

## Manual: OpenGL/C++ Projects in Eclipse/CDT

*Important:* The example applications for the CGI exercises are provided as Eclipse projects, offering pre-defined build configurations for Linux (*Debug* and *Release*) and Windows (*DebugWin* and *ReleaseWin*). The following information is most relevant for creating new projects.

A few global workspace settings, however, are recommended.

- Project/Build automatically:  
Disable
- Window/Preferences/General/Editors/File Associations:  
File types = \*.glsl, Associated editors = C/C++ Editor (add 1 entry)
- Window/Preferences/C/C++/Build/Environment:  
LC\_Messages = en\_US.UTF-8 (add one entry)
- Window/Preferences/C/C++/Code Analysis:  
Disable all sections

The following Eclipse settings have to be adjusted for the build configurations *Debug* and *Release* of new projects. Using the *All configurations* option may be helpful.

### Linux

*Important:* The provided Eclipse projects offer pre-defined Linux build configurations *Debug* and *Release*.

The following software environment is assumed, as provided in the graphics lab and the PC pools.

- Ubuntu 12.04 (at least Ubuntu 11.04), GCC/G++ 4.6 (at least 4.5), Eclipse 3.7 with CDT 8.0 (Eclipse 4.2 with CDT has bugs);
- OpenGL 3.2 (or higher), GLEW 1.6 (or higher), GLFW 3.0.2 (or higher)
- Ubuntu packages libxxf86vm-dev, libxrandr-dev, libxi-dev, xinput

*Note:* GLFW 3 has to be built from source using CMake (cf. <http://www.glfw.org/>). The GLFW 2.7 packages provided by current Ubuntu distributions are outdated.

### Stand-alone Application (Linux)

Example: `opengl3_simple`

Note: GLEW 1.6 (or higher) and GLFW 3.0.2 (or higher) must be installed.

Creation: New C++ project/Executable/Empty Project

Project/Properties:

- C/C++ Build
  - Tool Chain Editor:

Current toolchain = Linux GCC, Current builder = Gnu Make Builder

- Settings/Tool Settings/GCC C++ Compiler/Miscellaneous:  
Other flags = ... -std=c++0x (add if C++11 is used)
- Settings/Tool Settings/GCC C++ Linker/Libraries:  
Libraries = GLEW, glfw3, GL, X11, Xxf86vm, Xrandr, pthread, Xi, rt (add 9 entries)
- Settings/Build Artifact:  
Artifact extension = <empty>
- Settings/Binary Parsers:  
Binary parser = Elf Parser

### Static Library (Linux)

Example: `vascg3` with GLEW included as source code

Note: GLFW 3.0.2 (or higher) must be installed.

Creation: New C++ project/Static Library/Empty Project

Project/Properties:

- C/C++ Build
  - Tool Chain Editor:  
Current toolchain = Linux GCC, Current builder = Gnu Make Builder
  - Settings/Tool Settings/GCC C++ Compiler/Miscellaneous:  
Other flags = ... -std=c++0x (add if C++11 is used)
  - Settings/Build Artifact:  
Artifact extension = a, Output prefix = lib
  - Settings/Binary Parsers:  
Binary parser = Elf Parser

### Application Using a Static Library (Linux)

Example: `vascg3_example` using `vascg3`

Creation: New C++ project/Executable/Empty Project

Project/Properties:

- C/C++ Build
  - Tool Chain Editor:  
Current toolchain = Linux GCC, Current builder = Gnu Make Builder
  - Settings/Tool Settings/GCC C++ Compiler/Includes:  
Include paths = "\${workspace\_loc:/vascg3}" (add 1 entry)

- Settings/Tool Settings/GCC C++ Compiler/Miscellaneous:  
Other flags = ... -std=c++0x (add if C++11 is used)
- Settings/Tool Settings/GCC C++ Linker/Libraries:  
Libraries = vascg3, glfw3, GL, X11, Xxf86vm, Xrandr, pthread, Xi, rt (add 9 entries)  
Library search path = "\${workspace\_loc:/vascg3/Debug}" (for Debug configuration)  
Library search path = "\${workspace\_loc:/vascg3/Release}" (for Release configuration)
- Settings/Build Artifact:  
Artifact extension = <empty>
- Settings/Binary Parsers:  
Binary parser = Elf Parser
- Project References:  
Select vascg3

## Windows

*Important:* The provided Eclipse projects offer pre-defined Windows build configurations *DebugWin* and *ReleaseWin*.

The following software environment is assumed.

- Windows 7 (XP or Vista might work), MinGW/MSYS with GCC/G++ 4.6 (at least 4.5), Eclipse 3.7 (32 bit) with CDT 8.0 (Eclipse 4.2 with CDT has bugs; 64 bit Eclipse has problems with 32 bit MinGW);
- OpenGL 3.2 (or higher), GLEW 1.6 (or higher), GLFW 3.0.2 (or higher)

The `bin` directories of MinGW and MSYS have to be added to the environment variable `PATH`.

*Note:* GLEW and GLFW 3 have to be built from source or installed as pre-compiled Windows binaries (cf. <http://glew.sourceforge.net/> and <http://www.glfw.org/>, respectively).

The library `vascg3` and the application `vascg3_example` have also been tested with Microsoft Visual Studio 10/2010 and 11/2012; solution and project files can be provided upon request.

## Stand-alone Application (Windows)

Example: `opengl3_simple`

Note: GLEW 1.6 (or higher) and GLFW 3.0.2 (or higher) must be installed.

Creation: New C++ project/Executable/Empty Project

Project/Properties:

- C/C++ Build
  - Tool Chain Editor:  
Current toolchain = MinGW GCC, Current builder = Gnu Make Builder
  - Settings/Tool Settings/GCC C++ Compiler/Miscellaneous:

Other flags = ... -std=c++0x (add if C++11 is used)

- Settings/Tool Settings/GCC C++ Linker/Libraries:  
Libraries = glew32, glfw3, opengl32, gdi32 (add 4 entries, glew32 has to appear before other OpenGL libraries)
- Settings/Build Artifact:  
Artifact extension = exe
- Settings/Binary Parsers:  
Binary parser = PE Windows Parser

### Static Library (Windows)

Example: `vascg3` with GLEW included as source code

Note: GLFW 3.0.2 (or higher) must be installed.

Creation: New C++ project/Static Library/Empty Project

Project/Properties:

- C/C++ Build
  - Tool Chain Editor:  
Current toolchain = MinGW GCC, Current builder = Gnu Make Builder
  - Settings/Tool Settings/GCC C++ Compiler/Miscellaneous:  
Other flags = ... -std=c++0x (add if C++11 is used)
  - Settings/Build Artifact:  
Artifact extension = a, Output prefix = lib
  - Settings/Binary Parsers:  
Binary parser = PE Windows Parser

### Application Using a Static Library (Windows)

Example: `vascg3_example` using `vascg3`

Creation: New C++ project/Executable/Empty Project

Project/Properties:

- C/C++ Build
  - Tool Chain Editor:  
Current toolchain = MinGW GCC, Current builder = Gnu Make Builder
  - Settings/Tool Settings/GCC C++ Compiler/Includes:  
Include paths = "\${workspace\_loc:/vascg3}" (add 1 entry)
  - Settings/Tool Settings/GCC C++ Compiler/Miscellaneous:  
Other flags = ... -std=c++0x (add if C++11 is used)

- Settings/Tool Settings/GCC C++ Linker/Libraries:
  - Libraries = vascg3, glfw3, opengl32, gdi32 (add 4 entries)
  - Library search path = "\${workspace\_loc:/vascg3/DebugWin}" (for DebugWin configuration)
  - Library search path = "\${workspace\_loc:/vascg3/ReleaseWin}" (for ReleaseWin configuration)
- Settings/Build Artifact:
  - Artifact extension = exe
- Settings/Binary Parsers:
  - Binary parser = PE Windows Parser
- Project References:
  - Select vascg3