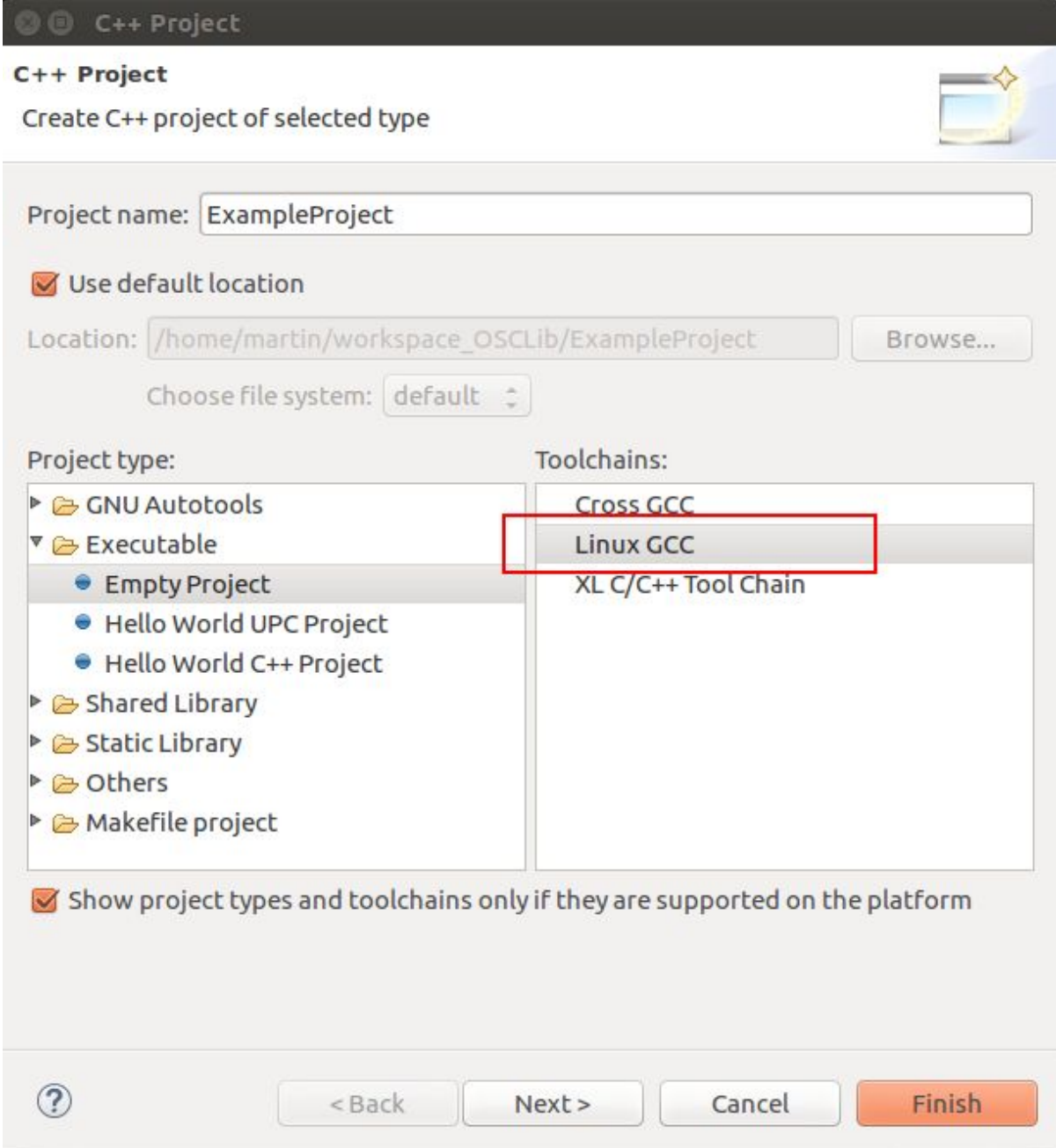


This document describes how to use the OSCLib shared libraries in an eclipse CDT project.

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



The image shows the Eclipse C++ Project Wizard dialog. The title bar says "C++ Project". Below the title bar, it says "C++ Project" and "Create C++ project of selected type". There is a "Project name:" field with the text "ExampleProject". Below that, there is a checked checkbox "Use default location". The "Location:" field shows the path "/home/martin/workspace_OSCLib/ExampleProject" and a "Browse..." button. Below the location field, there is a "Choose file system:" dropdown menu with "default" selected. The "Project type:" section shows a tree view with "Executable" selected. The "Toolchains:" section shows a list with "Linux GCC" selected, which is highlighted by a red rectangle. At the bottom, there is a checked checkbox "Show project types and toolchains only if they are supported on the platform". The bottom of the dialog has a question mark icon, "< Back", "Next >", "Cancel", and "Finish" buttons.

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

Project name:

☒ Use default location

Location:

Choose file system:

Project type:

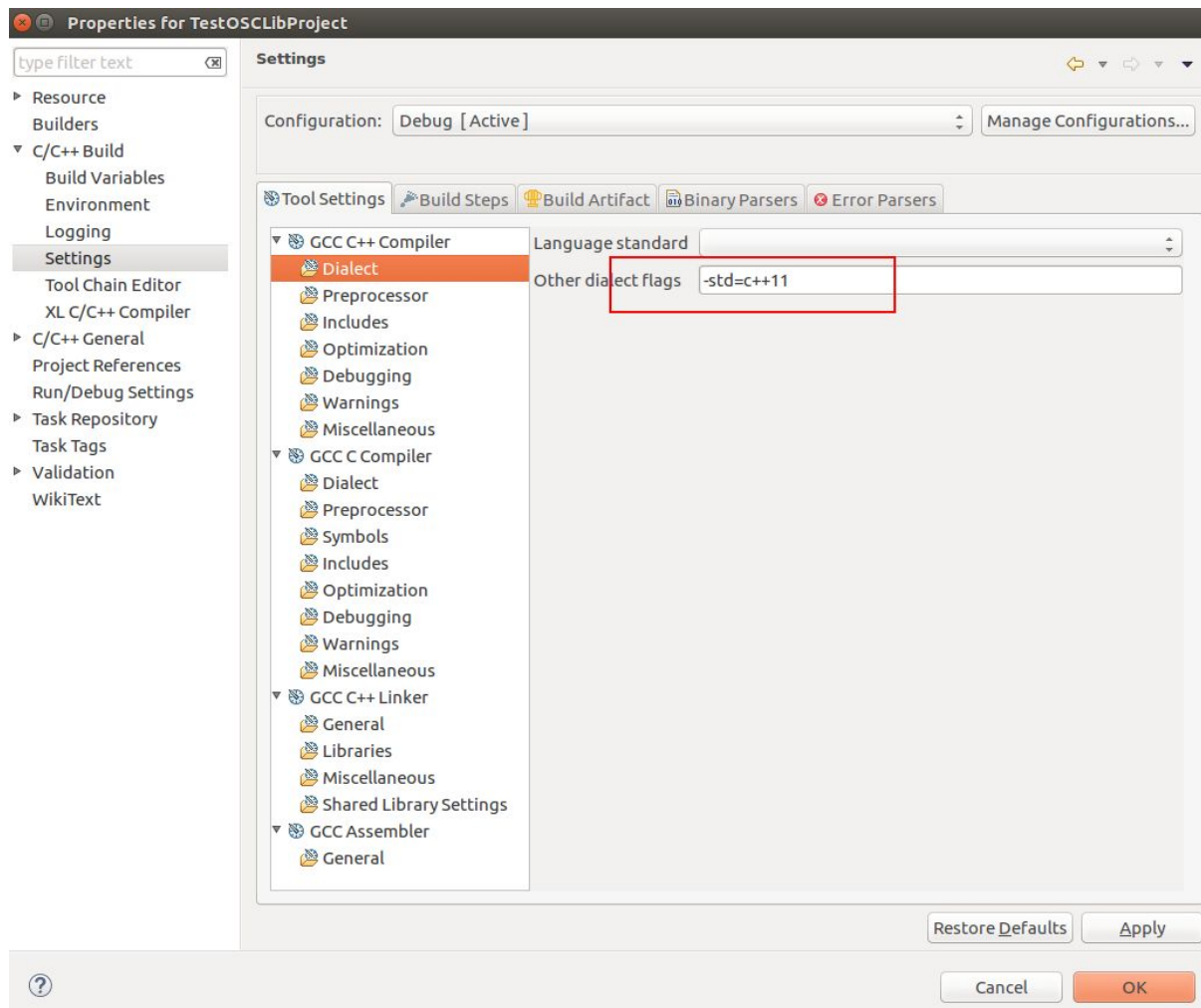
- GNU Autotools
- Executable
 - Empty Project
 - Hello World UPC Project
 - Hello World C++ Project
- Shared Library
- Static Library
- Others
- Makefile project

Toolchains:

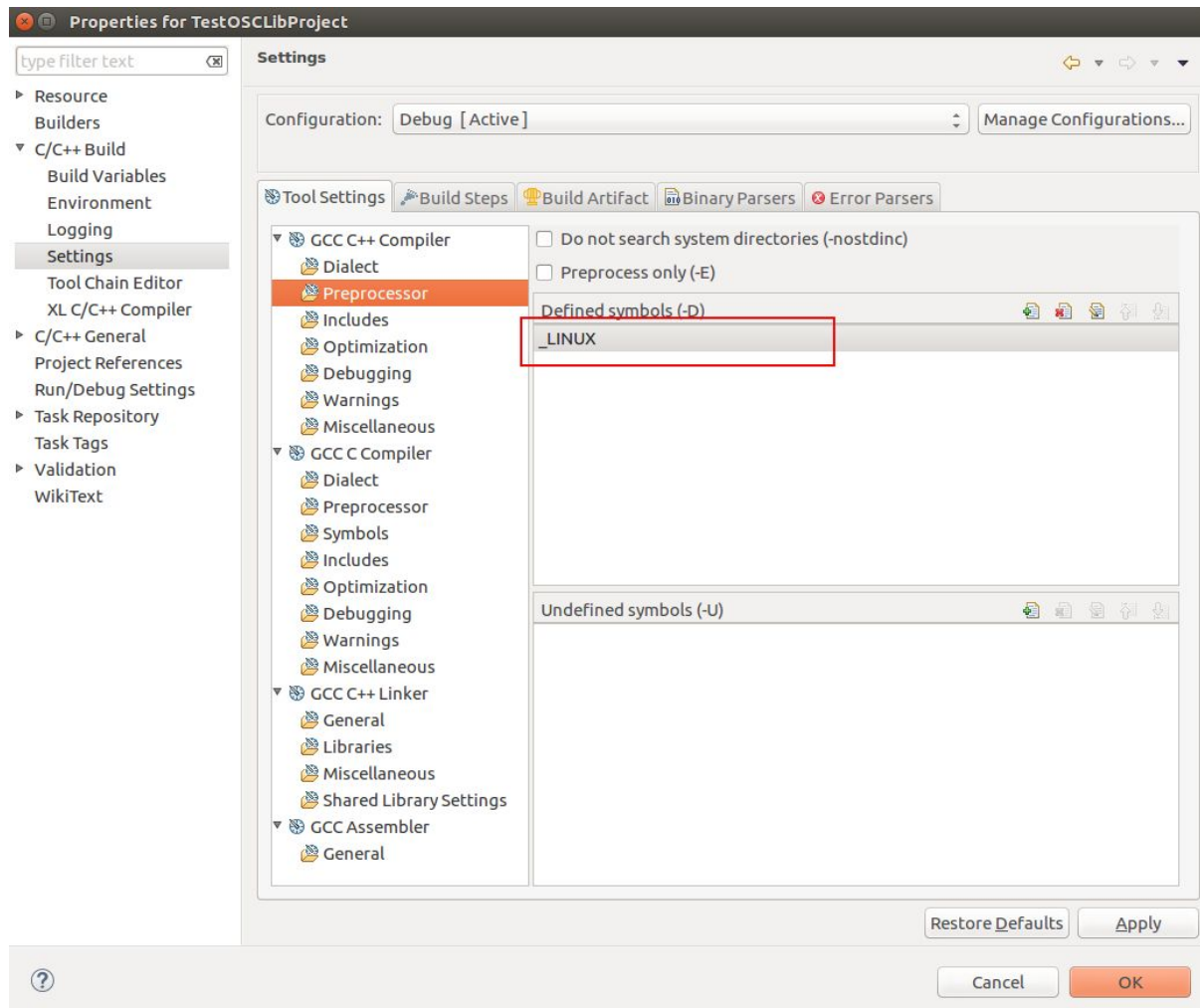
- Cross GCC
- Linux GCC**
- XL C/C++ Tool Chain

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

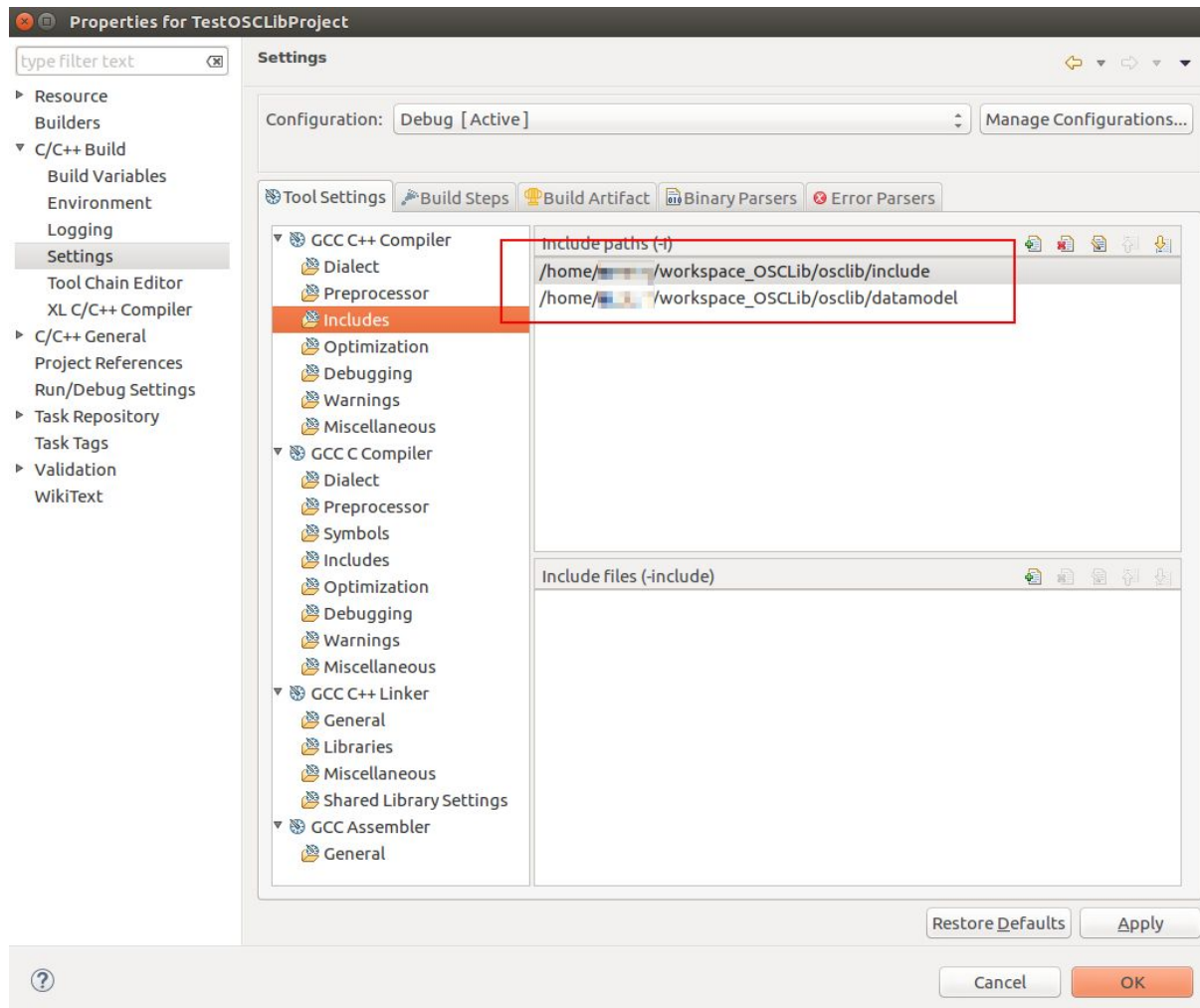
In the project properties, go to “C/C++ Build and select “Settings”. Then select the “GCC C++ Compiler Dialect” node in the Tool Settings tab. Under “Other dialect flags” enter “-std=c++11”:



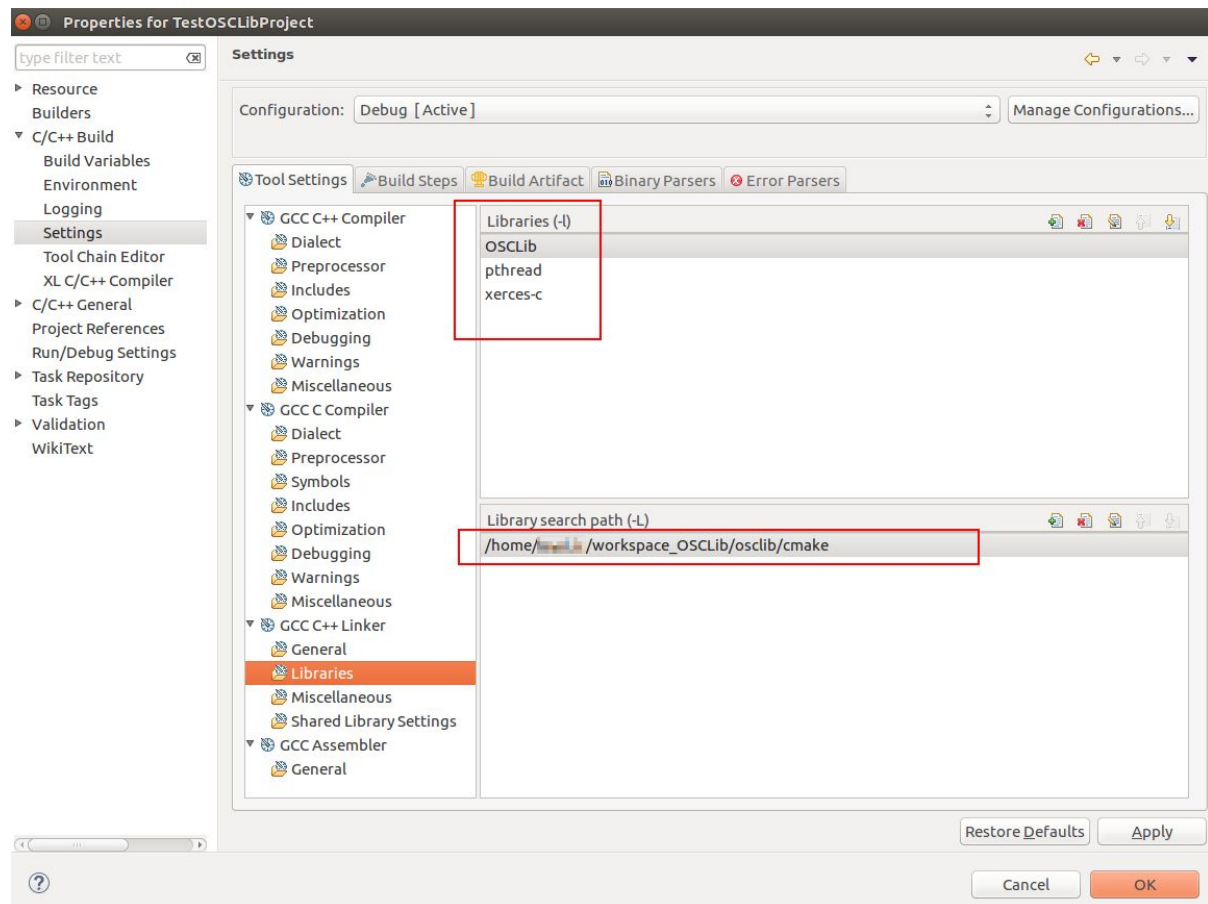
Select the “Preprocessor” node and enter the defined symbol “_LINUX”:



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”. Then enter the path to the shared object in the “Library search path” section:



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:

