Compiler Setup and DirectX/OpenGL Setup

A very important part of programming is choosing a decent development environment setup. Microsoft's Visual Studio product range is by far one of the greatest IDEs ever developed, so go and buy a recent version of Microsoft Visual Studio, many computer science departments also have agreements with Microsoft to provide these IDEs free of charge to their students – please enquire about this. IDEs like Visual Studio include everything a developer might need; from real-time debugging tools to utilities designed for the elimination of memory leaks.

Visual Studio also provides an uncluttered and aesthetically pleasing environment to work in. Generally speaking, I'm yet to come across many game development companies not using Visual Studio – it is the complete development package, even supporting mobile development. That said; one possible downside to Visual Studio is its platform dependency. It is a Windows-only product – although this works fine in the world of computer games, it doesn't fit well into the grand scheme of things as far as platform interoperability goes. Of course this doesn't hinder your application in any way. All the DirectX and OpenGL code available on the book's website was written and compiled using Visual Studio 2005.

We'll now briefly look at using Microsoft Visual Studio .NET 2005 for the compilation of DirectX and OpenGL programs – all that's needed to run and compile DirectX programs is installation of the latest DirectX SDK available from Microsoft's website (http://msdn.microsoft.com/en-us/directx/default.aspx). Compilation of the presented OpenGL programs requires the basic OpenGL and GLUT libraries. Windows comes with OpenGL, and Visual studio provides the needed OpenGL libraries – GLUT, available from http://www.xmission.com/~nate/glut.html must, however, be installed. Simply download the latest zip file and extract "glut32.dll" to "…\Windows\system\" with "glut32.lib" going in Visual Studio's "…\VC\PlatformSDK\Lib" folder. The main GLUT header file, glut.h, must be extracted to Visual Studio's "PlatformSDK\Include\gl" folder. Figure A.1 shows the contents of the "…\VC\PlatformSDK\Include\gl" folder:

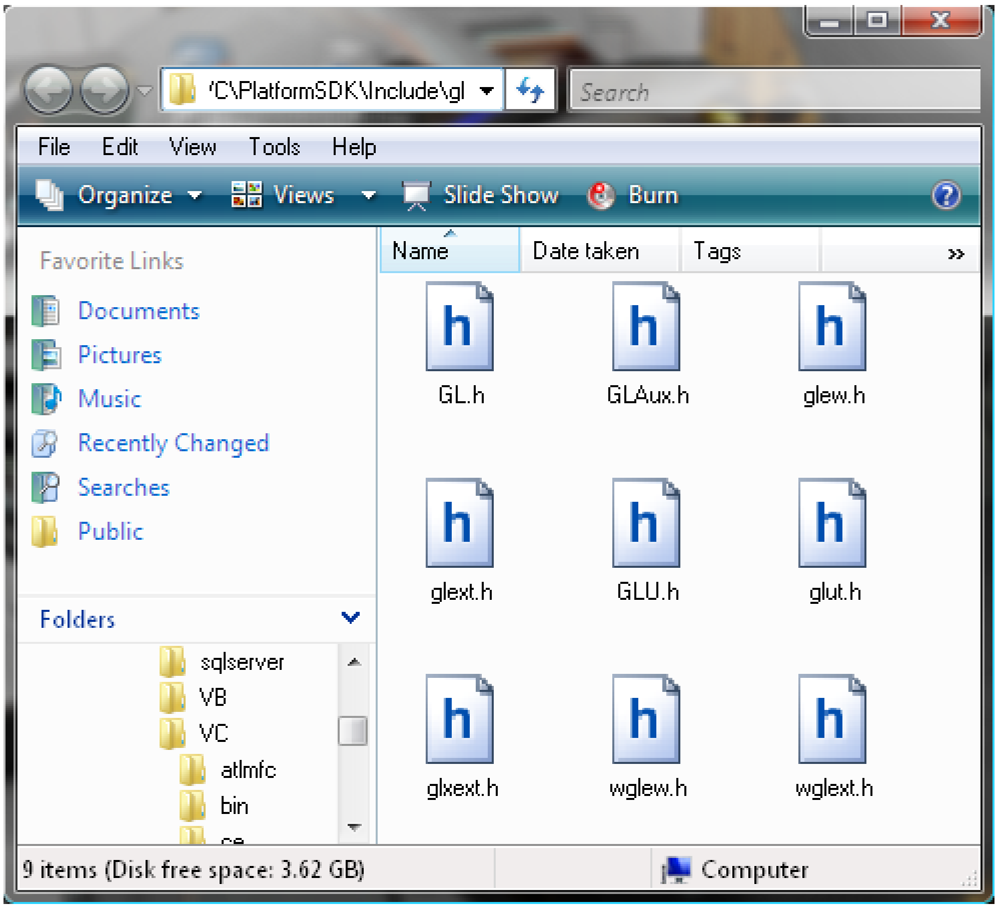
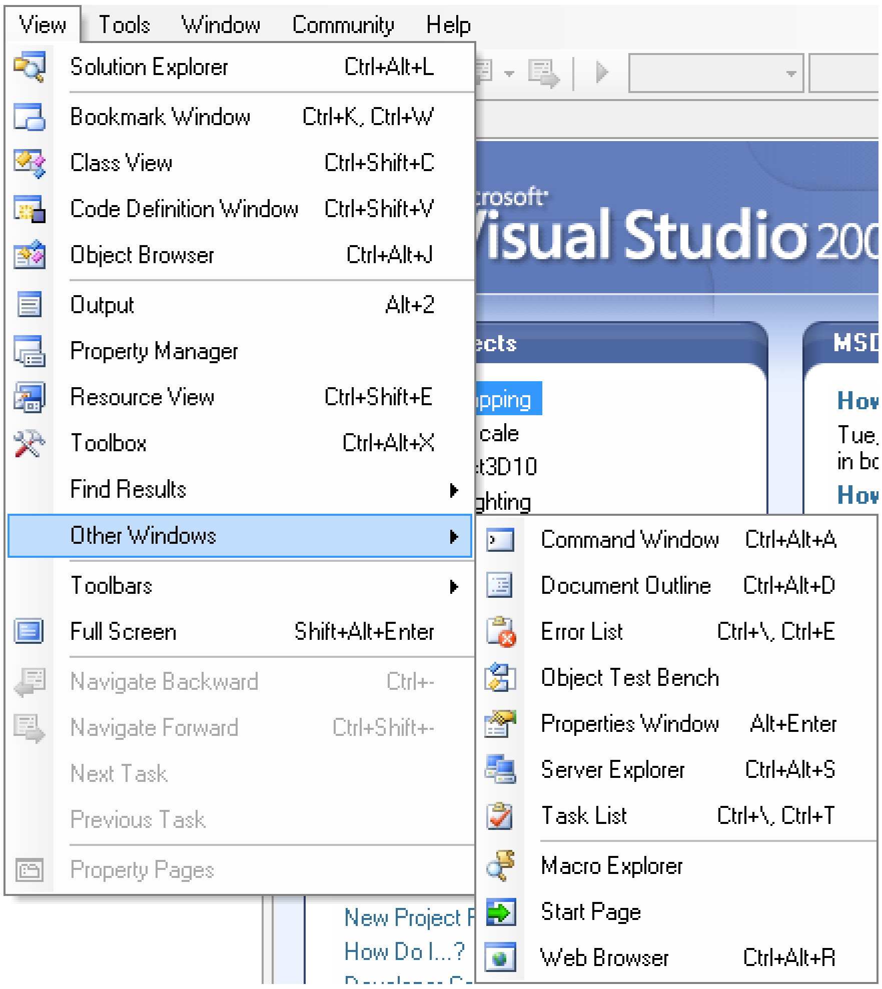


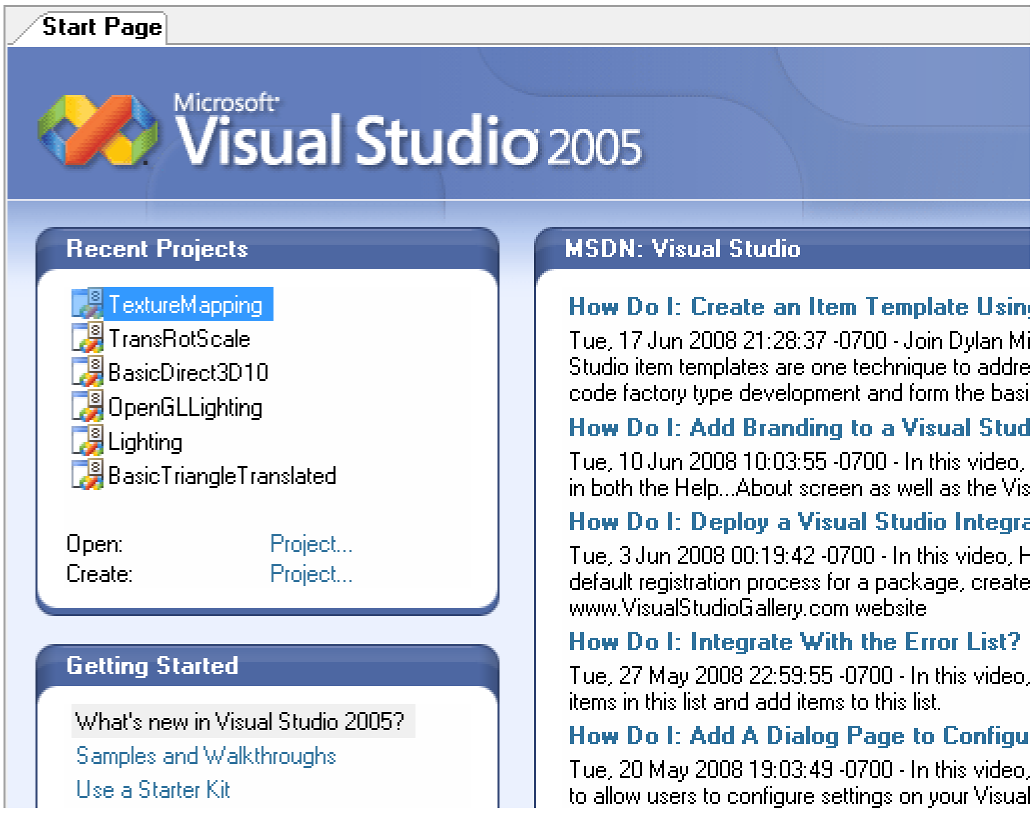
Fig A.1 Contents of the "…\VC\PlatformSDK\Include\gl" folder.

Microsoft Visual Studio 2005 Environment

Launching Visual Studio for the first time allows you to customise the default IDE layout via the selection of several predefined window layouts. You can also manually set up the visible windows by clicking on "View". This process is outlined by Figure A.2.


View
Tools
Window
Community
Help
Solution Explorer Ctrl+Alt+L
Bookmark Window Ctrl+K, Ctrl+W
Class View Ctrl+Shift+C
Code Definition Window Ctrl+Shift+V
Object Browser Ctrl+Alt+J
Output Alt+2
Property Manager
Resource View Ctrl+Shift+E
Toolbox Ctrl+Alt+X
Find Results
Other Windows
Toolbars
Full Screen Shift+Alt+Enter
Navigate Backward Ctrl+-
Navigate Forward Ctrl+Shift+-
Next Task
Previous Task
Property Pages
Command Window Ctrl+Alt+A
Document Outline Ctrl+Alt+D
Error List Ctrl+\, Ctrl+E
Object Test Bench
Properties Window Alt+Enter
Server Explorer Ctrl+Alt+S
Task List Ctrl+\, Ctrl+T
Macro Explorer
Start Page
Web Browser Ctrl+Alt+R
New Project
How Do I...?
Fig A.2 Manual setup of Visual Studio 2005's layout.

Once you're comfortable with the IDE's layout, you're ready to start programming. If you click on the "Projects" tab (Figure A.3) you'll see a list of the recently loaded or created Visual Studio projects. To create a new project, simply click on the "New Project" button. This will launch the "New Project" wizard (Figure A.4).



Start Page

Microsoft Visual Studio 2005

Recent Projects

TextureMapping
TransRotScale
BasicDirect3D10
OpenGLLighting
Lighting
BasicTriangleTranslated

Open: Project...
Create: Project...

Getting Started

What's new in Visual Studio 2005?
Samples and Walkthroughs
Use a Starter Kit

MSDN: Visual Studio

How Do I: Create an Item Template Usin
Tue, 17 Jun 2008 21:28:37 -0700 - Join Dylan Mi
Studio item templates are one technique to addre
code factory type development and form the basi

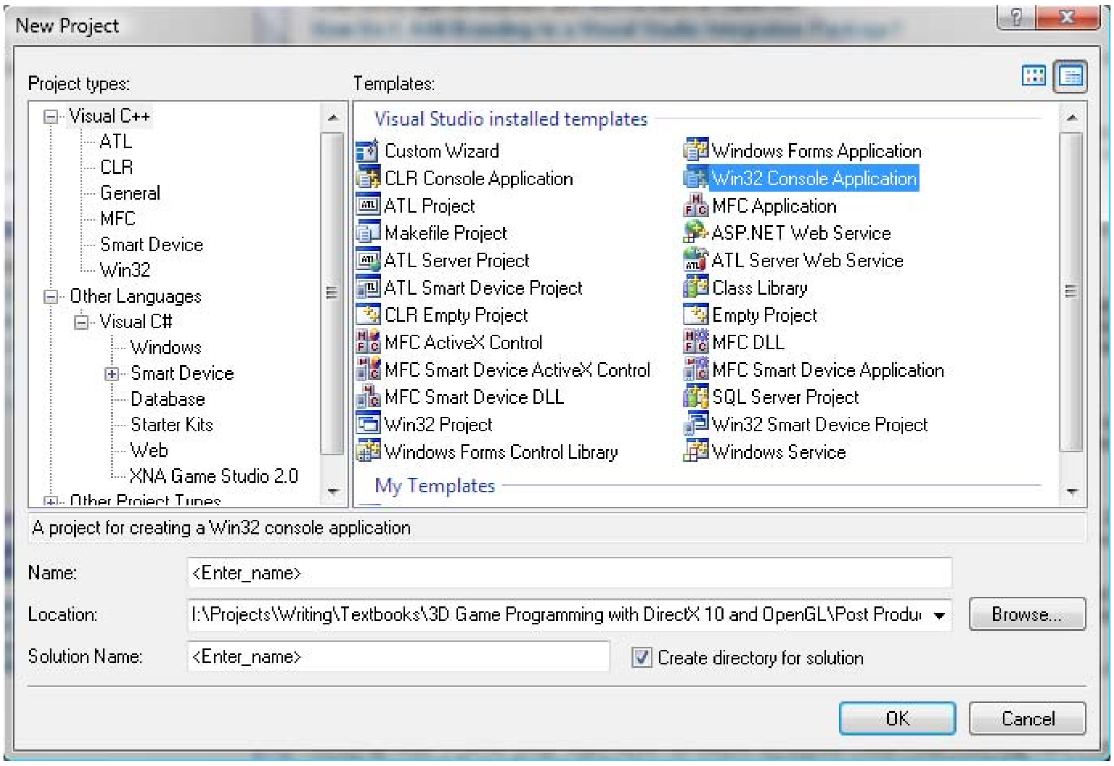
How Do I: Add Branding to a Visual Stud
Tue, 10 Jun 2008 10:03:55 -0700 - In this video,
in both the Help...About screen as well as the Vis

How Do I: Deploy a Visual Studio Integra
Tue, 3 Jun 2008 00:19:42 -0700 - In this video, H
default registration process for a package, create
www.VisualStudioGallery.com website

How Do I: Integrate With the Error List?
Tue, 27 May 2008 22:59:55 -0700 - In this video,
items in this list and add items to this list.

How Do I: Add A Dialog Page to Configu
Tue, 20 May 2008 19:03:49 -0700 - In this video,
to allow users to configure settings on your Visual

Fig A.3 Most recently loaded or created projects.



New Project

Project types:
- Visual C++
  - ATL
  - CLR
  - General
  - MFC
  - Smart Device
  - Win32
- Other Languages
  - Visual C#
    - Windows
    - Smart Device
    - Database
    - Starter Kits
    - Web
    - XNA Game Studio 2.0
- Other Project Types

Templates:

Visual Studio installed templates

Custom Wizard	Windows Forms Application
CLR Console Application	Win32 Console Application
ATL Project	MFC Application
Makefile Project	ASP.NET Web Service
ATL Server Project	ATL Server Web Service
ATL Smart Device Project	Class Library
CLR Empty Project	Empty Project
MFC ActiveX Control	MFC DLL
MFC Smart Device ActiveX Control	MFC Smart Device Application
MFC Smart Device DLL	SQL Server Project
Win32 Project	Win32 Smart Device Project
Windows Forms Control Library	Windows Service

My Templates

A project for creating a Win32 console application

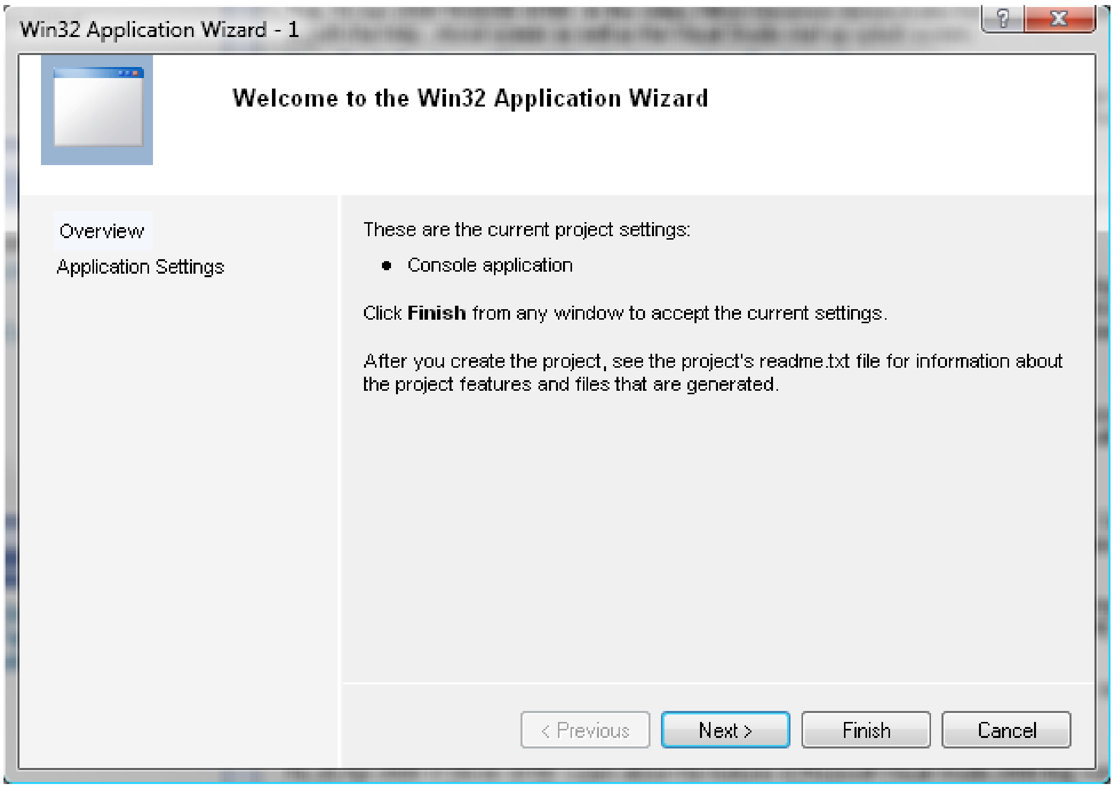
Name: <Enter_name>
Location: I:\Projects\Writing\Textbooks\3D Game Programming with DirectX 10 and OpenGL\Post Produ
Solution Name: <Enter_name>
☑ Create directory for solution

Fig A.4 Creating a new Visual Studio project – Step 1.

Your project selection tree may vary depending on the options selected during the Visual Studio installation. When installing Visual Studio 2005, I chose to install Microsoft Visual C++ and Microsoft Visual C# (with XNA later added).

You’ll have additional project creation options if you’ve installed other languages such as Microsoft Visual Basic.

For OpenGL/GLUT based programs, for example, you should create a “Win32 Console Application”, calling the program whatever you like (under “Name”) and saving it to any location (under “Location”). Once you’ve entered these values you can click on “OK” (Figure 1.5 shows the generated window).


Win32 Application Wizard - 1
Welcome to the Win32 Application Wizard
Overview
Application Settings
< Previous
Next >
Finish
Cancel

These are the current project settings:

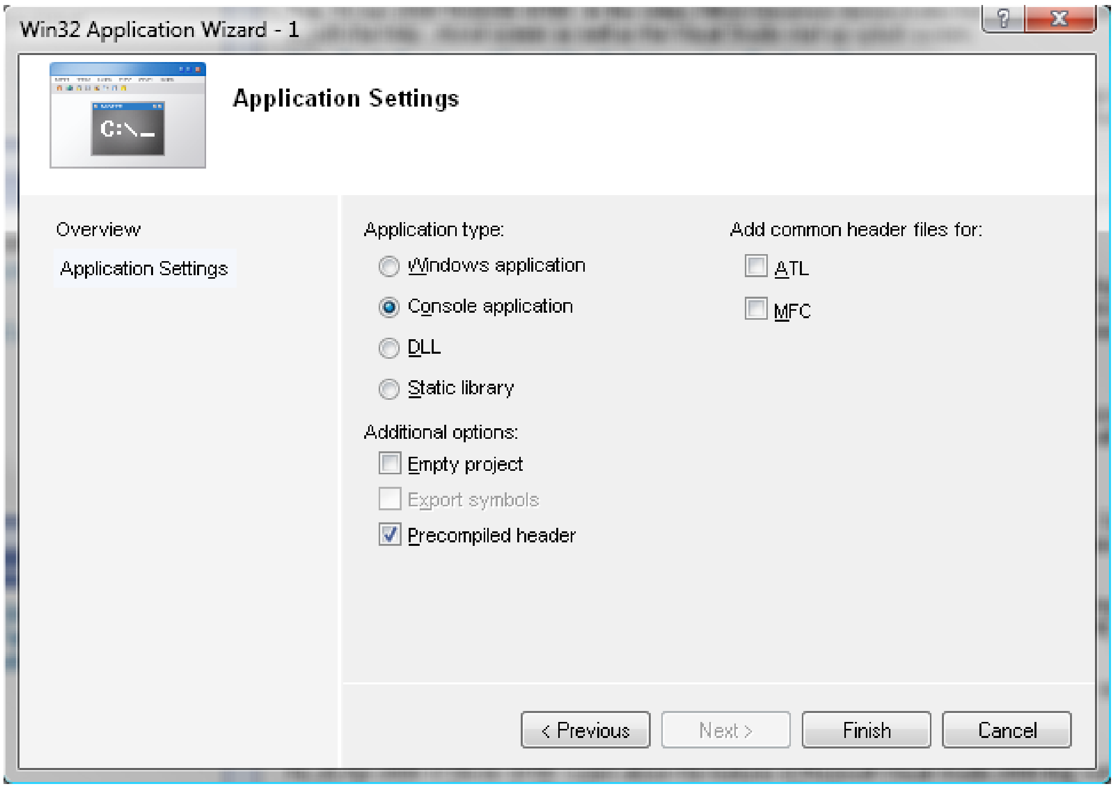
- Console application

Click **Finish** from any window to accept the current settings.

After you create the project, see the project's readme.txt file for information about the project features and files that are generated.

Fig A.5 Creating a new Visual Studio project – Step 2.

The final thing to do is click on the “Next” button followed by selection of the “Empty project” checkbox (Figure A.6). You can now click on “Finish”.


Win32 Application Wizard - 1

Application Settings

Overview

Application Settings

Application type:

- Windows application
- [✓ Console application]
- DLL
- Static library

Add common header files for:

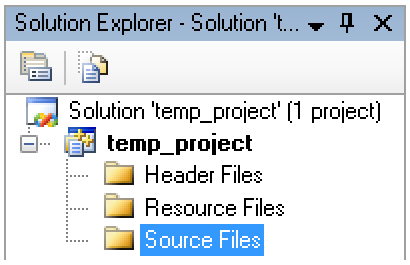
- ☐ ATL
- ☐ MFC

Additional options:

- ☐ Empty project
- ☐ Export symbols
- ☑ Precompiled header

Fig A.6 Creating a new Visual Studio project – Step 3.

You will now have an empty project similar to the one depicted by Figure A.7. You can either include already existing files into your project (Figure A.8a, A.8b) or create new source files for it (Figure A.9a, A.9b).


Solution Explorer - Solution 't...

Solution 'temp_project' (1 project)

- **temp_project**
  - Header Files
  - Resource Files
  - Source Files

Fig A.7 The newly created project.

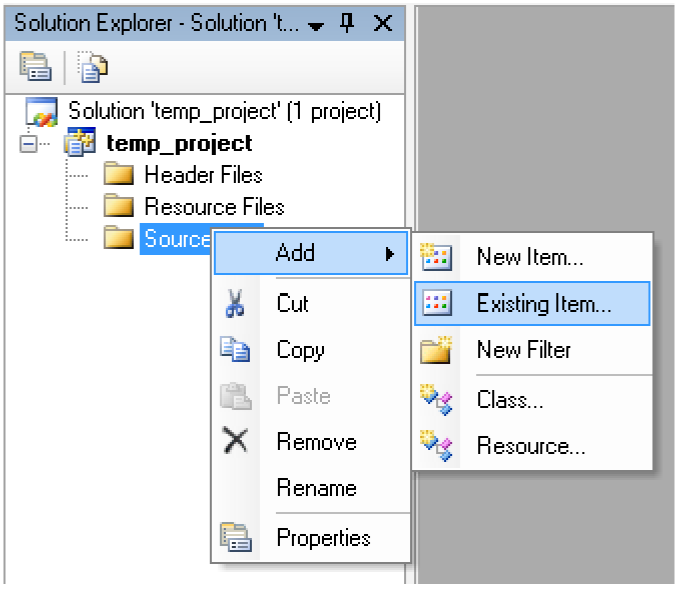


Fig A.8a Adding an existing source file to your project.

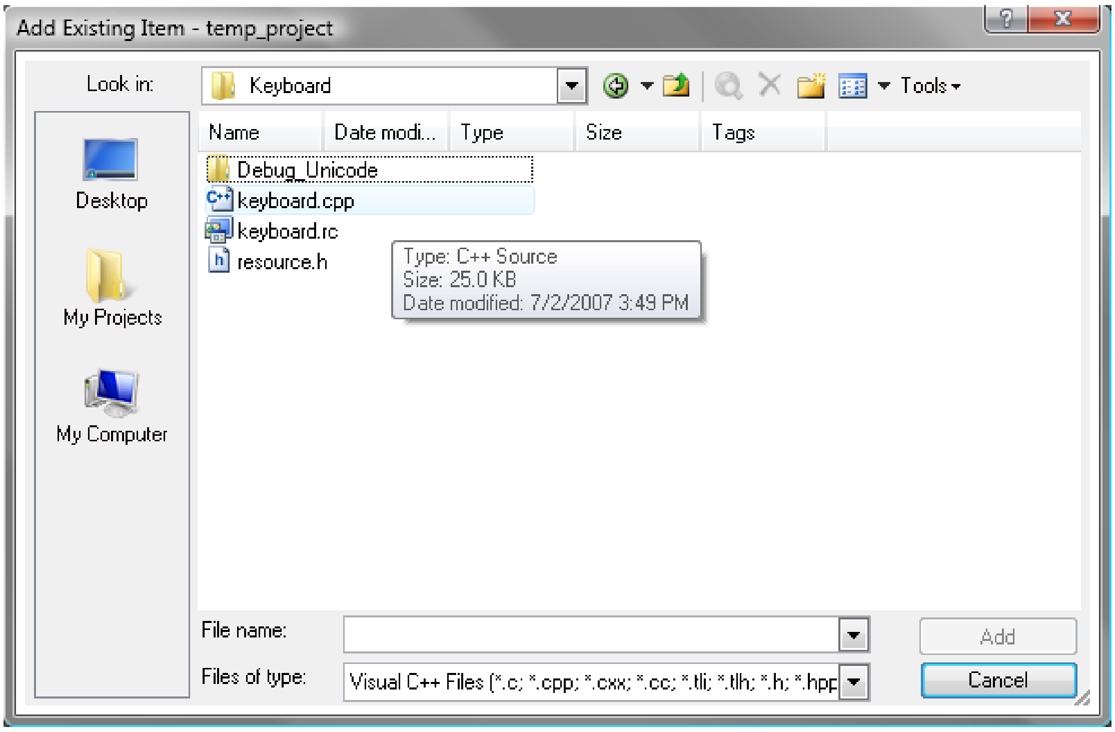


Fig A.8b Selecting existing files to add to the project.

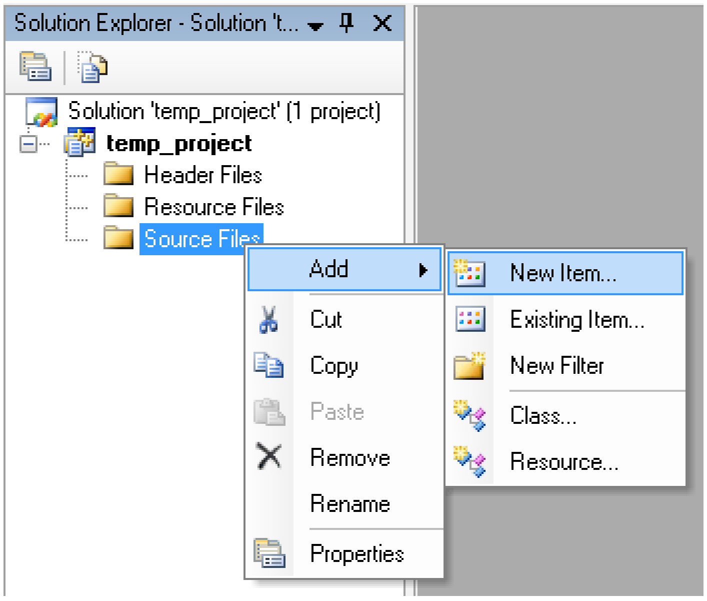
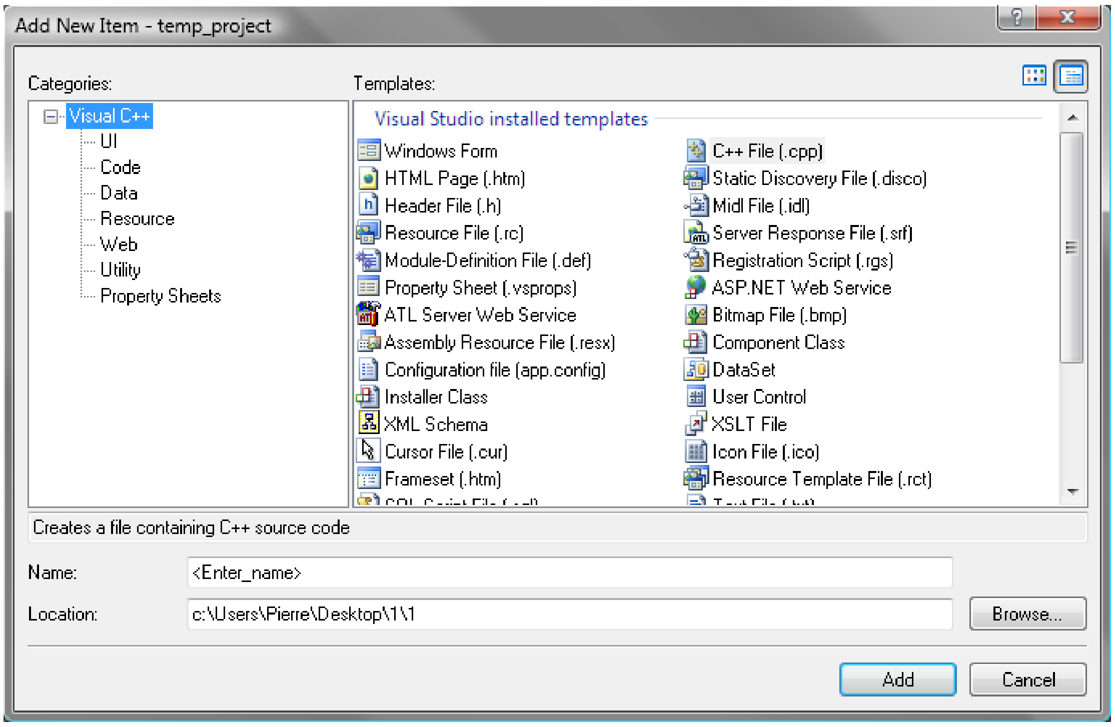


Fig A.9a Adding a new source file to your project.



Creates a file containing C++ source code

Name: <Enter_name>

Location: c:\Users\Pierre\Desktop\1\1

Fig A.9a Selecting the type of file and its name.

Once you have created/imported your application’s source files, you’re ready to compile the code (following the appropriate setup of Visual Studio’s project properties). In addition to compiling entire applications, Visual Studio also offers the option of compiling a selected source file – a very useful feature. Figure A.10a and A.10b illustrates this feature. Clicking on the source file with the right mouse button will yield the illustrated options.

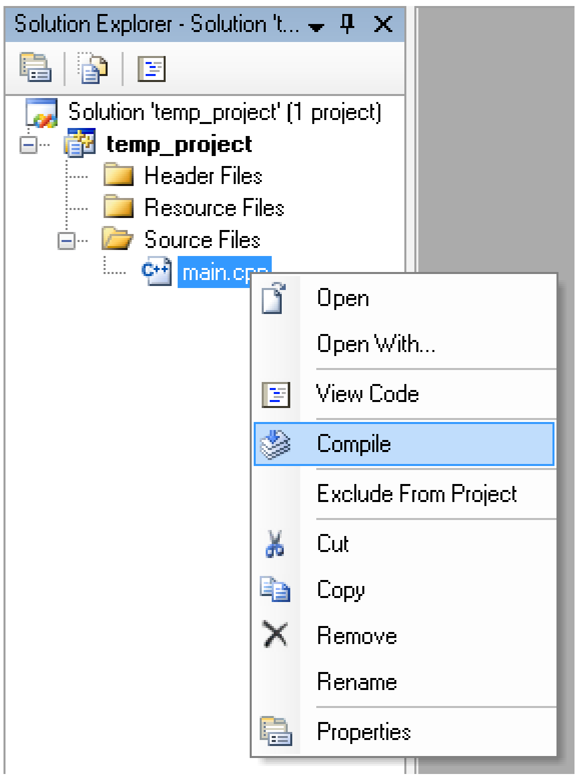


Fig A.10a Compiling a single source file.

```
Compiling...
main.cpp

temp_project - 0 error(s), 0 warning(s)
```

Fig A.10b Output after compiling a single source file.

When you're ready to build your application (i.e. create the executable) you can either click the little "Start" icon (Figure A.11a) or access the "Build" options by clicking on "Build" in the toolbar (Figure A.11b). Clicking the "Start" icon will cause your application to launch after it has been compiled and linked.

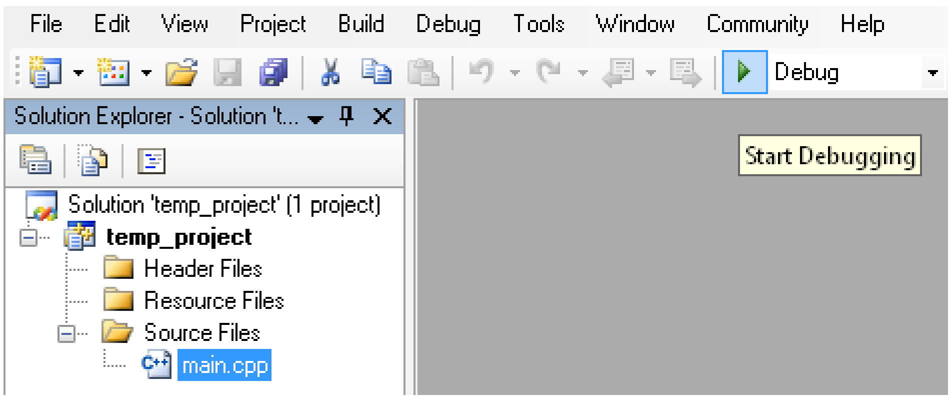


Fig A.11a Building and executing your application.

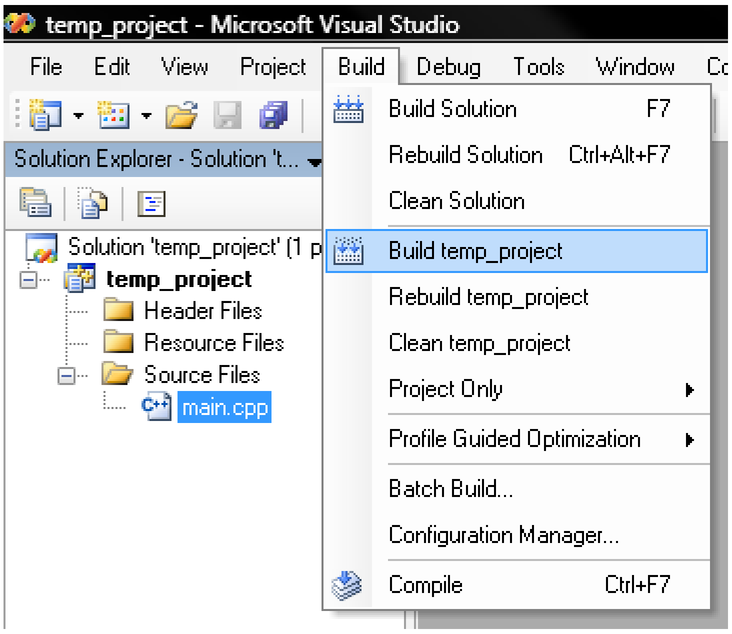


Fig A.11b Building your application without executing it.

Microsoft Visual Studio 2005 Project Properties

Before compiling a project, you'll need to configure Visual Studio's linker. To do this, open your project's "Properties" dialog (Figure A.12).

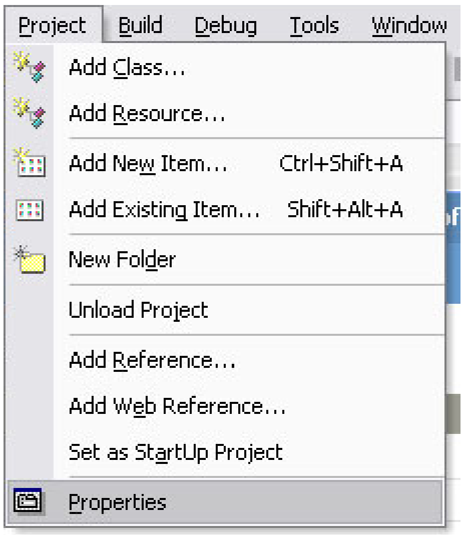
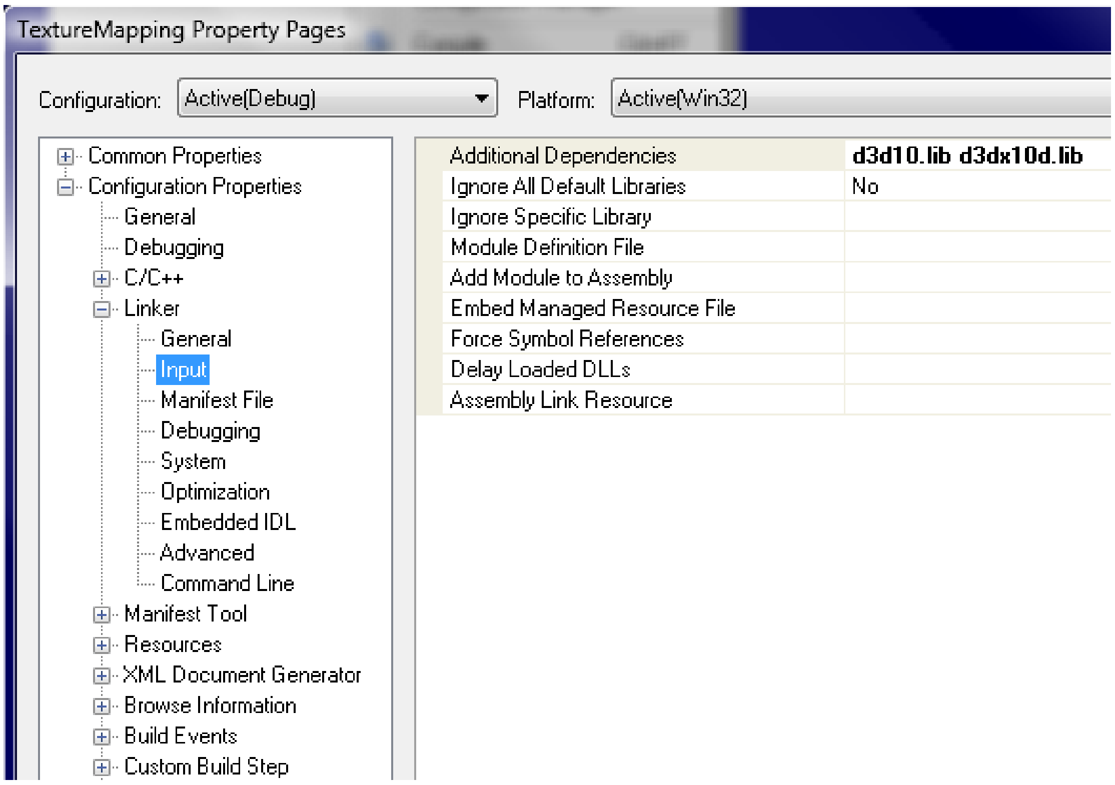


Fig A.12 Accessing the Properties dialog.

Now select the "Linker" sub-tree node followed by the "Input" option. Here you'll include all the necessary OpenGL/GLUT/DirectX/etc libraries (shown in Figure A.13a for basic DirectX functionality and Figure A13b for OpenGL).



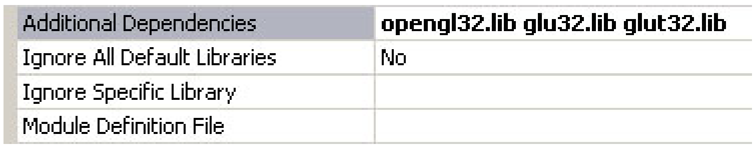
TextureMapping Property Pages

Configuration: Active(Debug)   Platform: Active(Win32)

- Common Properties
- Configuration Properties
  - General
  - Debugging
  - C/C++
  - Linker
    - General
    - Input
    - Manifest File
    - Debugging
    - System
    - Optimization
    - Embedded IDL
    - Advanced
    - Command Line
  - Manifest Tool
  - Resources
  - XML Document Generator
  - Browse Information
  - Build Events
  - Custom Build Step

Additional Dependencies	**d3d10.lib d3dx10d.lib**
Ignore All Default Libraries	No
Ignore Specific Library	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded DLLs	
Assembly Link Resource	

Fig A.13a Basic Linker → Input setting for a DirectX application.



Additional Dependencies	**opengl32.lib glu32.lib glut32.lib**
Ignore All Default Libraries	No
Ignore Specific Library	
Module Definition File	

Fig A.13a Basic Linker → Input setting for an OpenGL application.