

How to make custom build rules in MSVS 2008 SP1

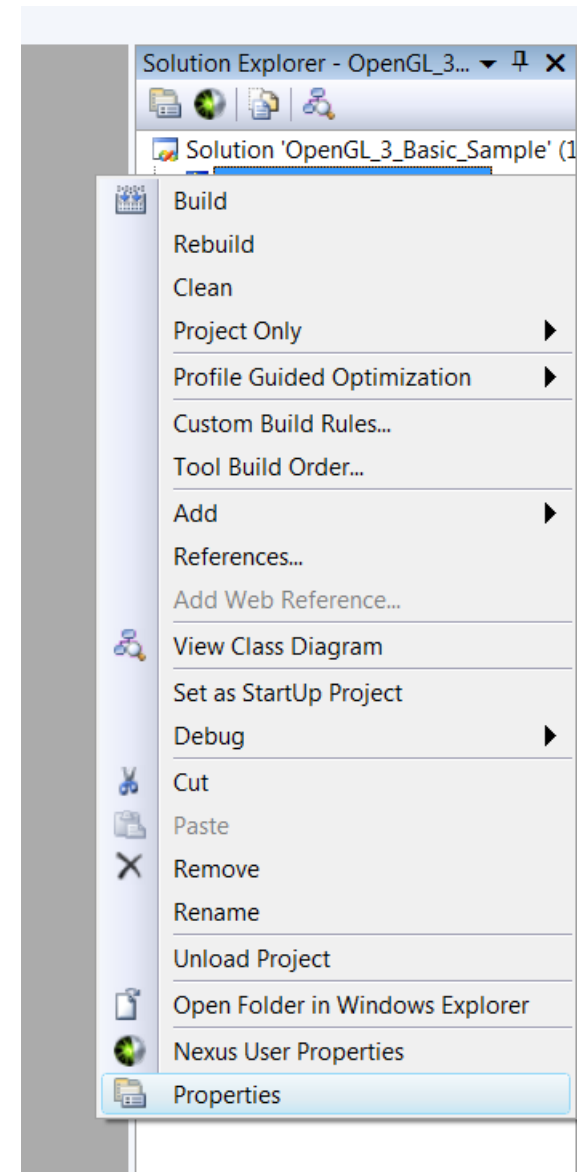
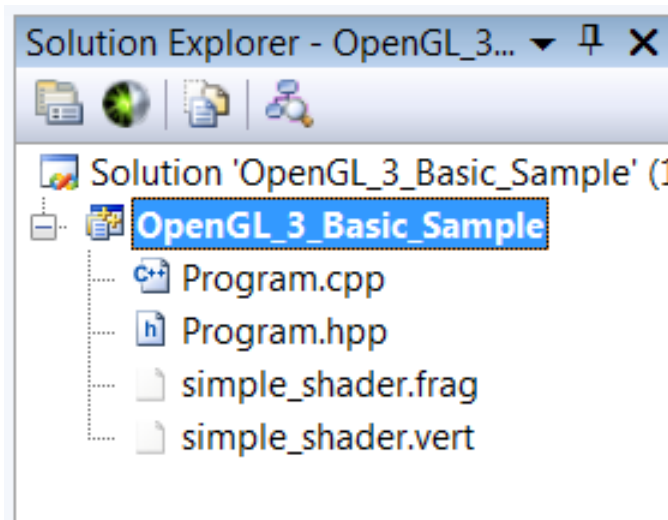
CG2LU-Team 2010

Institute of Computer Graphics and Algorithms

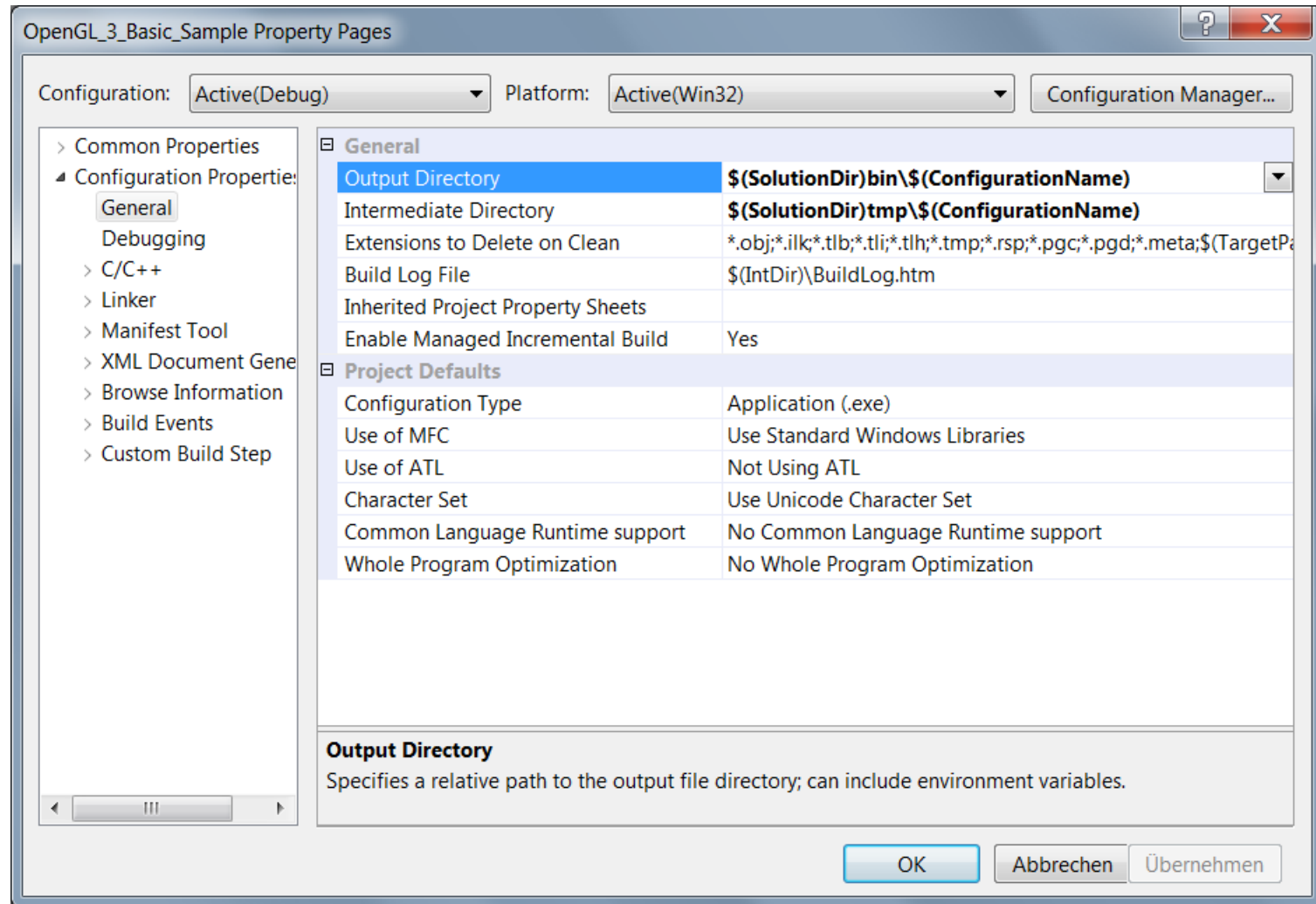
Vienna University of Technology



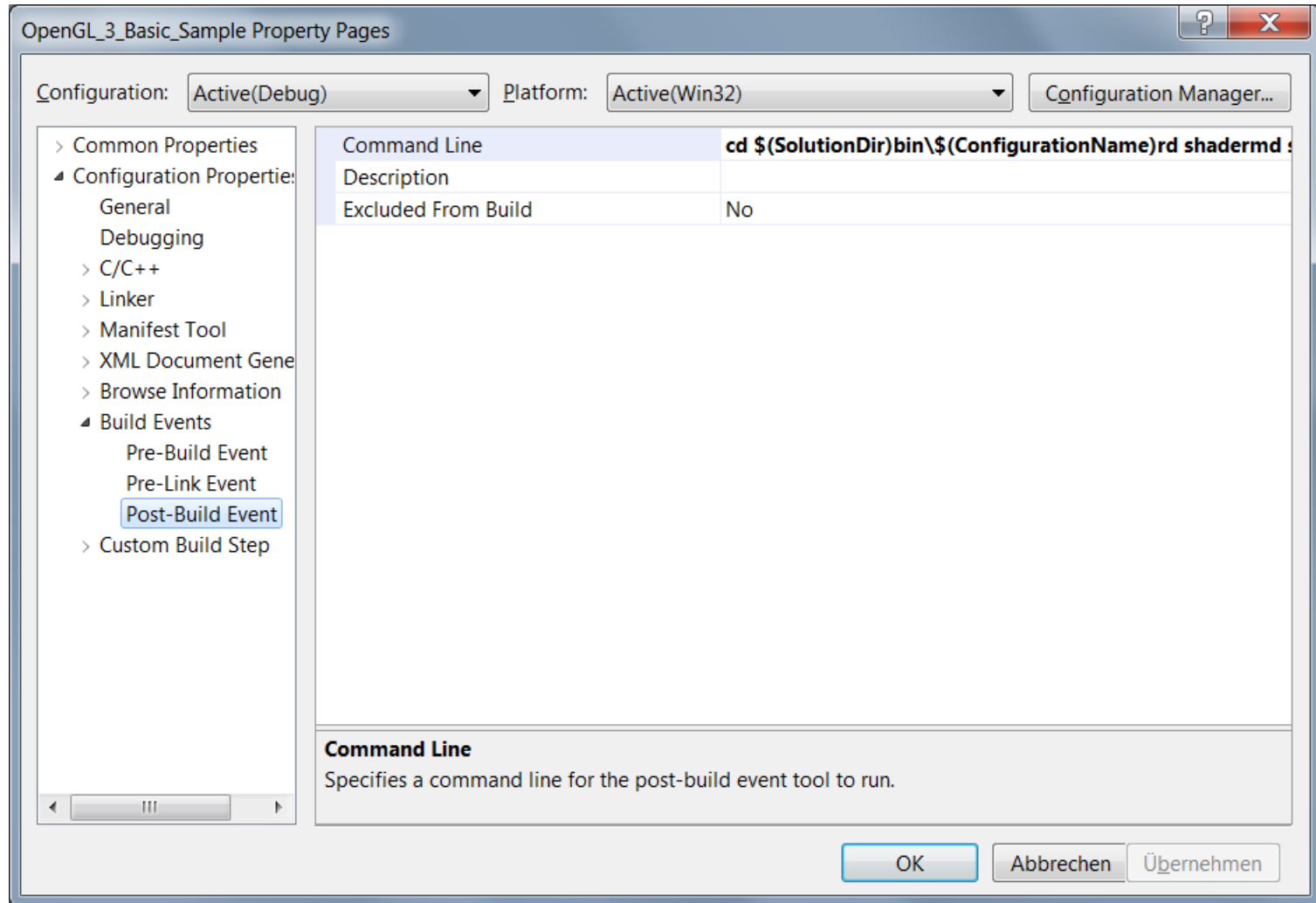
- 1. Add the Shader-Files to your project
- 2. Open the Project-Properties



■ Properties might look like this:



- Go to *Build Events* → *Post-Build-Events*



- In “Command Line” enter the command to be executed.
- For example:

```
cd $(SolutionDir)bin\$(ConfigurationName)
rd shader
md shader
copy "$(SolutionDir)shader\simple_shader.frag" "$(SolutionDir)\bin\$(ConfigurationName)\shader\simple_shader.frag"
copy "$(SolutionDir)shader\simple_shader.vert" "$(SolutionDir)\bin\$(ConfigurationName)\shader\simple_shader.vert"
```

- ◆ This will copy the two given shader-files *every* time you do a build to the given directory.



- To enable Syntax-Highlighting for GLSL-Shader (endings *.vert, *.frag) in Visual Studio, use NShader [1]

```
simple_shader.vert simple_shader.frag

#version 130 // delete this line if using GLSL 1.2
precision mediump float; // delete this line if using GLSL 1.2

// uniform shader-parameters
uniform mat4 perspective;
uniform mat4 view;
uniform mat4 model;

// vertex-shader input variables
in vec4 vertex;
in vec4 color;
in vec3 normal;

// vertex-shader output variables (passed to fragment-shader)
out vec4 frag_color;
out vec3 world_normal;
out vec4 world_position;

void main()
{
    // transform normal into world space
    vec4 normal4 = vec4(normal, 0.0f);
    world_normal = (model * normal4).xyz;

    // transform vertex to world-space
    world_position = model * vertex;

    // just pass color to fragment-shader
    frag_color = color;

    // transform vertex down the pipeline
    gl_Position = perspective * view * model * vertex;
}
```



- [1] NShader, <http://nshader.codeplex.com/>

