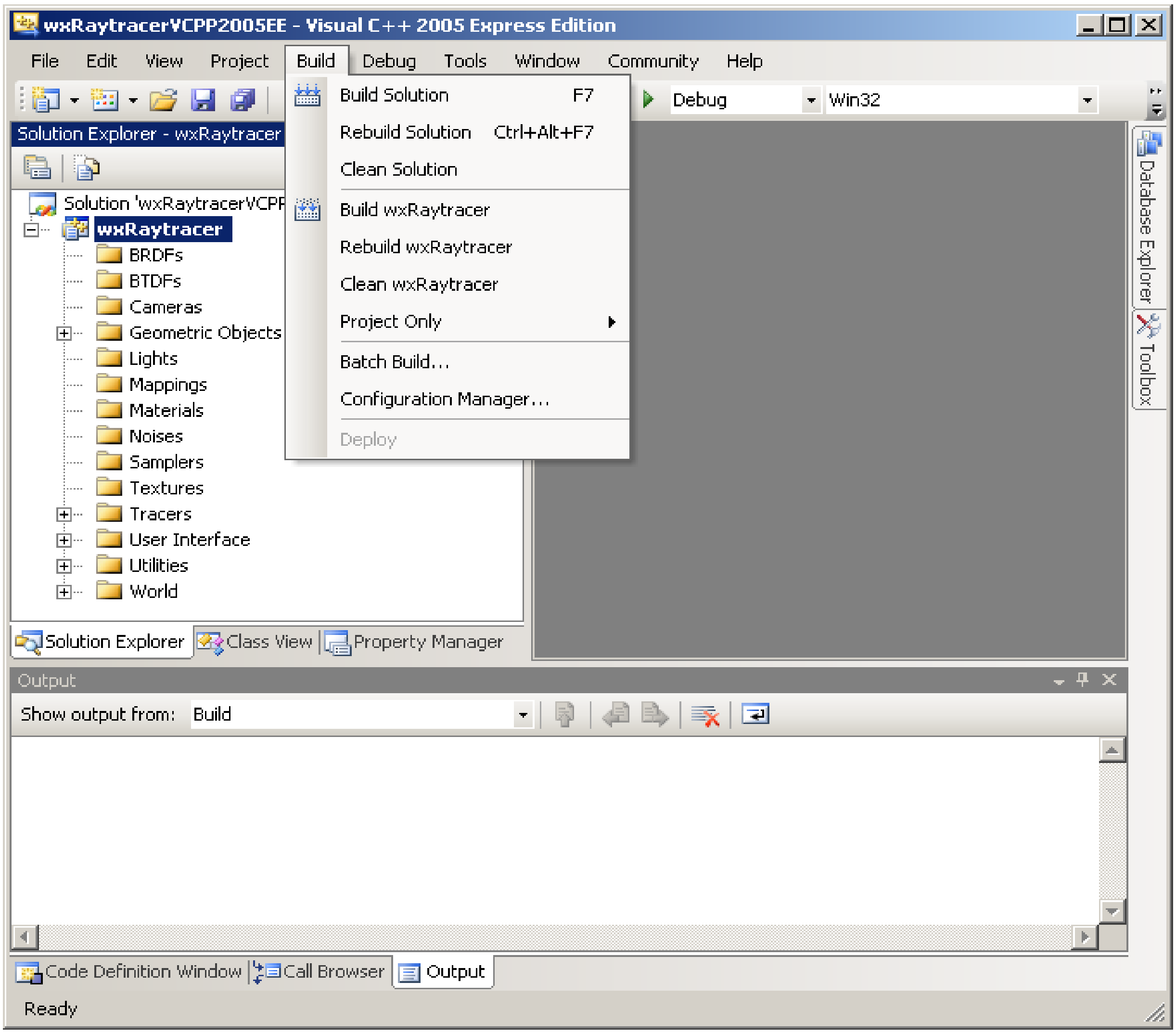


Look in: wxraytracer

Files of type: All Project Files (*.sln;*.dsw;*.vcw;*.dmp;*.mdmp; *.kd

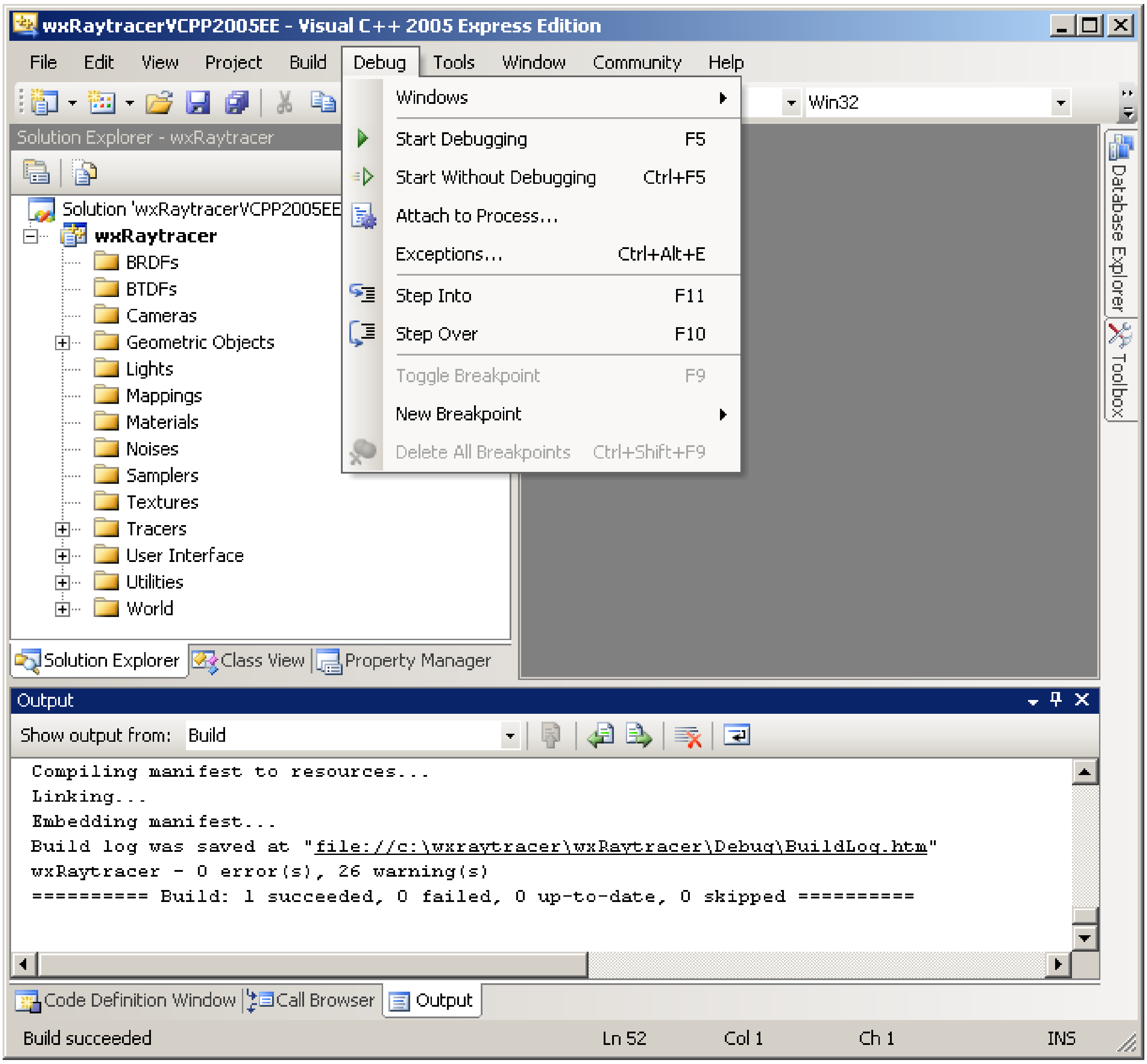
Step 3: Compile the project

Select “Build” > “Build Solution” or hit F7 to invoke the compiler. The output from the compiler will be displayed in the “Output” window at the bottom.



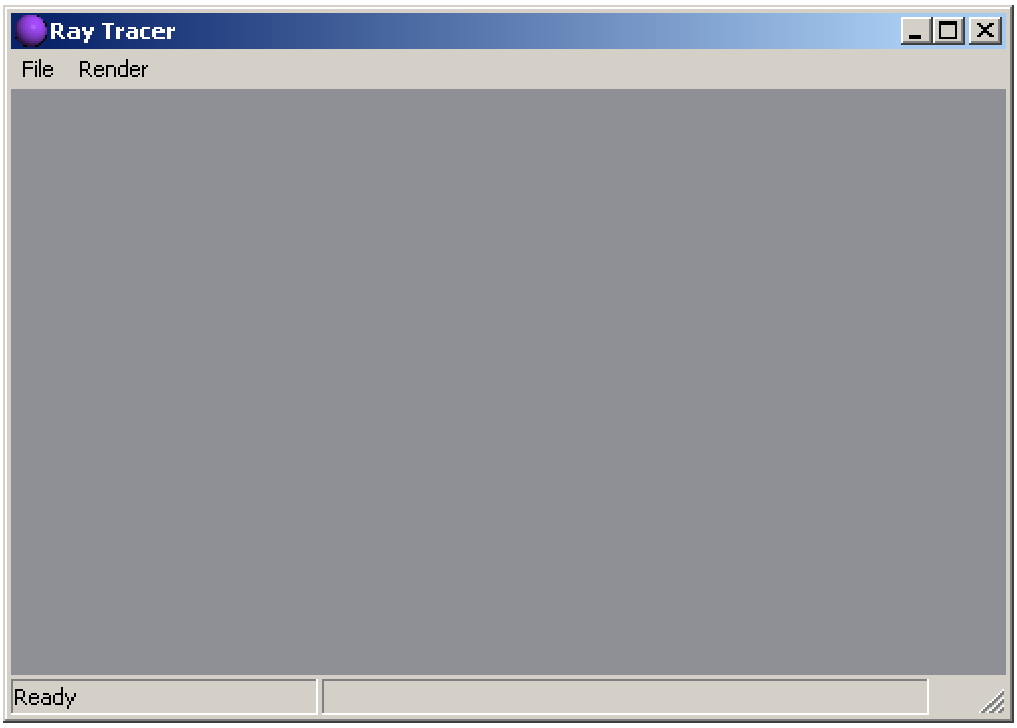
Step 4: Execute the ray tracer

Select “Debug” > “Start Without Debugging” or hit Ctrl+F5 to start the ray tracer.



```
Compiling manifest to resources...
Linking...
Embedding manifest...
Build log was saved at "file://c:\wxraytracer\wxRaytracer\Debug\BuildLog.htm"
wxRaytracer - 0 error(s), 26 warning(s)
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

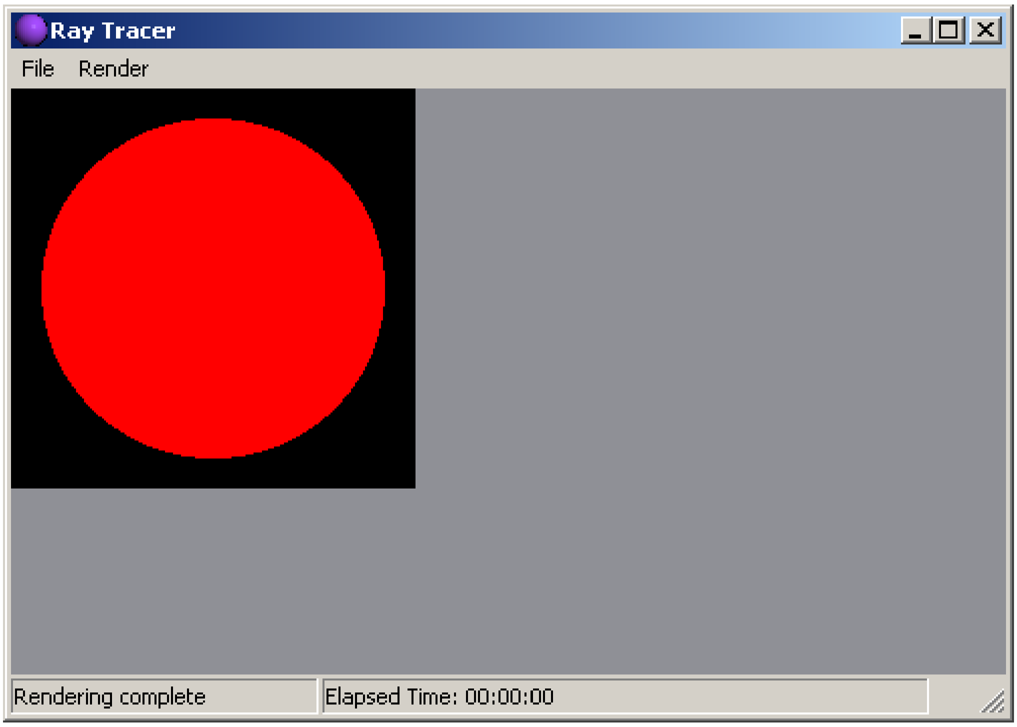
The ray tracer window will open.



Ready

Step 5: Render your first image

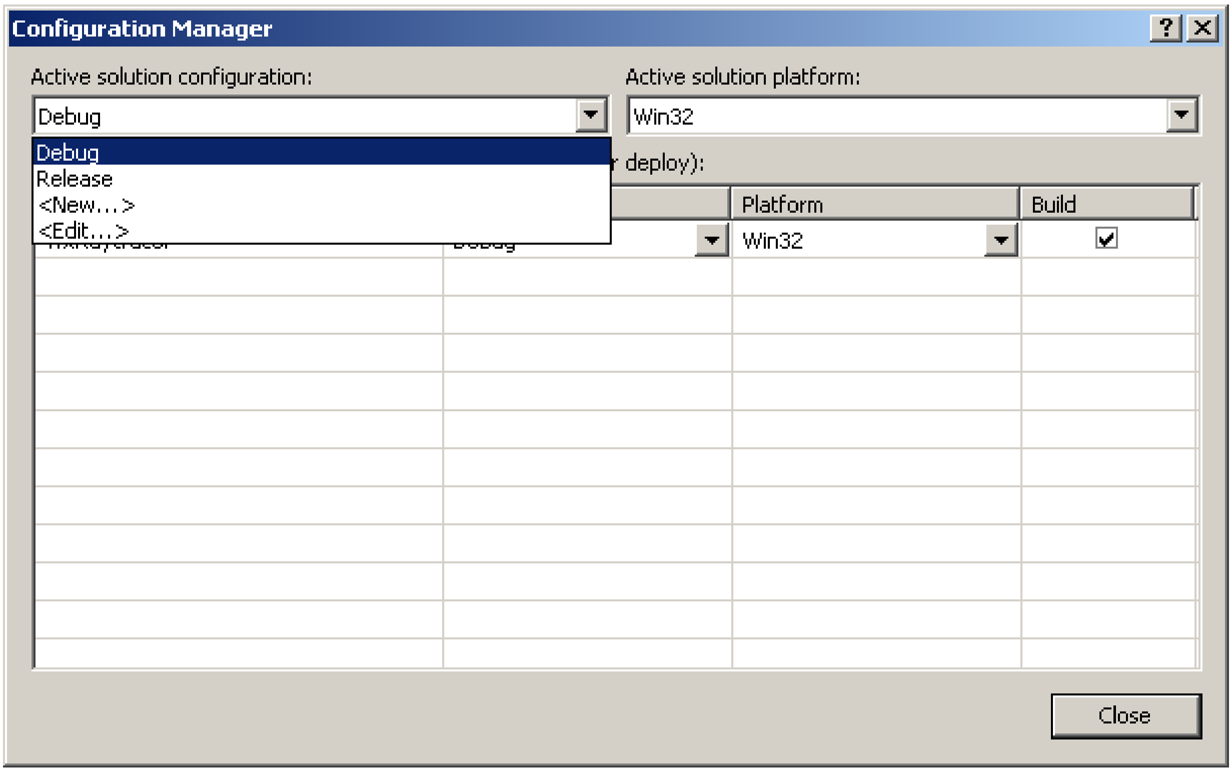
Select “Render” > “Start” from the menu to start rendering.



Rendering complete

Elapsed Time: 00:00:00

Note: the project has two different build configurations; debug and release. While debugging or rendering less complex images you might want to use the debug configuration, but when rendering larger and complex images you should change the active configuration to release. The reason for this is that the executable produced by the release build configuration will execute much faster and thus result in your image rendering faster. You can change the active configuration using the Configuration Manager which is opened by selecting “Build” > “Configuration Manager...”.



Configuration Manager

Active solution configuration: Debug

Active solution platform: Win32

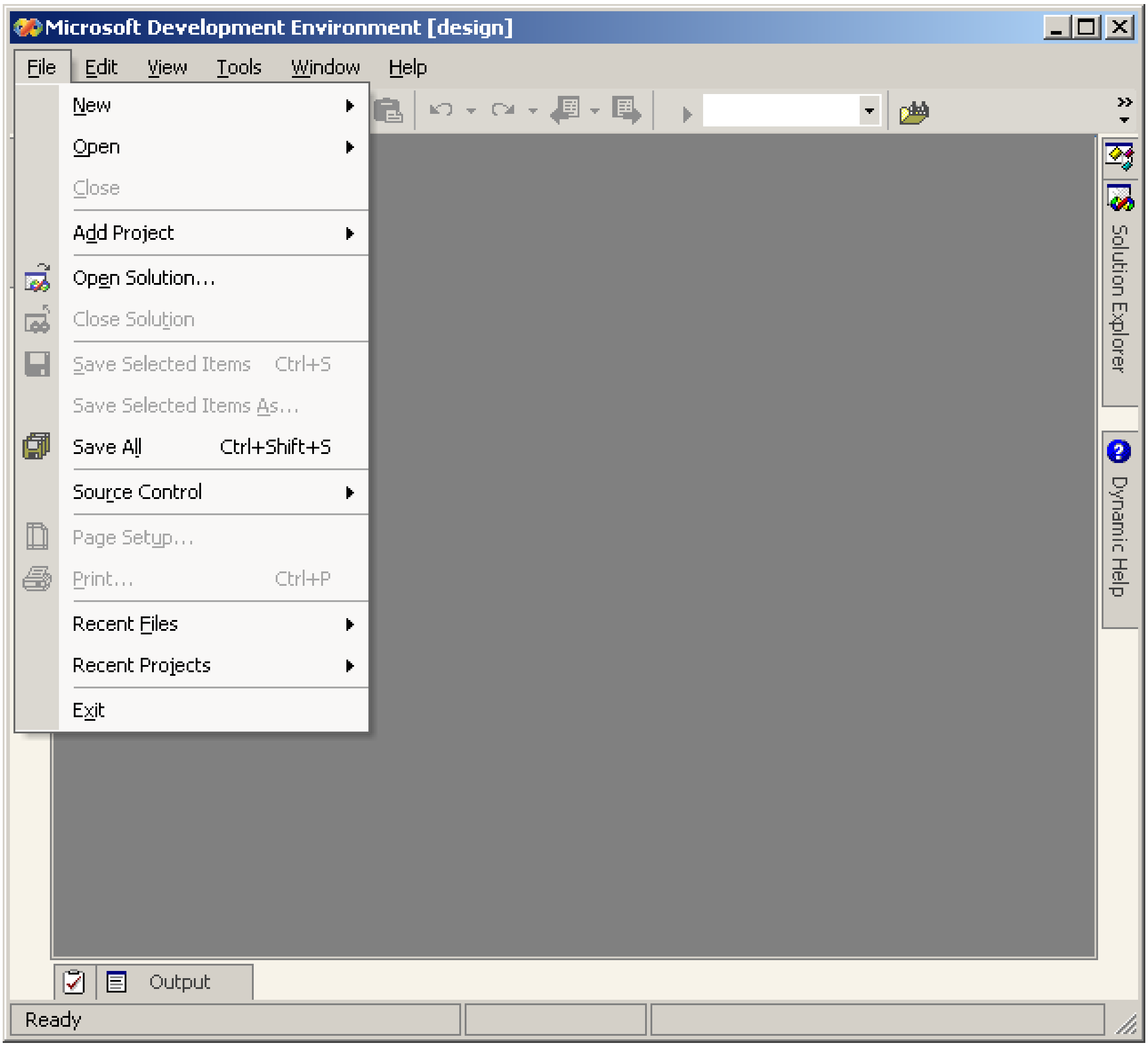
Microsoft Visual Studio .NET 2003

Step 1: Unzip the project files

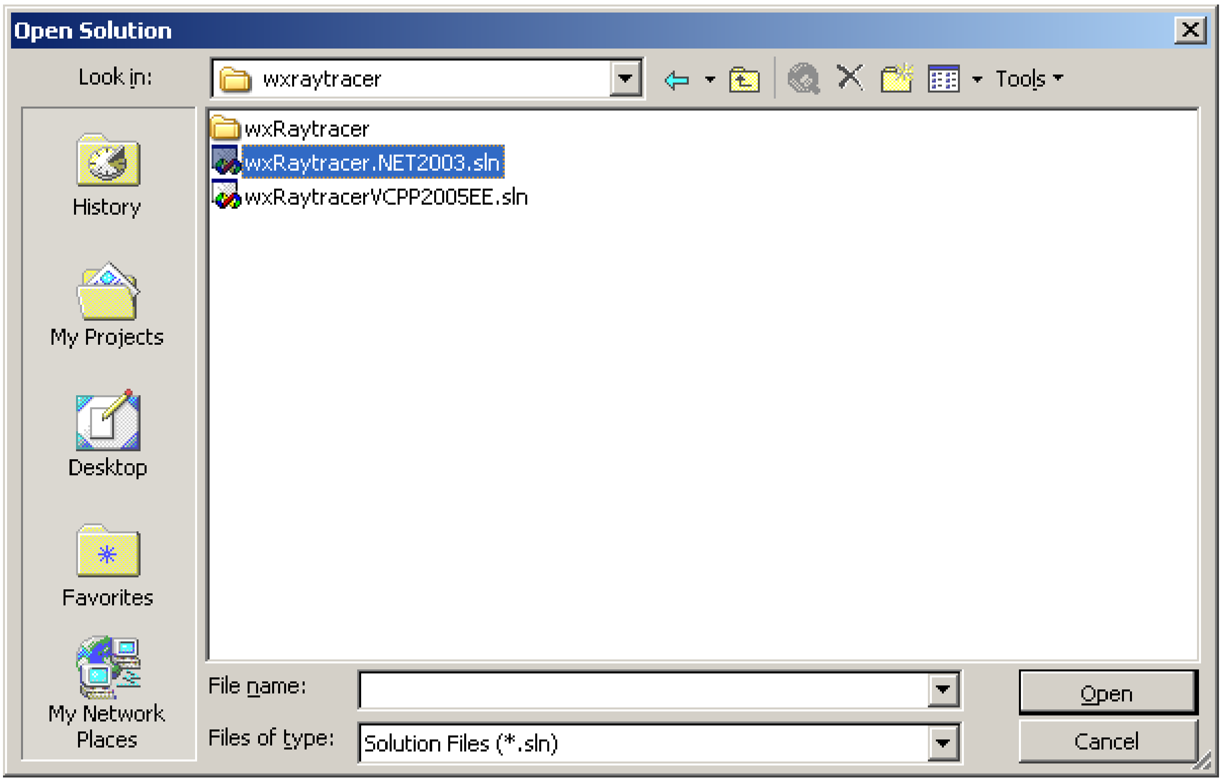
The project files are compressed as a zip archive. To decompress the files, a zip file archiver tool is required. If you don't already have one on your system, a powerful open source tool which will do the trick can be downloaded from http://www.7-zip.org

Step 2: Open the project in Visual Studio

Select “File” > “Open Solution...” to open the “Open Solution” dialogue.



Navigate to the folder where you unzipped the project files and open the file “wxRaytracer.NET2003.sln”.



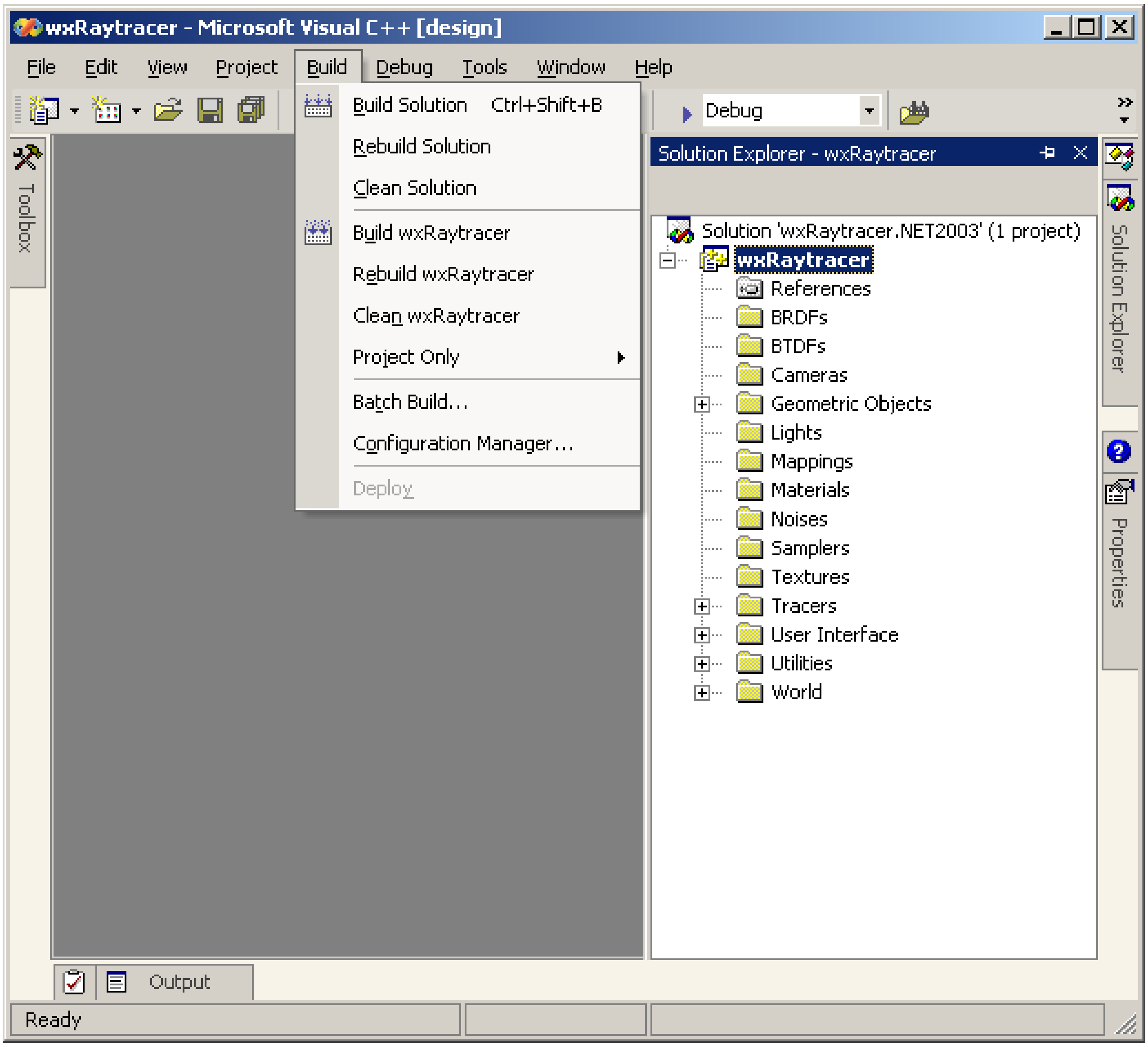
wxRaytracer
wxRaytracer.NET2003.sln
wxRaytracerVCPP2005EE.sln

File name:

Files of type: Solution Files (*.sln)

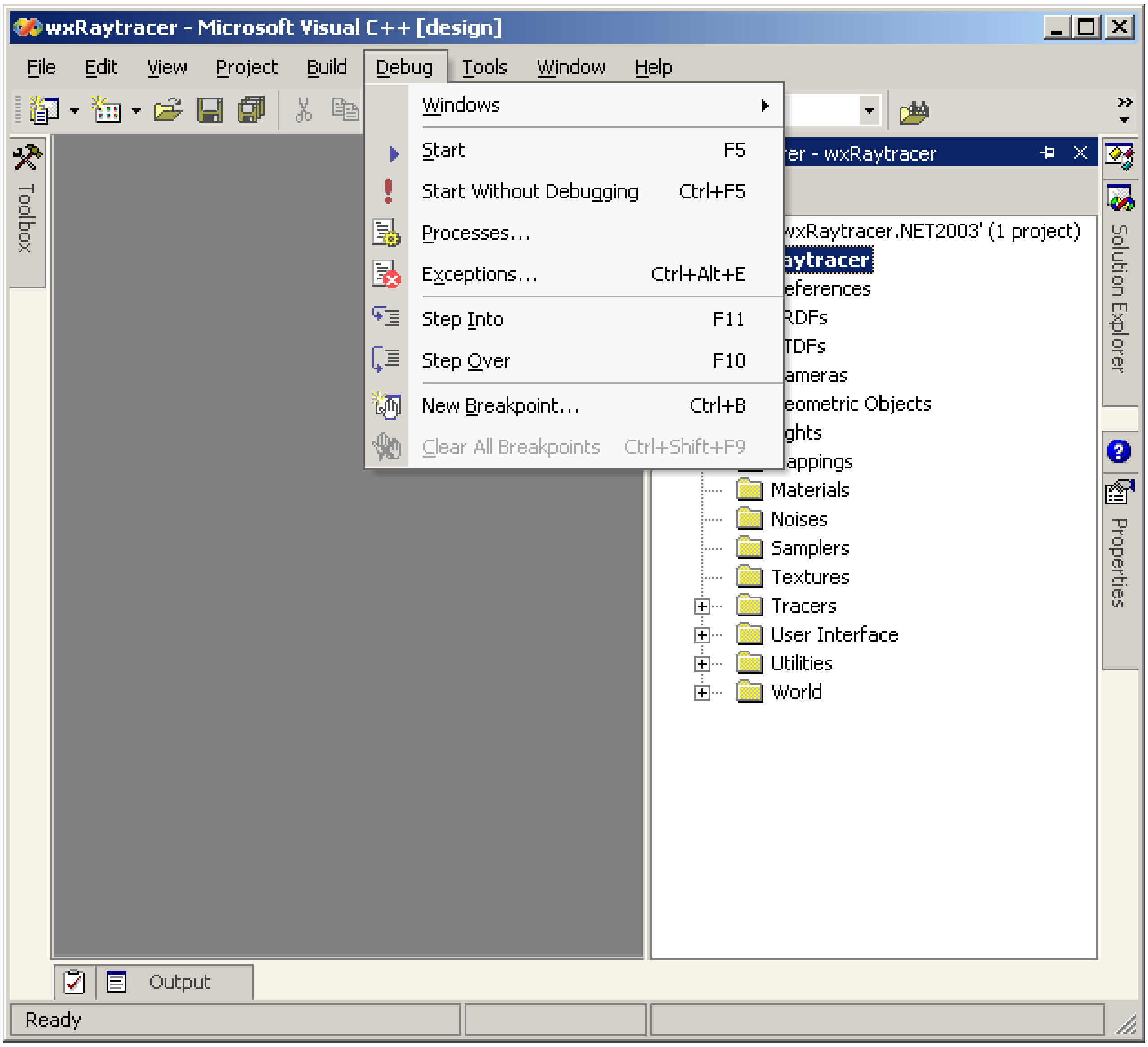
Step 3: Compile the project

Select “Build” > “Build Solution” or hit Ctrl+Shift+B to invoke the compiler. The output from the compiler will be displayed in the “Output” window at the bottom.

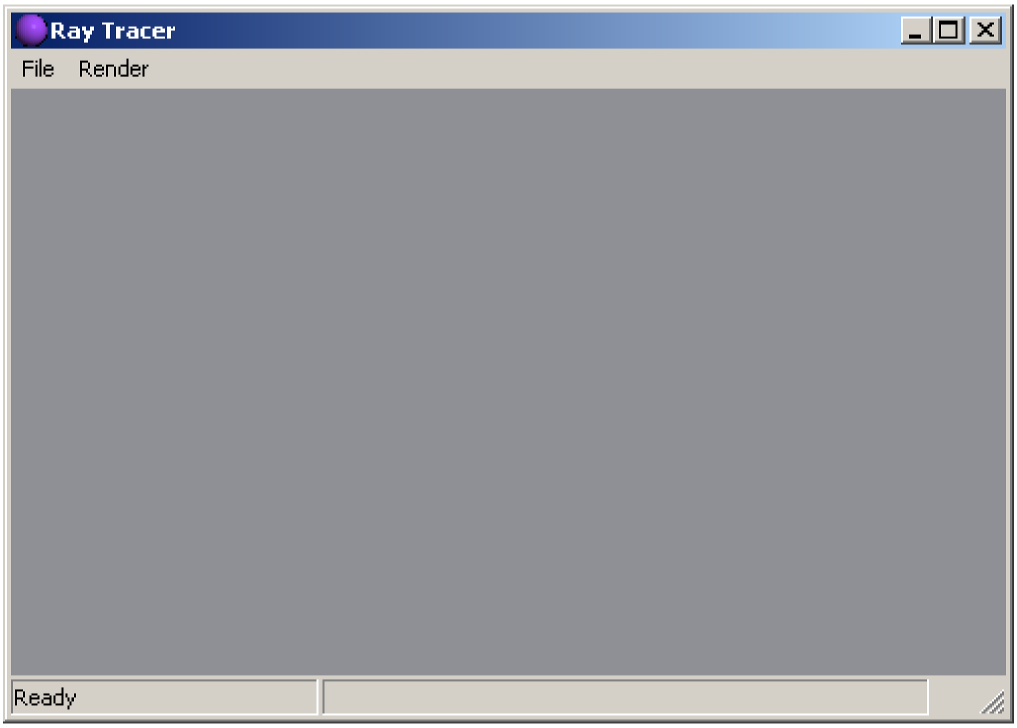


Step 4: Execute the ray tracer

Select “Debug” > “Start Without Debugging” or hit Ctrl+F5 to start the ray tracer.



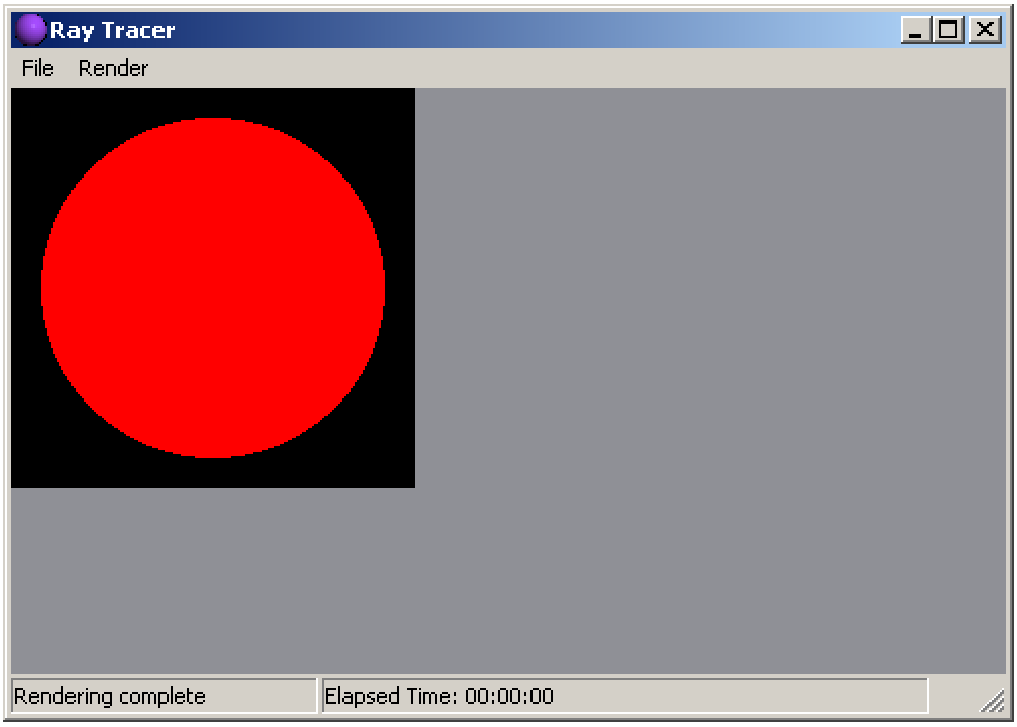
The ray tracer window will open.



Ready

Step 5: Render your first image

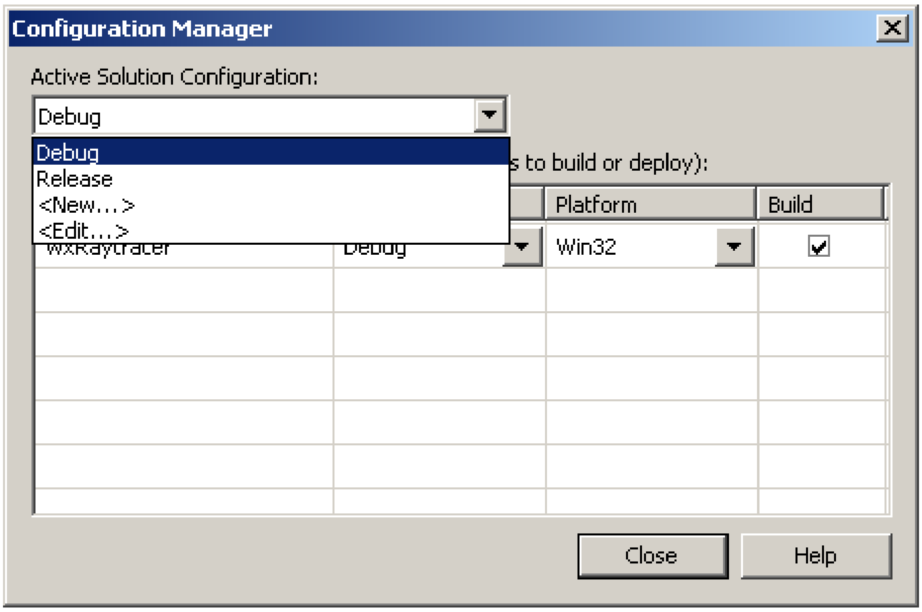
Select “Render” > “Start” from the menu to start rendering.



Rendering complete

Elapsed Time: 00:00:00

Note: the project has two different build configurations; debug and release. While debugging or rendering less complex images you might want to use the debug configuration, but when rendering larger and complex images you should change the active configuration to release. The reason for this is that the executable produced by the release build configuration will execute much faster and thus result in your image rendering faster. You can change the active configuration using the Configuration Manager which is opened by selecting “Build” > “Configuration Manager...”.



Active Solution Configuration: Debug

		Platform	Build
wxRaytracer	Debug	Win32	☑

Microsoft Visual C++ 6.0

Step 0: Installing the latest service pack

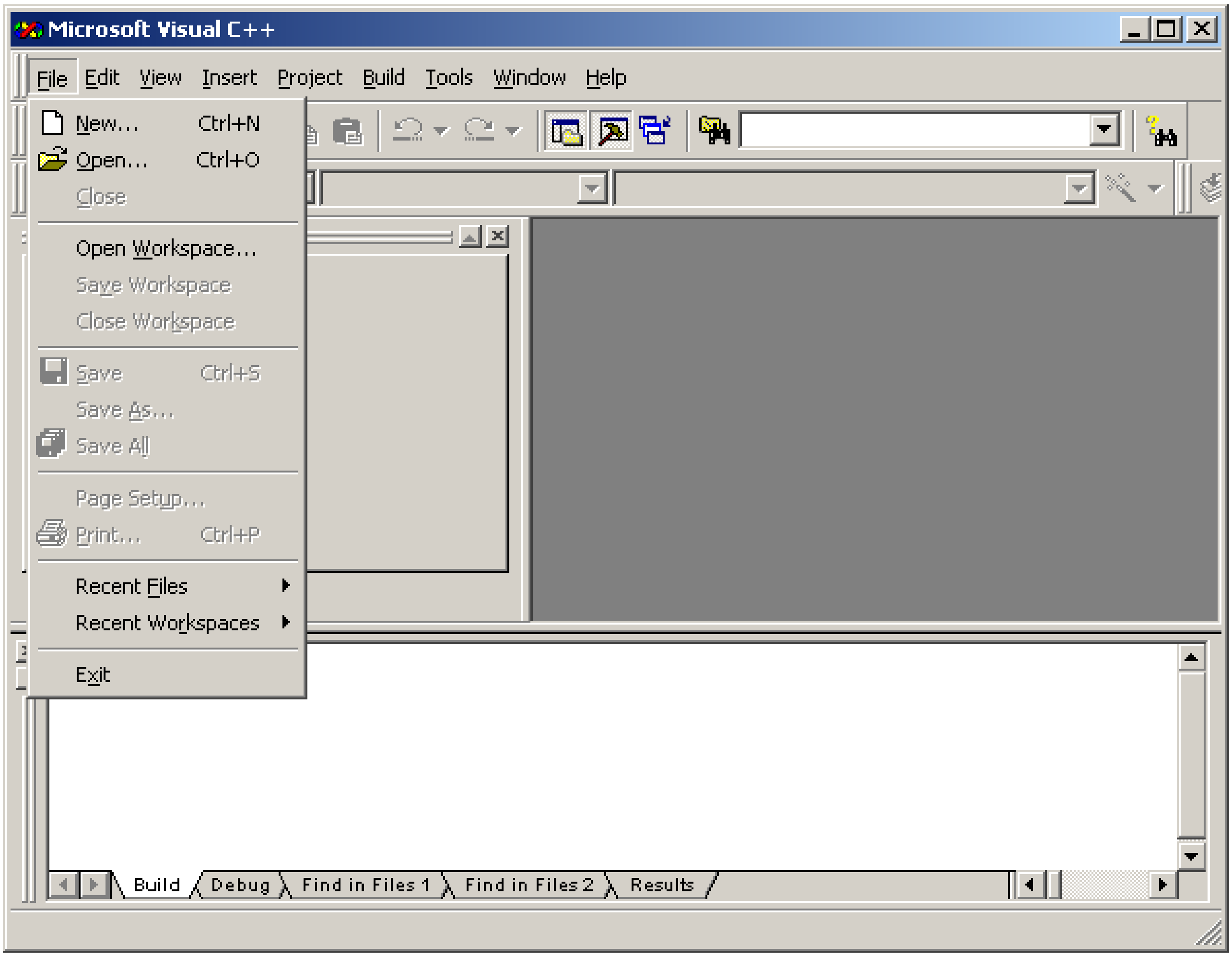
The project requires Visual Studio Service Pack 6 or later installed. This service pack can be downloaded from the Microsoft website and is ~ 60MB to download.

Step 1: Unzip the project files

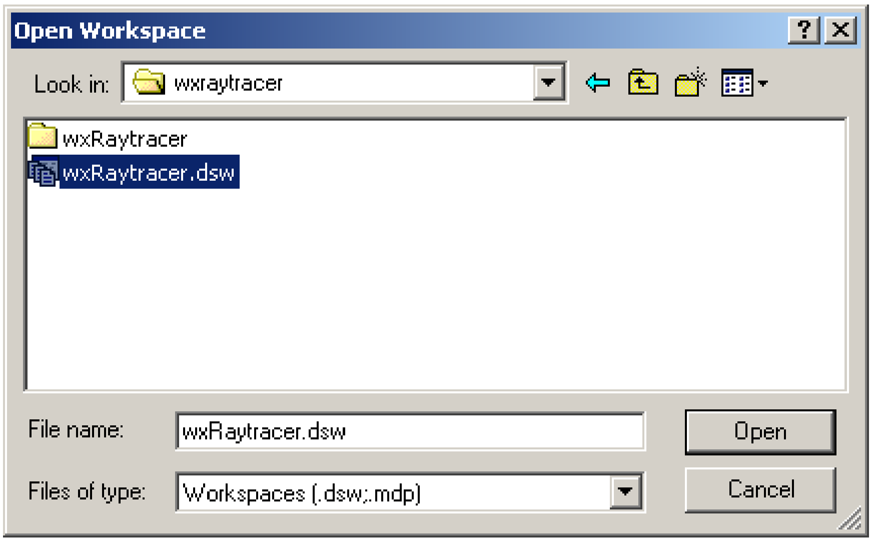
The project files are compressed as a zip archive. To decompress the files, a zip file archiver tool is required. If you don't already have one on your system, a powerful open source tool which will do the trick can be downloaded from http://www.7-zip.org

Step 2: Open the project in Visual C++

Select “File” > “Open Workspace...” to open the “Open Workspace” dialogue.



Navigate to the folder where you unzipped the project files and open the file “wxRaytracer.dsw”.



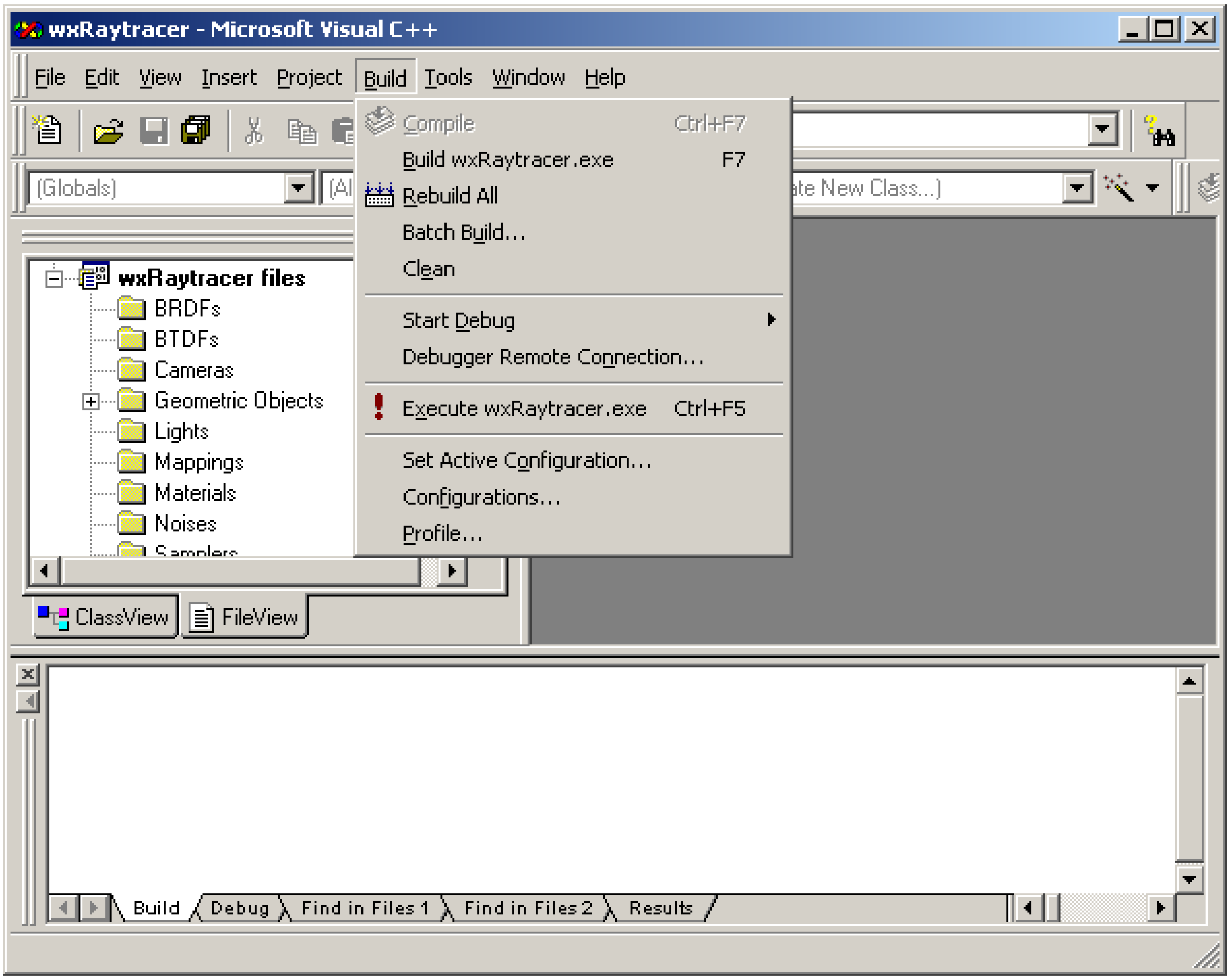
Look in: wxraytracer

File name: wxRaytracer.dsw

Files of type: Workspaces (.dsw;.mdp)

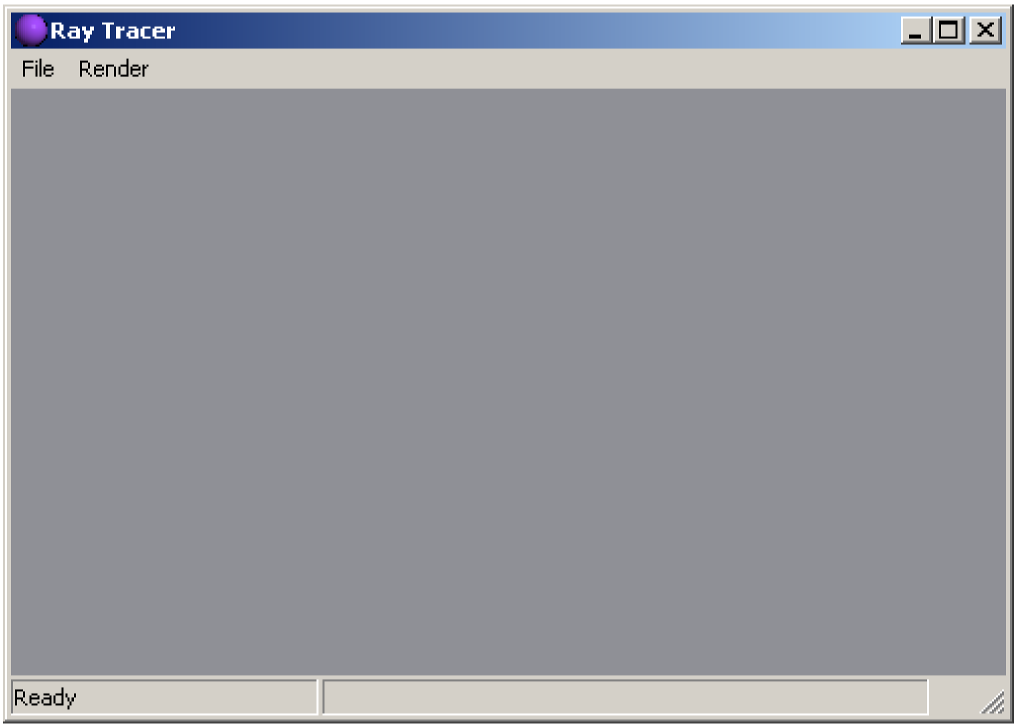
Step 3: Compile the project

Select “Build” > “Build wxRaytracer.exe” or hit F7 to invoke the compiler. The output from the compiler will be displayed in the “Build” tab of the window at the bottom.



Step 4: Execute the ray tracer

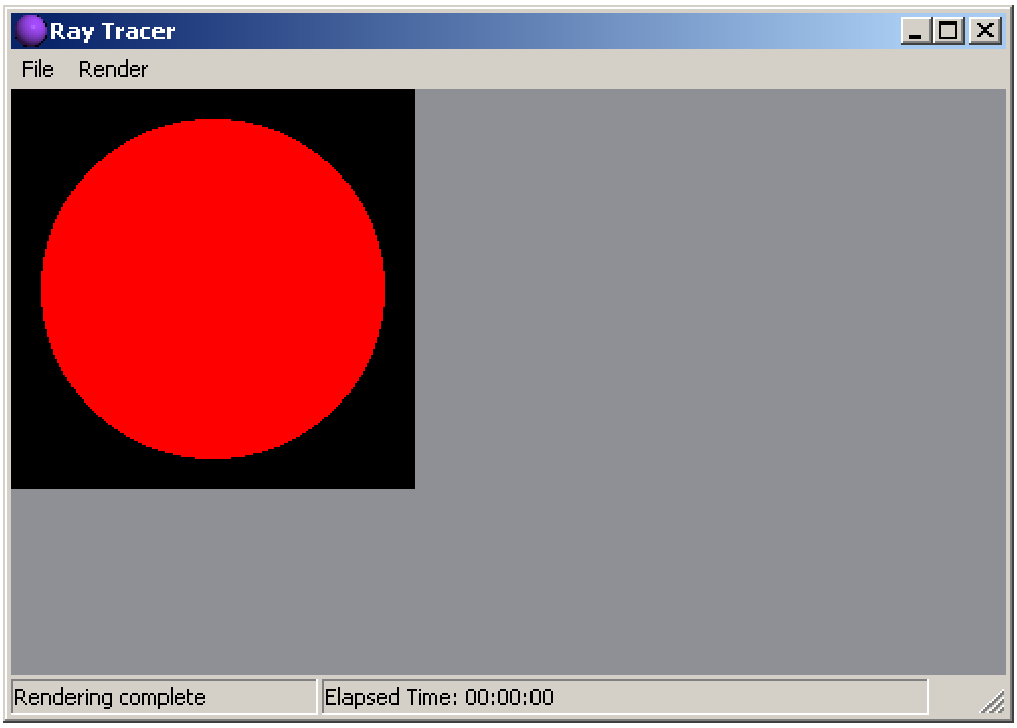
Select “Build” > “Execute wxRaytracer.exe” or hit Ctrl+F5 to start the ray tracer. The ray tracer window will open.



Ready

Step 5: Render your first image

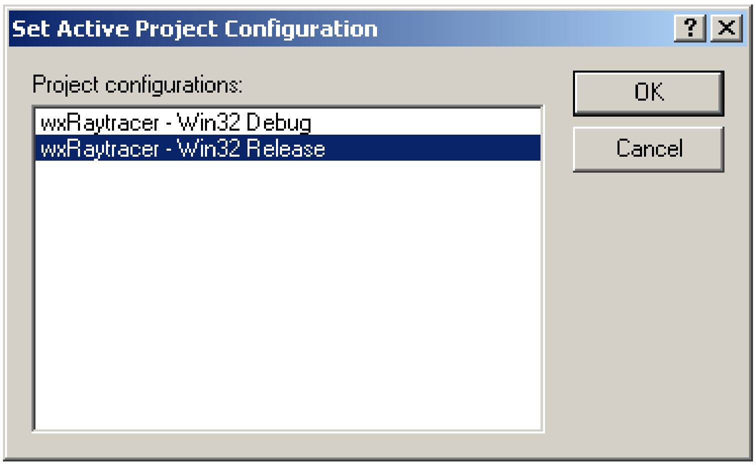
Select “Render” > “Start” from the menu to start rendering.



Rendering complete

Elapsed Time: 00:00:00

Note: the project has two different build configurations; debug and release. While debugging or rendering less complex images you might want to use the debug configuration, but when rendering larger and complex images you should change the active configuration to release. The reason for this is that the executable produced by the release build configuration will execute much faster and thus result in your image rendering faster. You can change the active configuration using the Configuration Manager which is opened by selecting “Build” > “Set Active Configuration...”.



Set Active Project Configuration

Project configurations:

wxRaytracer - Win32 Debug

wxRaytracer - Win32 Release

Eclipse 3.2.1 on Linux

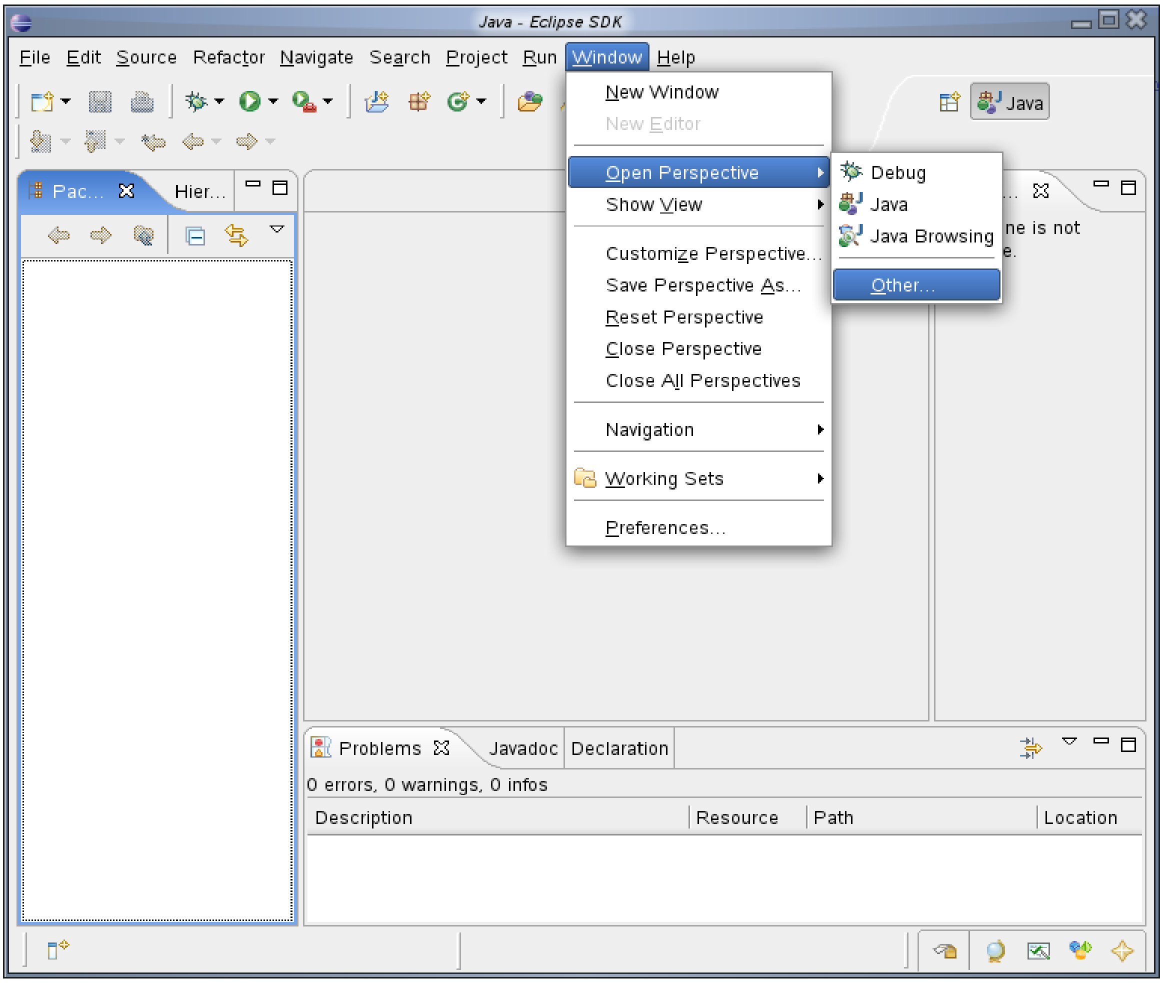
In addition to Eclipse the following must be installed before starting this tutorial:

- Eclipse CDT: needed for C++ support in Eclipse.
- Compiler and build system: g++, make etc.
- wxWidgets library: see http://www.wxwidgets.org

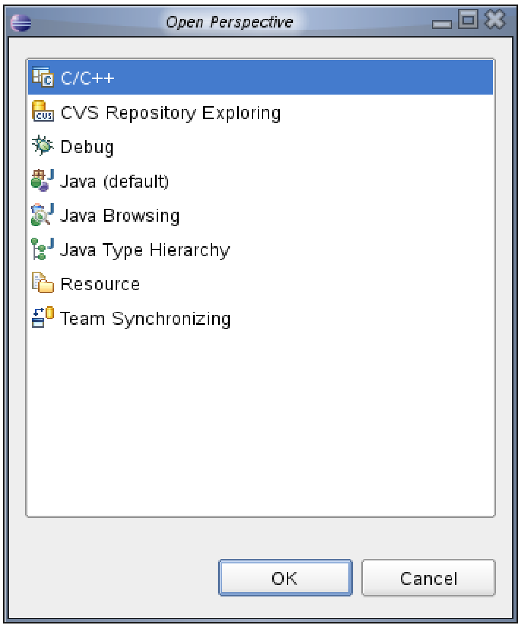
This Eclipse project was created to be used with version 2.8.4 of wxWidgets. Later versions should also work, but might require some configuration changes. These changes are explained as part of this tutorial.

Step 1: Open C++ perspective

Eclipse is configured for Java by default. Before the ray tracer files can be opened, the perspective must be changed to C/C++. Select “Window” > “Open Perspective” > “Other...” to open the “Open Perspective” dialogue.

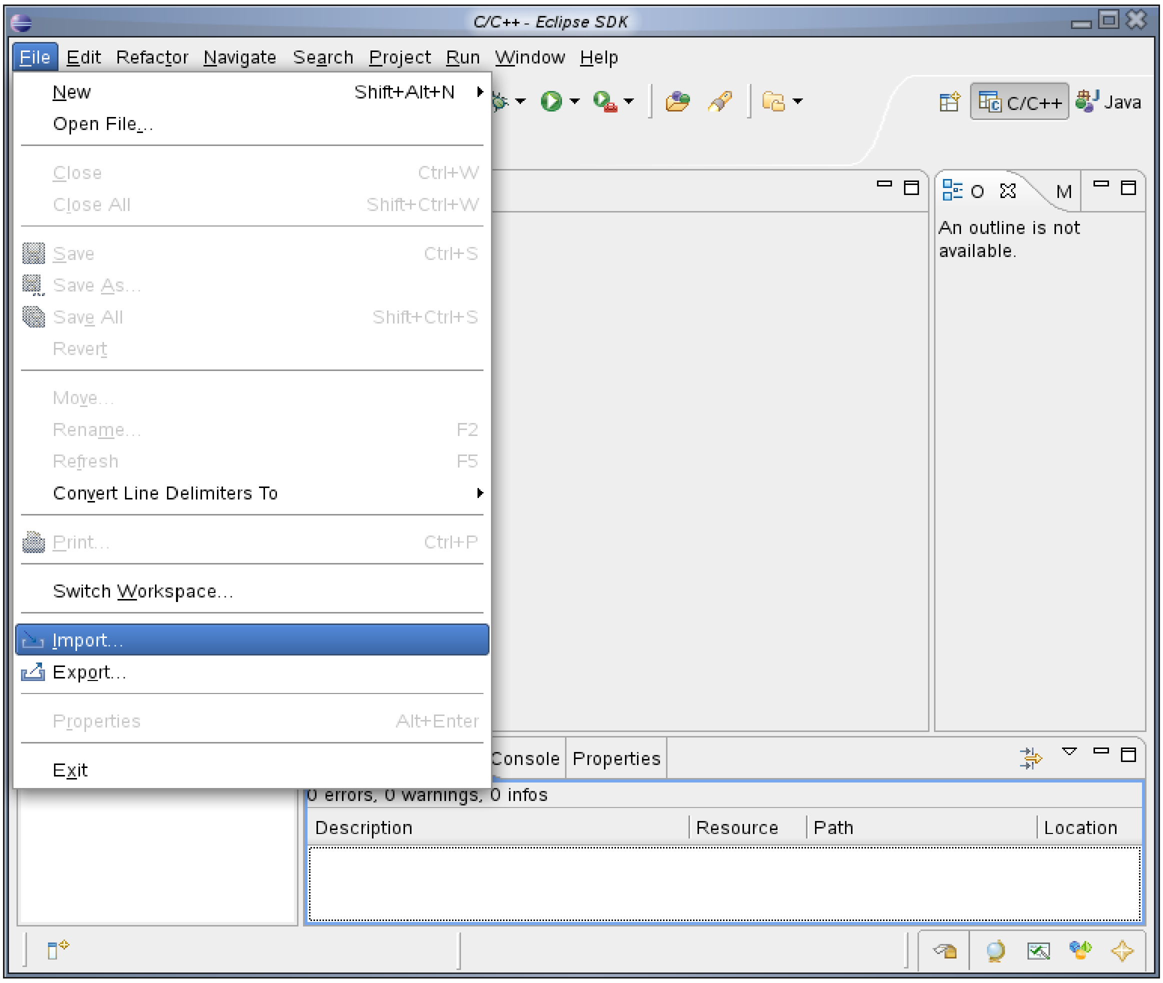


Select “C/C++” in the “Open Perspective” dialogue.

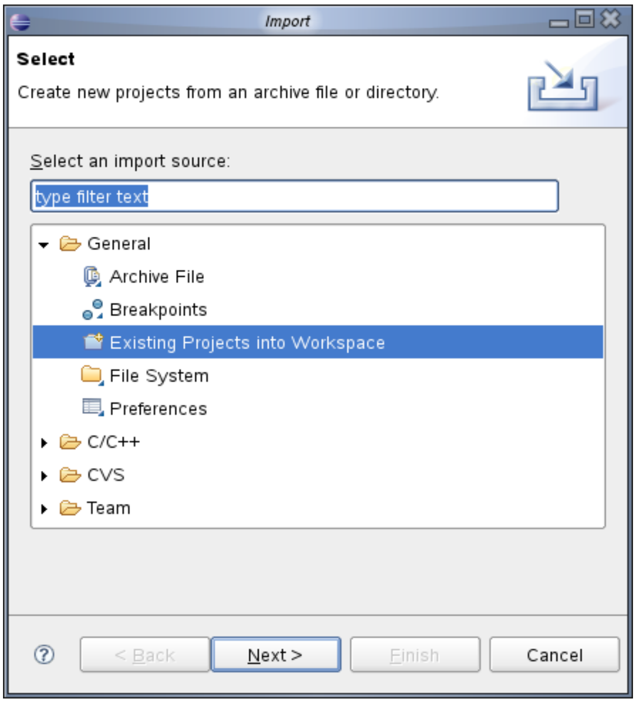


Step 2: Importing the ray tracer project into the current workspace

Select “File” > “Import...” to open the “Import” dialogue.



Select “Existing Projects into Workspace” in the “Import” dialogue.



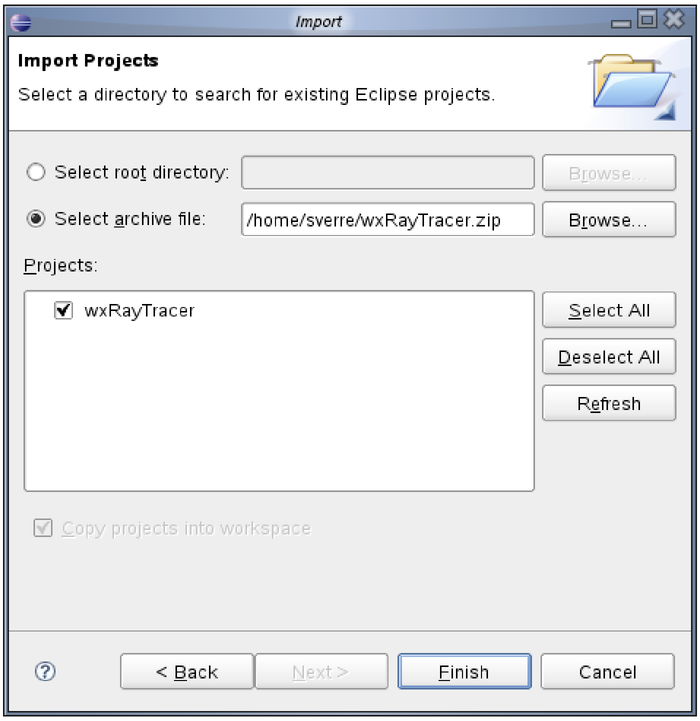
Import

Select

Create new projects from an archive file or directory.

Select an import source: type filter text

The Eclipse project for the ray tracer is embedded in a zip archive. Navigate to the archive file as shown in the dialogue box below. Make sure the “wxRayTracer” project is selected before clicking “Finish”.



Import

Import Projects

Select a directory to search for existing Eclipse projects.

○ Select root directory:

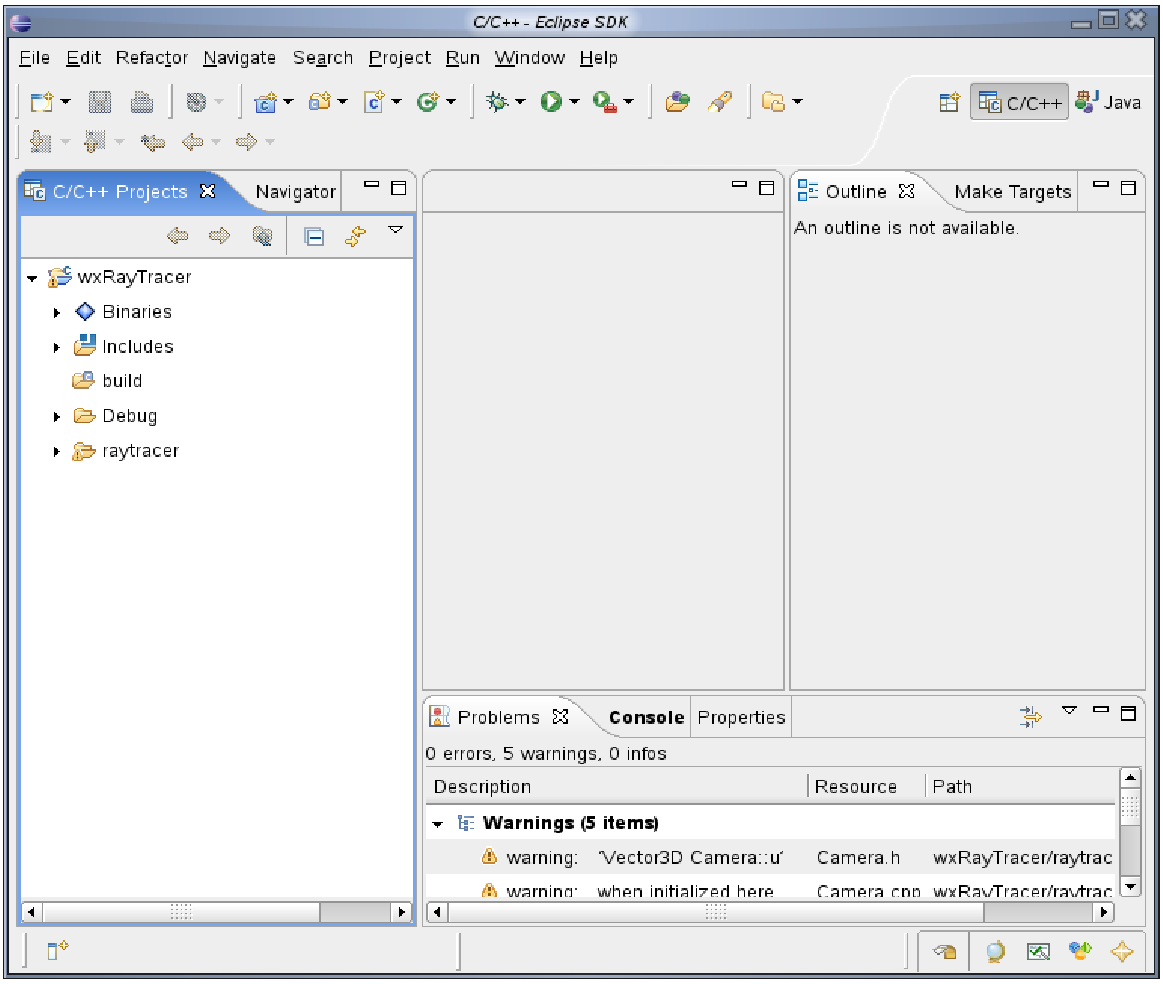
● Select archive file: /home/sverre/wxRayTracer.zip

Projects:

☑ wxRayTracer

☑ Copy projects into workspace

By default, Eclipse will compile the project automatically as soon as the “Finish” button is clicked in the “Import” dialogue box above. If the existing project settings match your environment, you should get a result similar to the screen below; no errors but a few warnings. If build errors are reported, your environment is not configured correctly or not matching the project settings. See “Configuring the Eclipse project” for information on how to configure the project for your environment.



An outline is not available.

0 errors, 5 warnings, 0 infos

Description	Resource	Path
Warnings (5 items)		
warning: 'Vector3D Camera::u'	Camera.h	wxRayTracer/raytrac
warning: when initialized here	Camera.cpp	wxRayTracer/raytrac

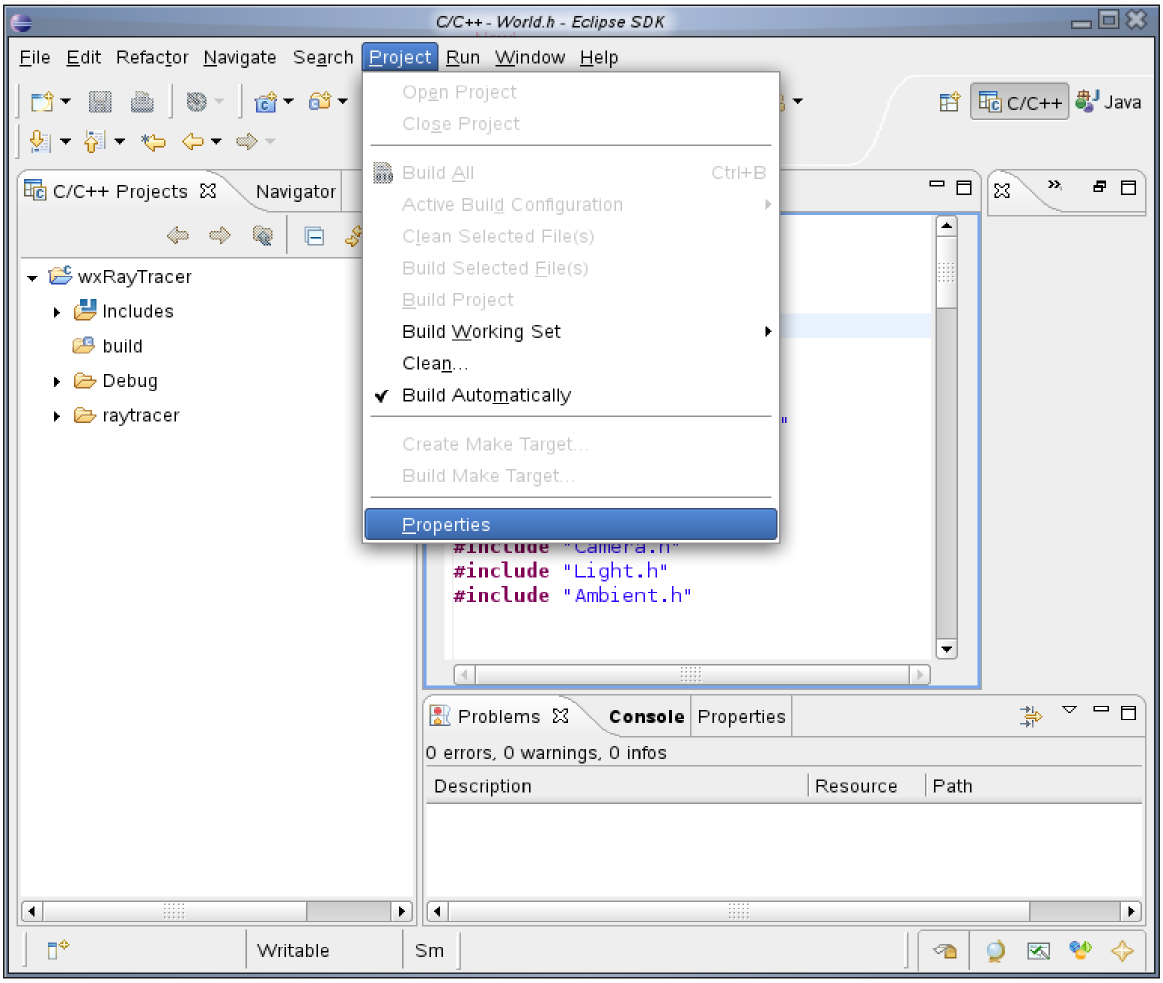
Step 3: Setting up run configurations

The project has two different build configurations; “Debug” and “Release”. While debugging or rendering less complex images you might want to use the debug configuration, but when rendering larger and complex images you should change the active configuration to release. The reason for this is that the executable produced by the release build configuration is optimised and will execute much faster and thus result in your image rendering faster. You can change the active configuration
in the project properties.

Even though the project is now compiled, run configurations need to be set up before the ray tracer can be executed in Eclipse. This step will set up run configurations for both build configurations.

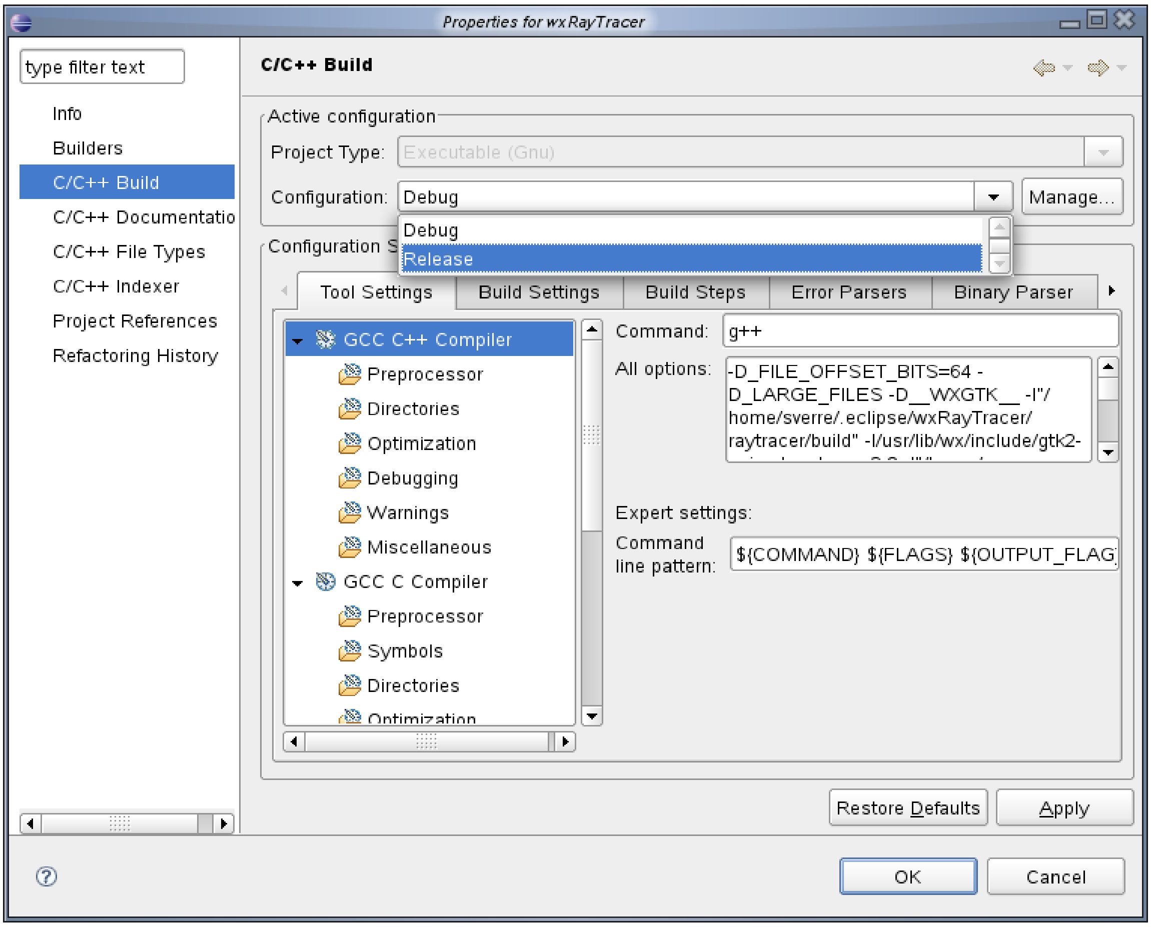
Currently the project has been built using the “Debug” configuration. In order to set up the run configurations, the project has to be compiled using the “Release” configuration. To build the project with the release configuration, the project properties have to be modified.

Select “Project” > “Properties” to open the project properties window. If the “Properties” item is disabled under the “Project” menu, open a source file from the project using the project browser on the left hand side and try again.



```
#include "Camera.h"
#include "Light.h"
#include "Ambient.h"
```

In the properties window, select “C/C++ Build” in the menu on the left hand side. Next, change the configuration from “Debug” to “Release” as shown below and click “OK”. Eclipse will now compile the project again using the release configuration.



Properties for wxRayTracer

C/C++ Build

Active configuration

Project Type: Executable (Gnu)

Configuration: Debug

Debug

Release

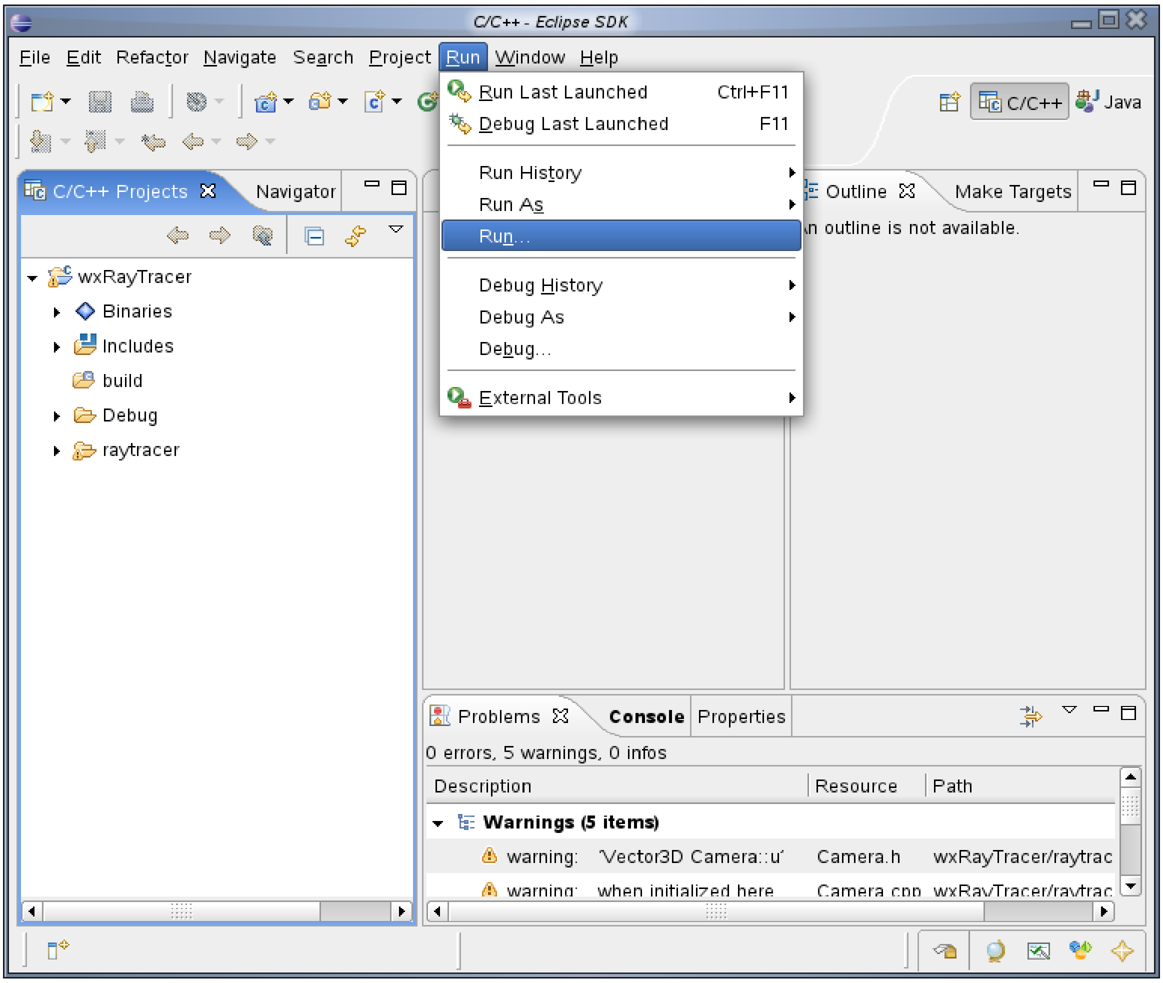
Command: g++

All options: -D_FILE_OFFSET_BITS=64 -D_LARGE_FILES -D__WXGTK__ -I"/home/sverre/.eclipse/wxRayTracer/raytracer/build" -I/usr/lib/wx/include/gtk2-

Expert settings:

Command line pattern: ${COMMAND} ${FLAGS} ${OUTPUT_FLAG

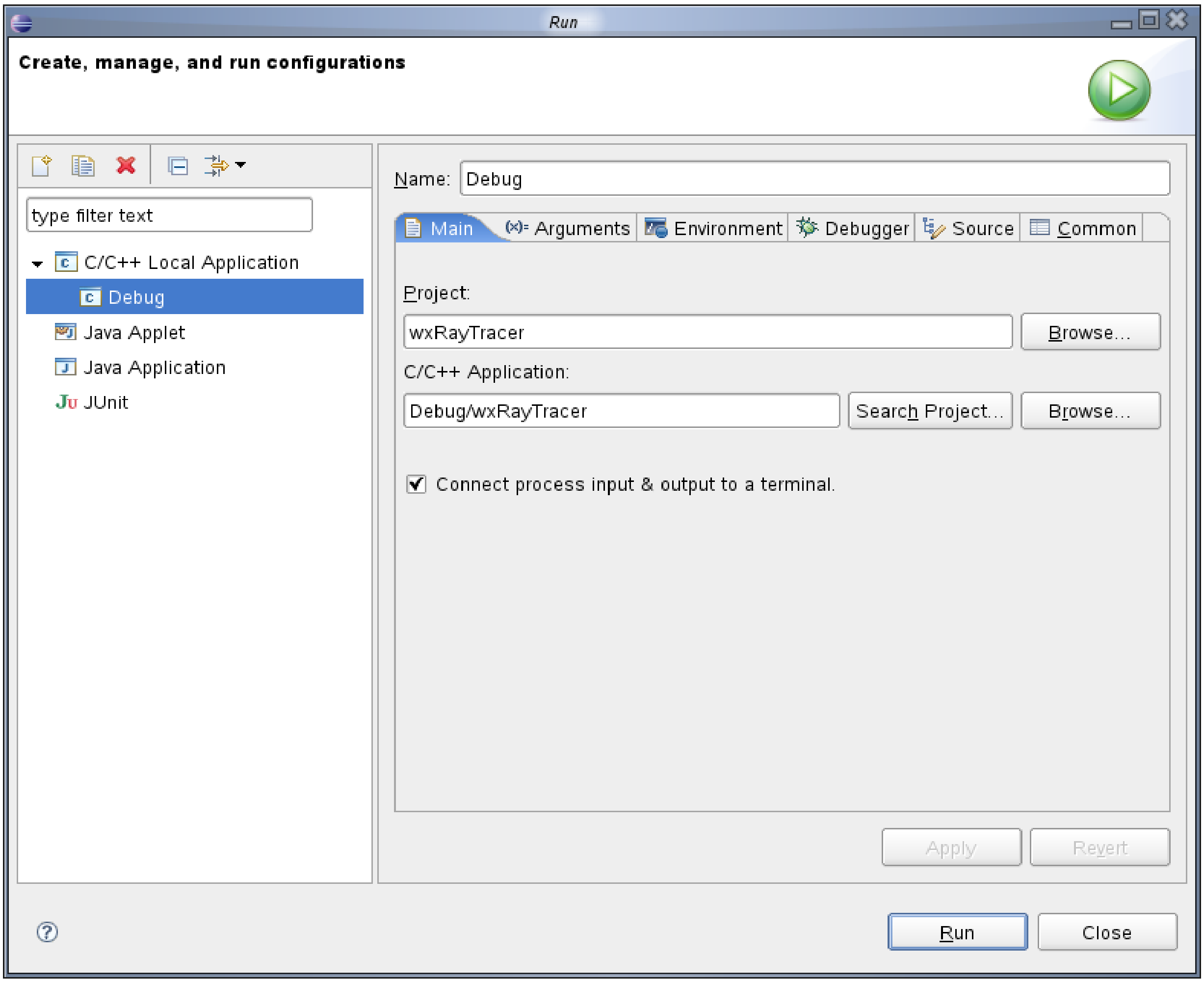
Select “Run” > “Run...” to open the “Run” dialogue.



0 errors, 5 warnings, 0 infos

Description	Resource	Path
Warnings (5 items)		
warning: 'Vector3D Camera::u'	Camera.h	wxRayTracer/raytrac
warning: when initialized here	Camera.cpp	wxRayTracer/raytrac

Select “C/C++ Local Application” and press the “New” button (leftmost button in the menu bar above the “type filter text” field). Now, enter the details below and click “Apply”.



Run

Create, manage, and run configurations

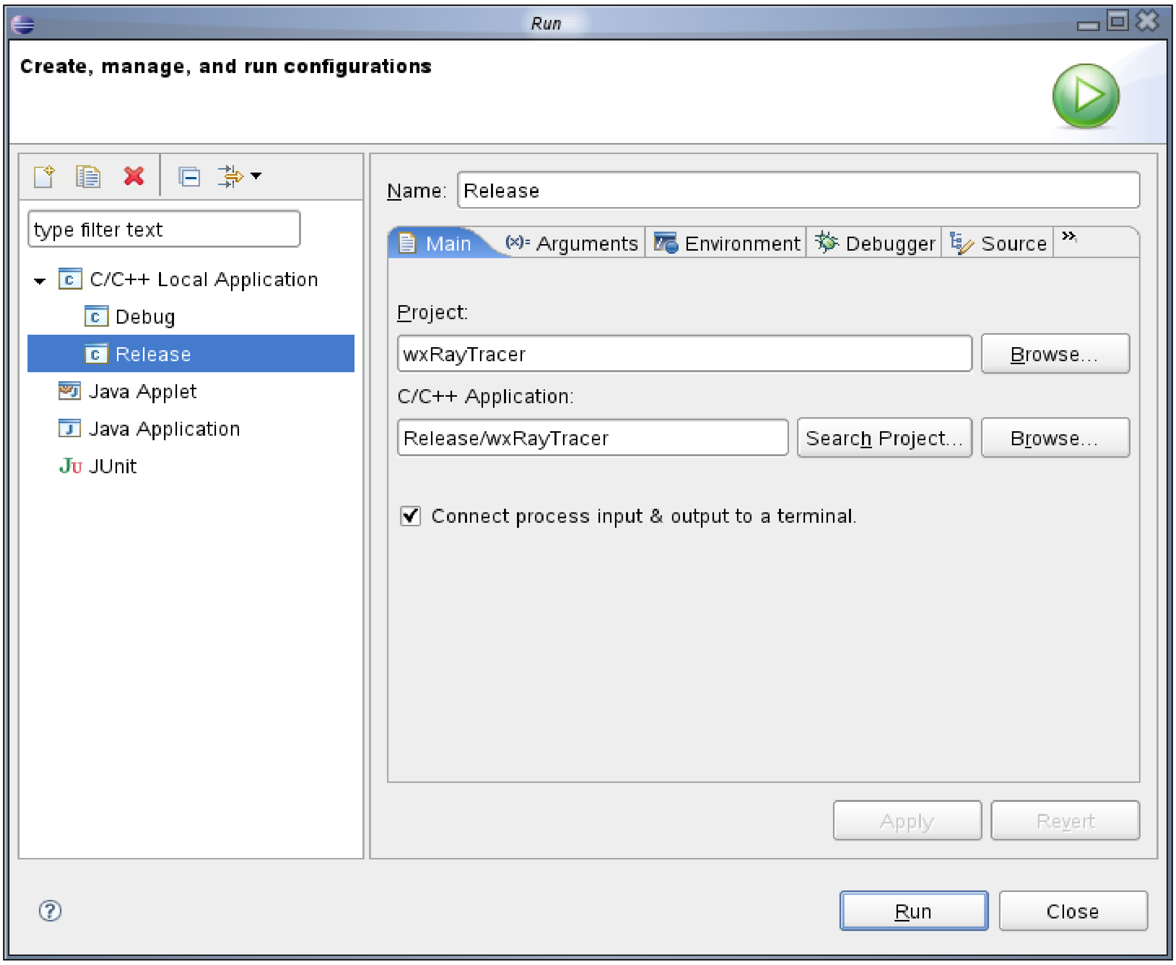
Name: Debug

Project: wxRayTracer

C/C++ Application: Debug/wxRayTracer

☑ Connect process input & output to a terminal.

Repeat the above step, but using the details below. When done, click the “Close” button.



Name: Release

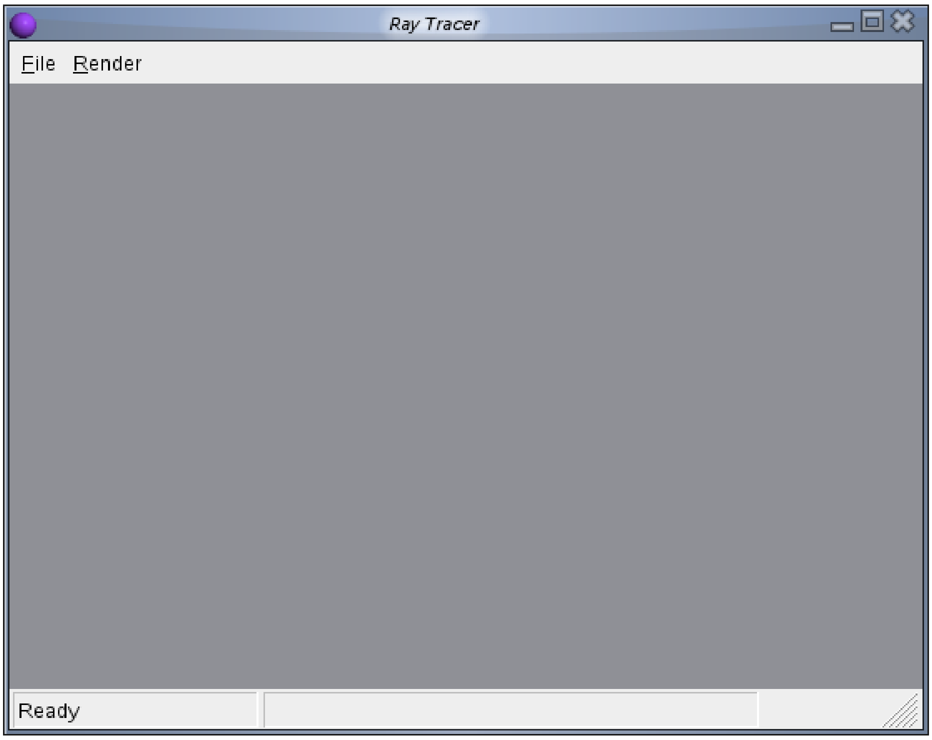
Project: wxRayTracer

C/C++ Application: Release/wxRayTracer

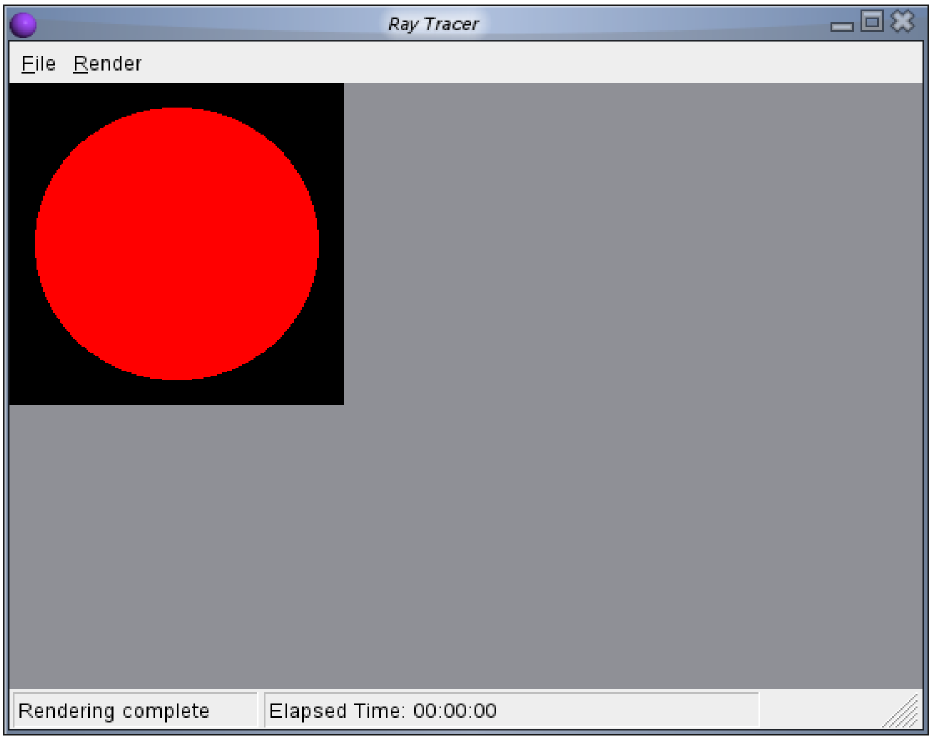
☑ Connect process input & output to a terminal.

Step 4: Execute the ray tracer

Select “Run” > “Run...” to open the “Run” dialogue. Under “C/C++ Local Application”, select “Debug” or “Release” and click the “Run” button. The ray tracer window will open.



Select “Render” > “Start” from the menu to start rendering.



Rendering complete

Elapsed Time: 00:00:00

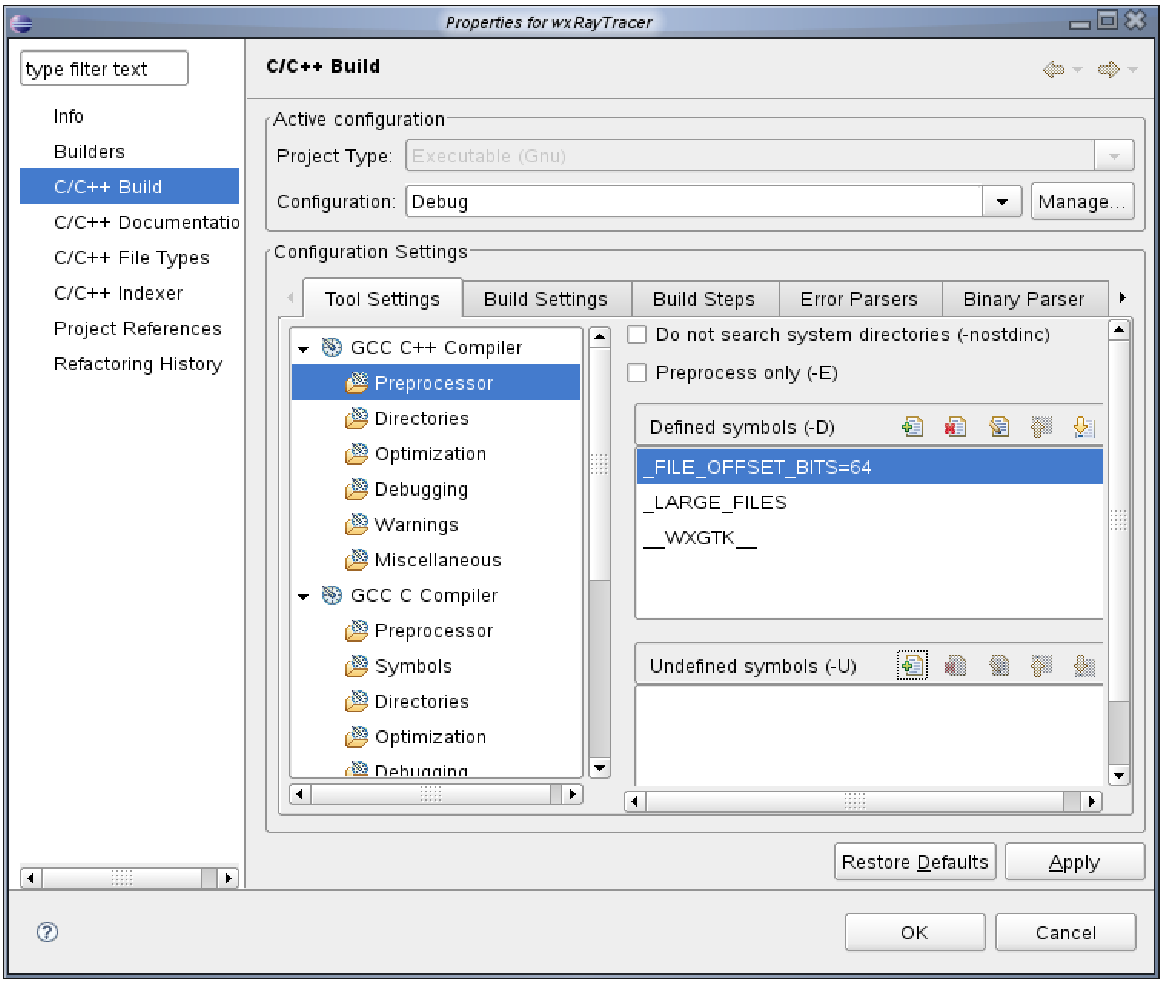
Configuring the Eclipse project

wxWidgets comes with a configuration tool called “wx-config”. On some installations, the executable might be have a version number added to the name of the executable i.e. “wx-config-2.8”. This section will consistently refer to “wx-config-2.8” but please change it to whatever fits your installation.

In order to retrieve information about the wxWidgets libraries available on your system, open up a console and type the command “wx-config-2.8 --cxxflags”. The output should be something similar to:

```
-I/usr/lib/wx/include/gtk2-unicode-release-2.8 -I/usr/include/wx-2.8
-D_FILE_OFFSET_BITS=64 -D_LARGE_FILES -D__WXGTK__ -pthread
```

Select “Project” > “Properties” to open the project properties window and select “C/C++ Build” in the left hand menu. Under “Tool Settings” select “GCC C++ Compiler” > “Preprocessor”. For everything in the output from “wx-config-2.8 --cxxflags” starting with “-D” click on the “Add” button in the “Defined Symbols” field and add the string but without the leading “-D”. Delete any existing symbols which were not output by the “wx-config-2.8 --cxxflags” command.



Properties for wxRayTracer

C/C++ Build

Active configuration

Project Type: Executable (Gnu)

Configuration: Debug

Configuration Settings

□ Do not search system directories (-nostdinc)

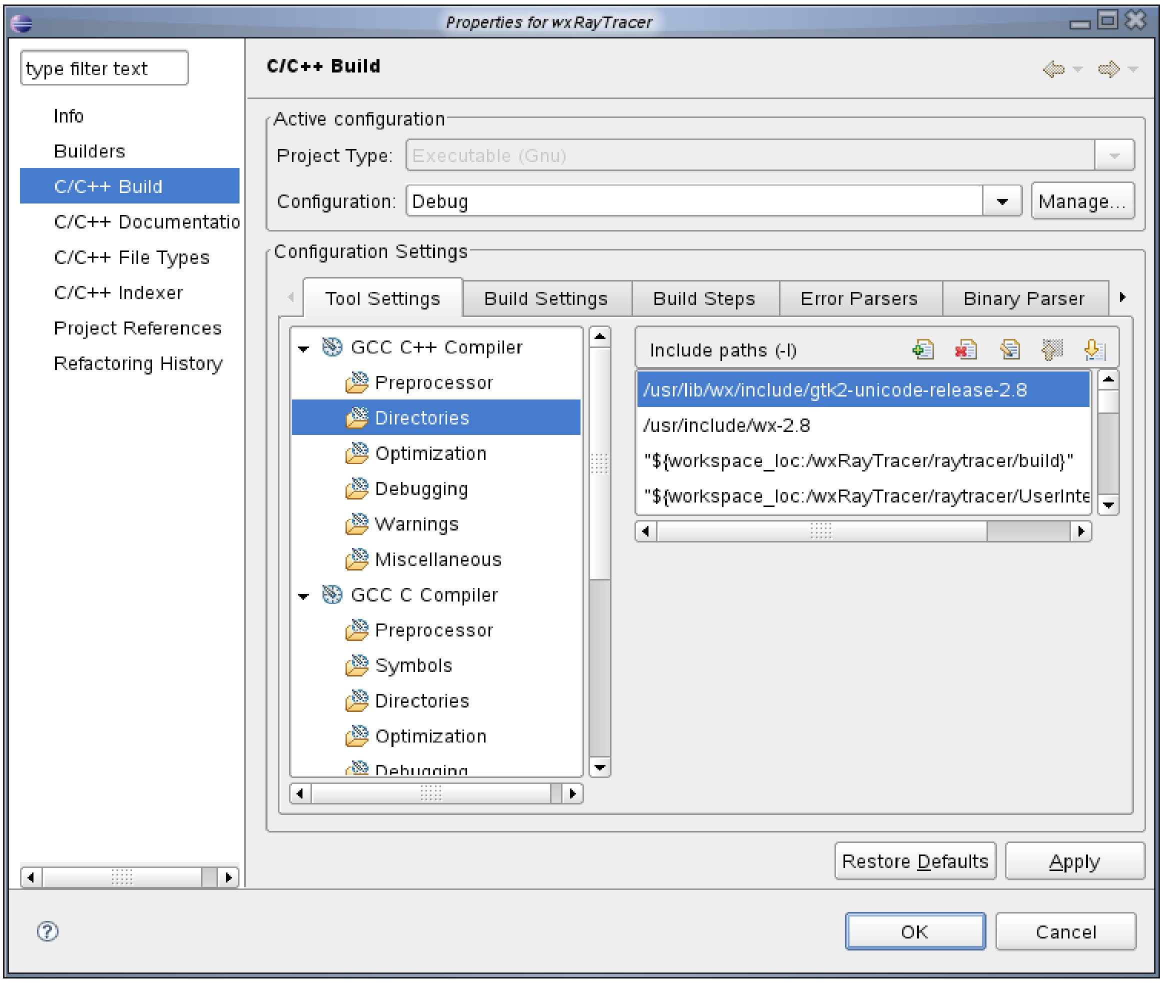
□ Preprocess only (-E)

Defined symbols (-D)

_FILE_OFFSET_BITS=64
_LARGE_FILES
__WXGTK__

Undefined symbols (-U)

Select “GCC C++ Compiler” > “Directories”. For everything in the output from “wx-config-2.8 --cxxflags” starting with “-I” click on the “Add” button in the “Include paths” field and add the string but without the leading “-I”. Delete any existing paths starting with “/” which were not output by the “wx-config-2.8 --cxxflags” command.



Properties for wxRayTracer

C/C++ Build

Active configuration

Project Type: Executable (Gnu)

Configuration: Debug

Configuration Settings

Tool Settings > GCC C++ Compiler > Directories

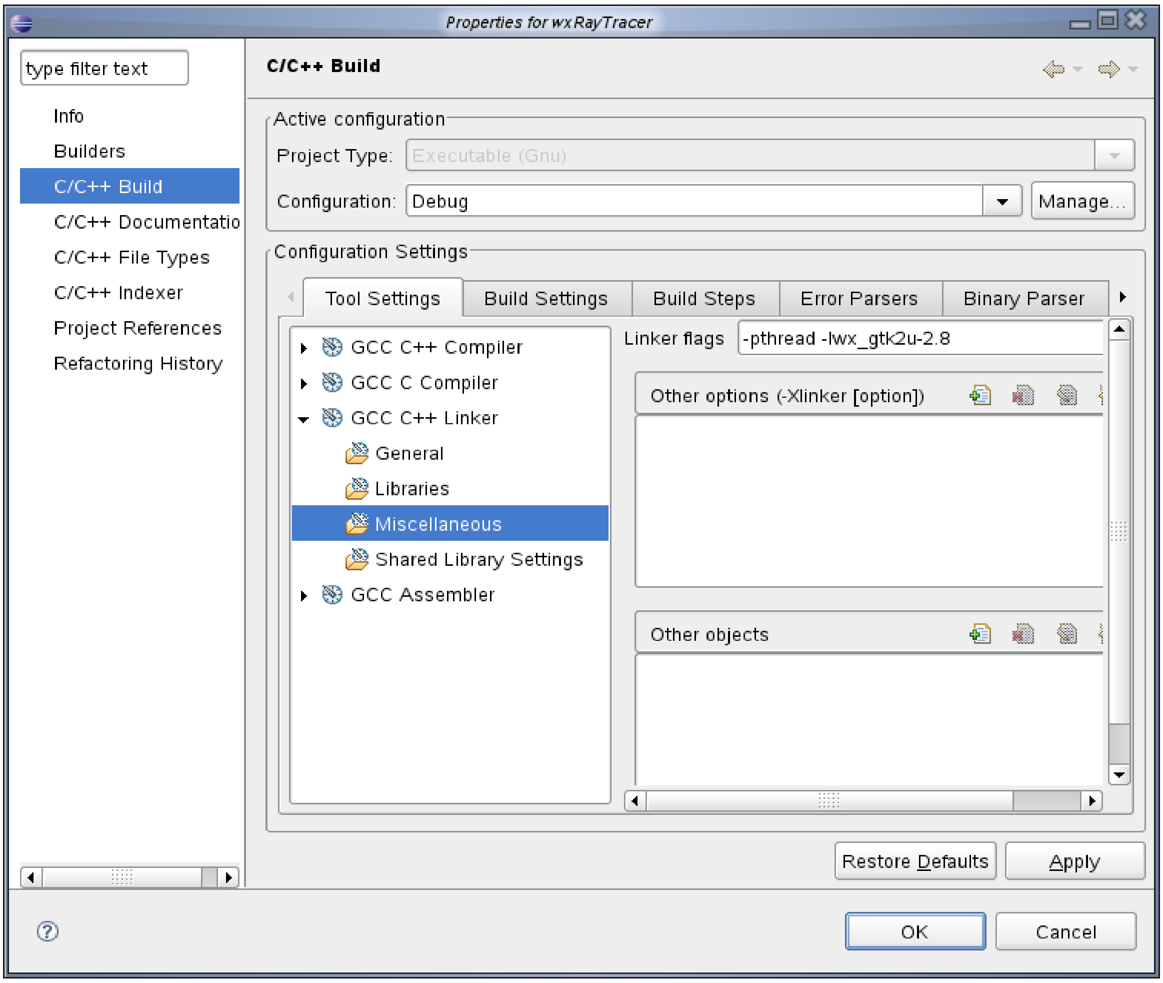
Include paths (-I)

/usr/lib/wx/include/gtk2-unicode-release-2.8
/usr/include/wx-2.8
"${workspace_loc:/wxRayTracer/raytracer/build}"
"${workspace_loc:/wxRayTracer/raytracer/UserInte

In order to retrieve information about the required wxWidgets linker flags to be used on your system, open up a console and type the command "wx-config-2.8 --libs". The output should be something similar to:

```
-pthread   -lwx_gtk2u-2.8
```

Select "GCC C++ Linker" > "Miscellaneous". Add the output from "wx-config-2.8 --libs" to the "Linker flags" field. Delete any existing linker flags which were not output by the "wx-config-2.8 --libs" command.



Linker flags -pthread -lwx_gtk2u-2.8

The project settings should now match your environment and the project should compile successfully.

References

http://www.wxwidgets.org/wiki/index.php/Eclipse
http://max.berger.name/howto/wxWidgets/wxWidgets_Eclipse.jsp