

This document describes how to use the OSCLib shared libraries in an eclipse CDT project.

Go to File -> New -> C++ Project. In the Project wizard enter a project name and select an executable. Then choose "Linux GCC Toolchain":

The screenshot shows the 'C++ Project' wizard in Eclipse. The title bar says 'C++ Project'. Below the title, it says 'C++ Project' and 'Create C++ project of selected type'. The 'Project name' field contains 'ExampleProject'. The 'Use default location' checkbox is checked. The 'Location' field shows '/home/martin/workspace\_OSCLib/ExampleProject' with a 'Browse...' button. The 'Choose file system' dropdown is set to 'default'. The 'Project type' list on the left has 'Executable' expanded, with 'Empty Project' selected. The 'Toolchains' list on the right has 'Linux GCC' highlighted with a red rectangle. At the bottom, the 'Show project types and toolchains only if they are supported on the platform' checkbox is checked. Navigation buttons at the bottom include a help icon, '< Back', 'Next >', 'Cancel', and 'Finish'.

**C++ Project**  
Create C++ project of selected type

Project name:

☒ Use default location

Location:

Choose file system:

Project type:

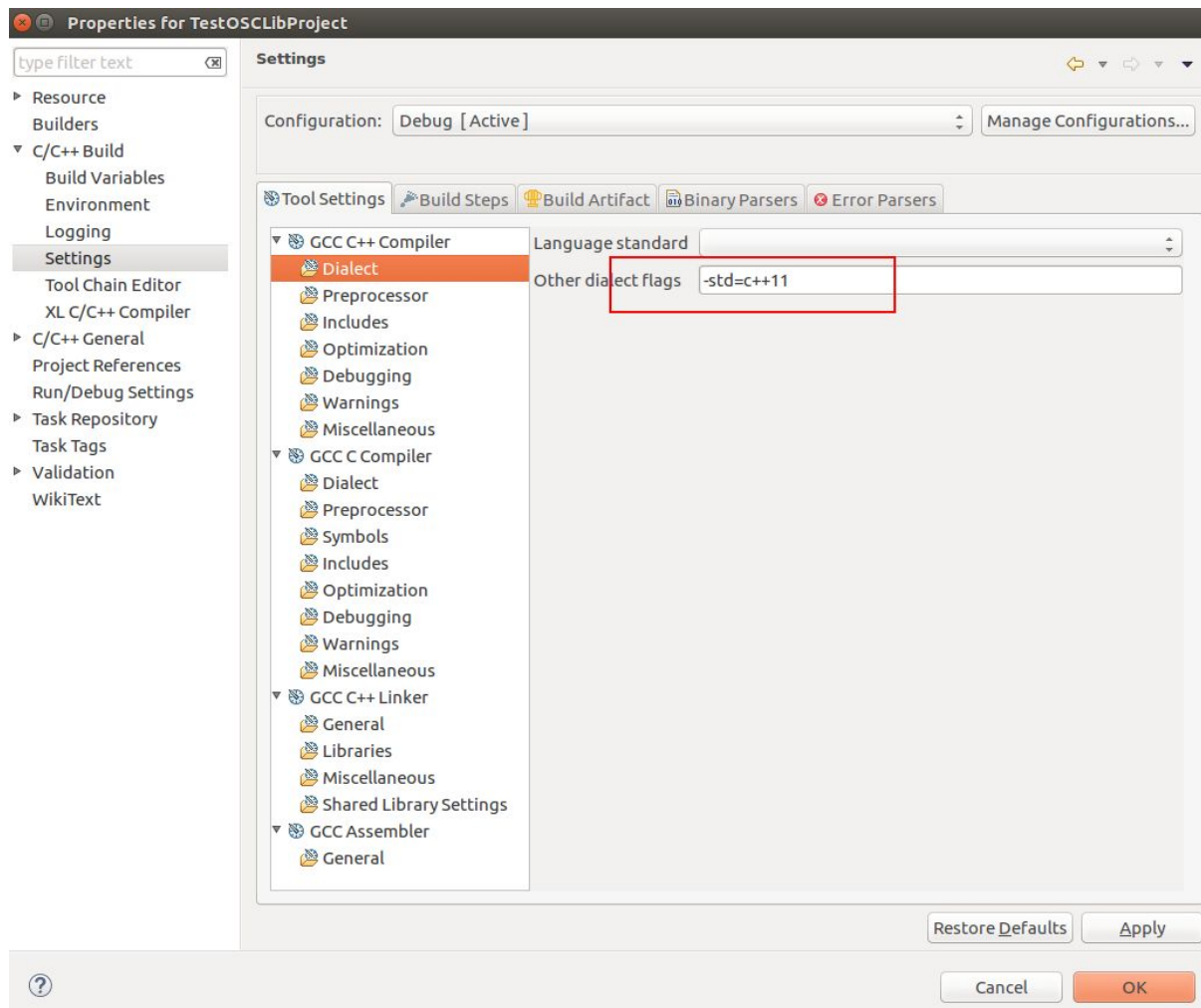
- ▶ GNU Autotools
- ▼ Executable
  - Empty Project
  - Hello World UPC Project
  - Hello World C++ Project
- ▶ Shared Library
- ▶ Static Library
- ▶ Others
- ▶ Makefile project

Toolchains:

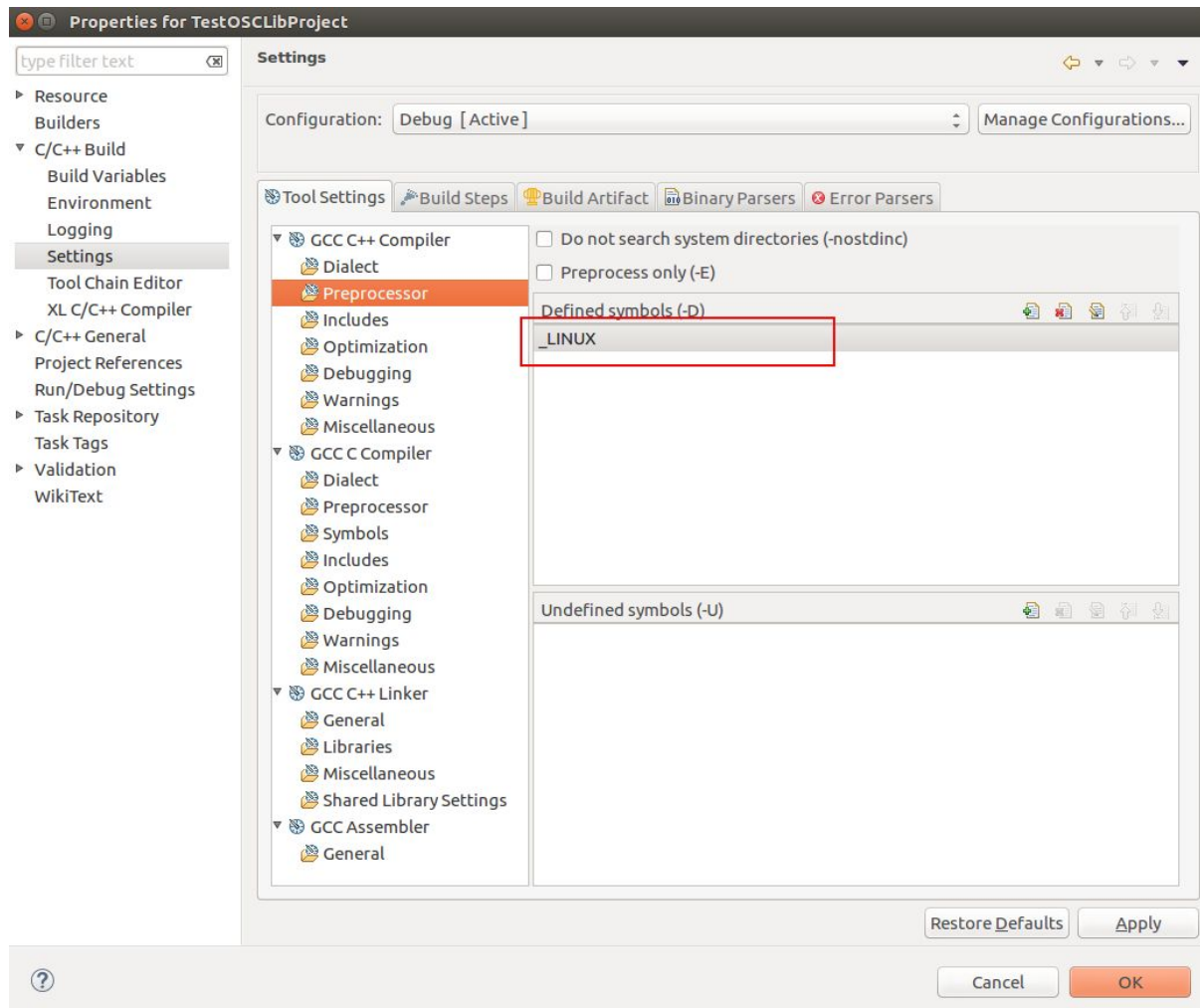
- Cross GCC
- Linux GCC**
- XL C/C++ Tool Chain

☒ Show project types and toolchains only if they are supported on the platform

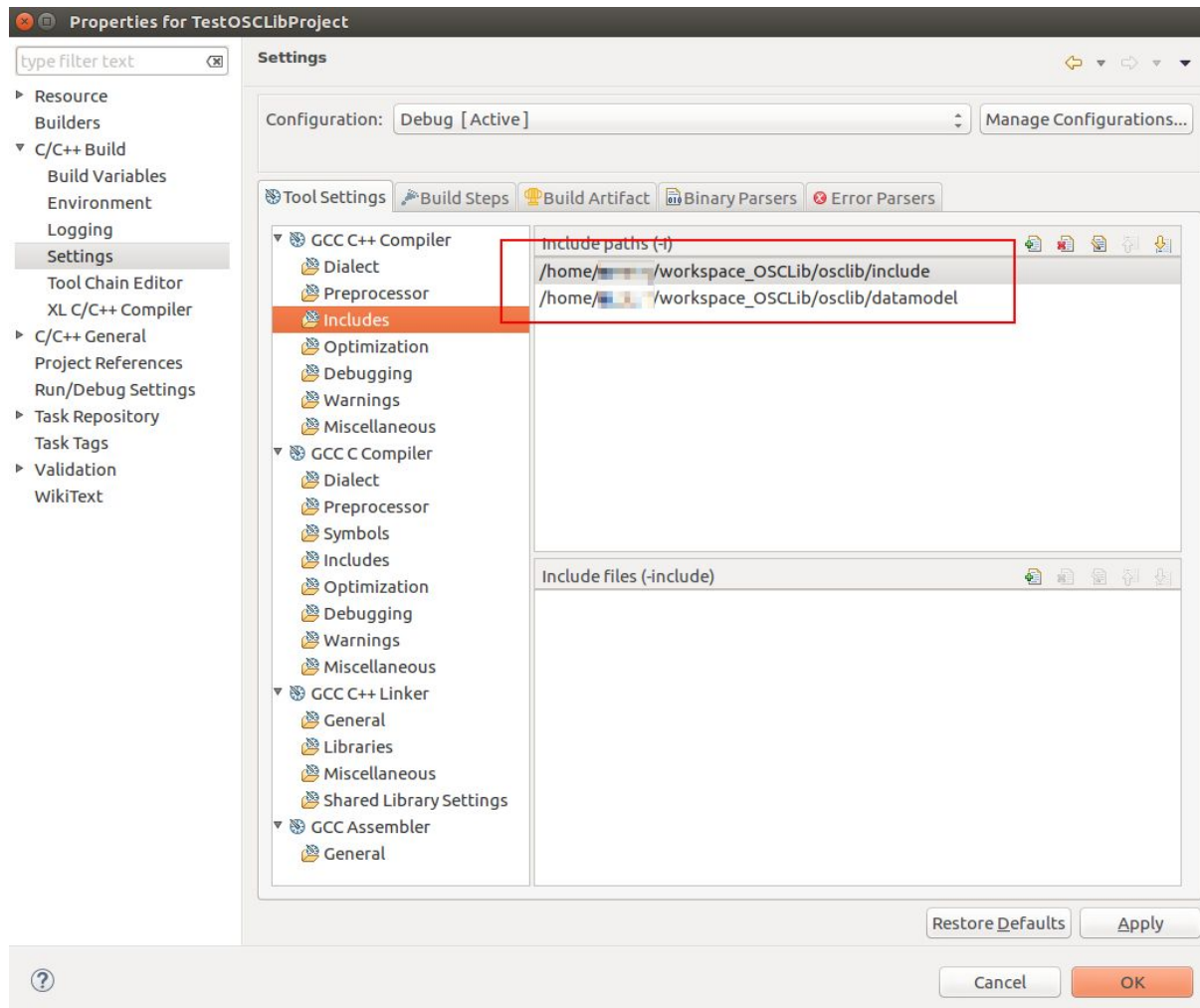
In the project properties, go to “C/C++ Build and select “Settings”. Then select the “GCC C++ Compiler Dialect” node in the Tool Settings tab. Under “Other dialect flags” enter “-std=c++11”:



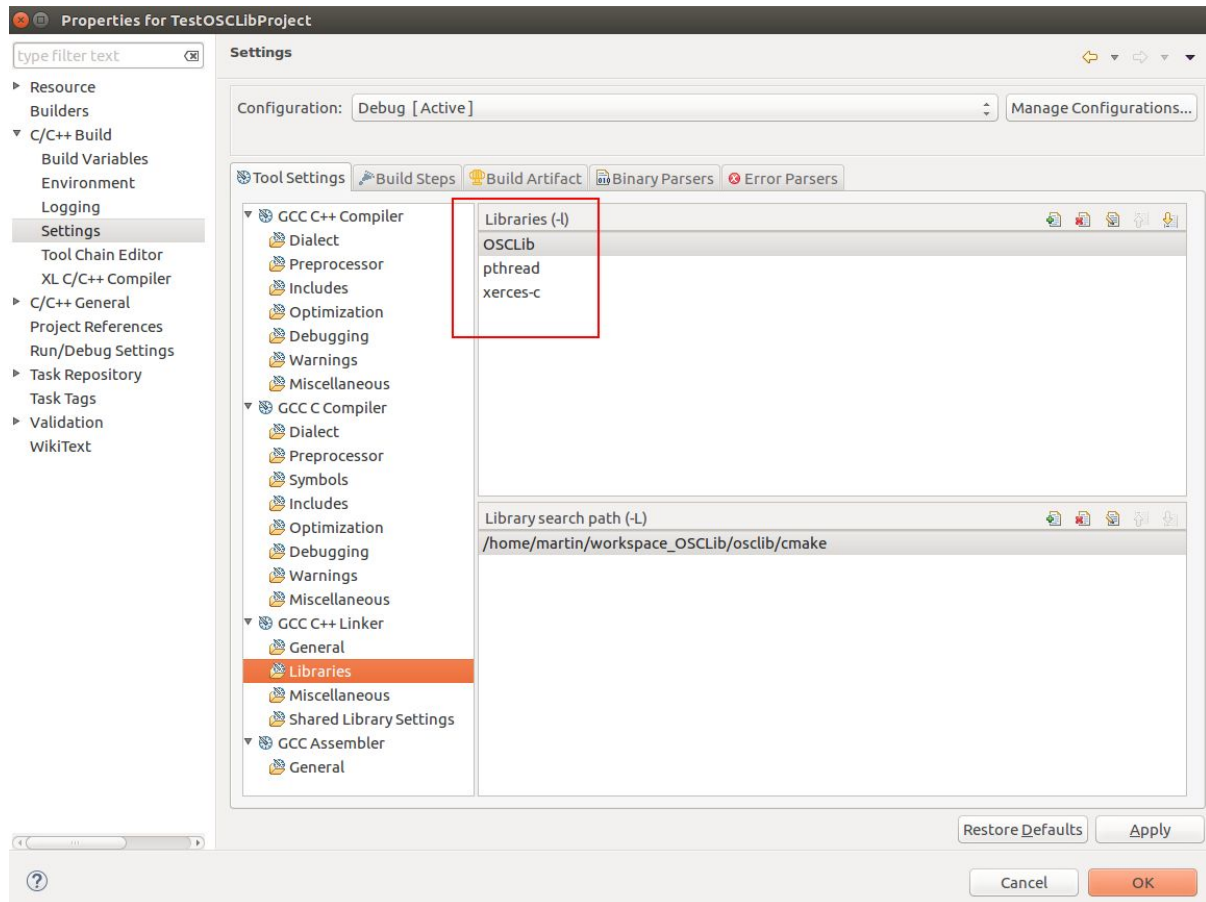
Select the “Preprocessor” node and enter the defined symbol “\_LINUX”:



Under “Includes” add the two directory entries “include” and “datamodel”:



Select the “GCC C++ Linker Libraries” node and add the entries “OSCLib”, “pthread” and “xerces-c”:



In the Run / Debug configurations, select the “Environment” tab and add the variable “LD\_LIBRARY\_PATH”. Then insert the path to the shared object (libOSCLib.so) as value:

