Using Intel® C++ Compiler in Eclipse* for Embedded Linux* targets

Contents

Introduction

In embedded area, Eclipse* is popular and widely used. GCC* is well supported in Eclipse* now, while using Intel® C/C++ compiler (ICC), you can also easily use it with Eclipse*. This article will guide you to integrate Intel® C/C++ compiler (component of Intel® System Studio 2015) with Eclipse* and starts to build a hello-world project.

This articles applies for users who developer on a **Windows* or Linux* host**, and use an embedded Linux targets, including Yocto Project*, Wind River* Linux*, CE Linux* (Note: Only Wind River* Linux* provides official SDK (Toolchains) for Windows* host.)

*How to integrate Intel® C++ compiler with Eclipse**

The integration is based on CDT (Eclipse* C/C++ Development Tools) plugins, and CDT is needed before installing the ICC integration plugins. The prerequisites for successful Eclipse* integration is:

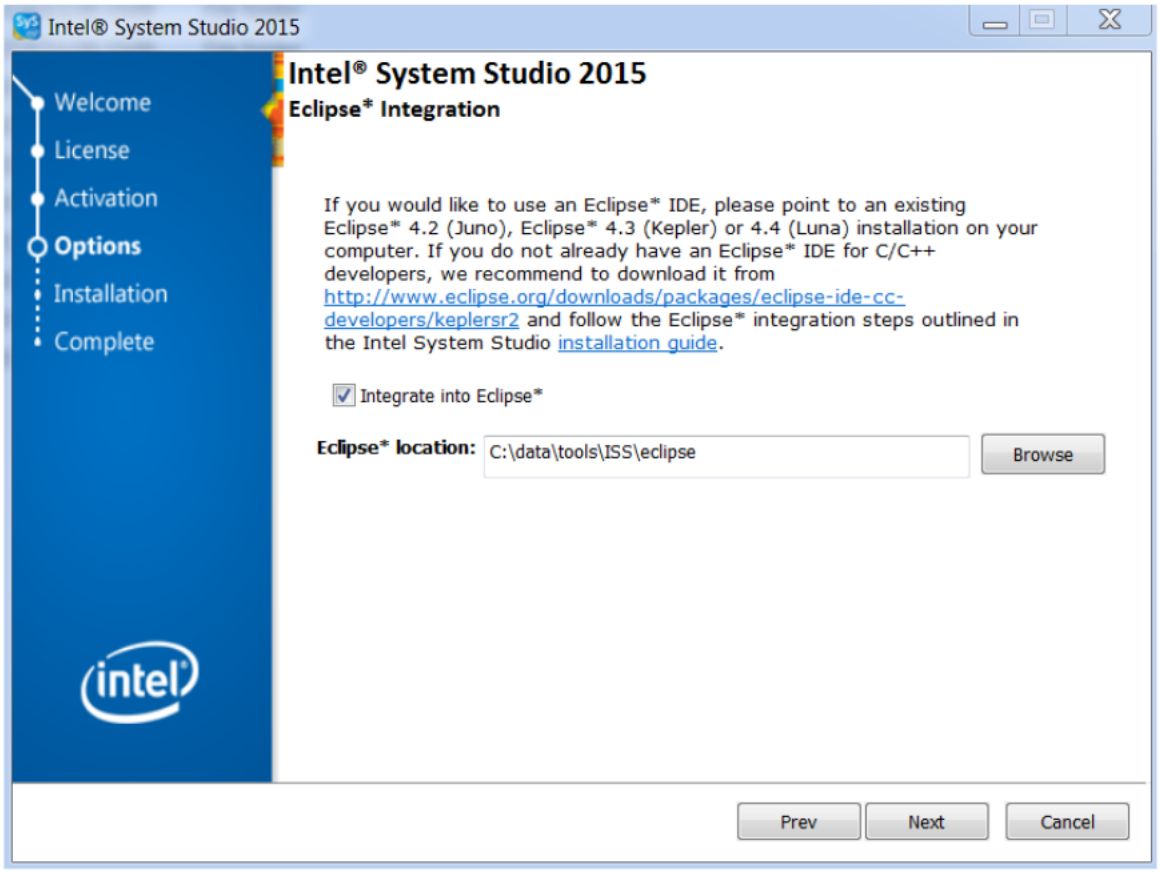
(1) Eclipse* 3.7 (Indigo) or above

(2) Eclipse* CDT 8.0

(3) Java Runtime Environment (JRE) version 6.0 (also called 1.6) update 11 or later.

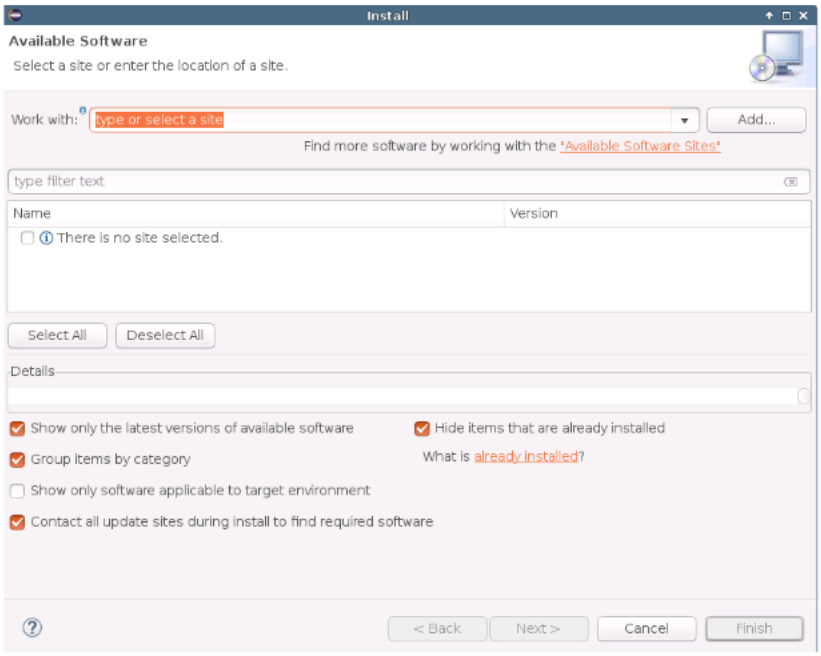
Automatic Integration during Intel System Studio installation

Pay attention to below screenshot when installing, check the “Integrate into Eclipse*”.



Manual Integration in Eclipse*

This is same with installing any other Eclipse* plugins, click the menu item "Help->Install New Software..." and install from local directory which contains the plugins. See below the pop-up window after clicking “Help -> Install New Software...”:



In the “type or select a site” input box, type with below format:

<Name (any string)> - file:<path of the CDT plugin of Intel System Studio>

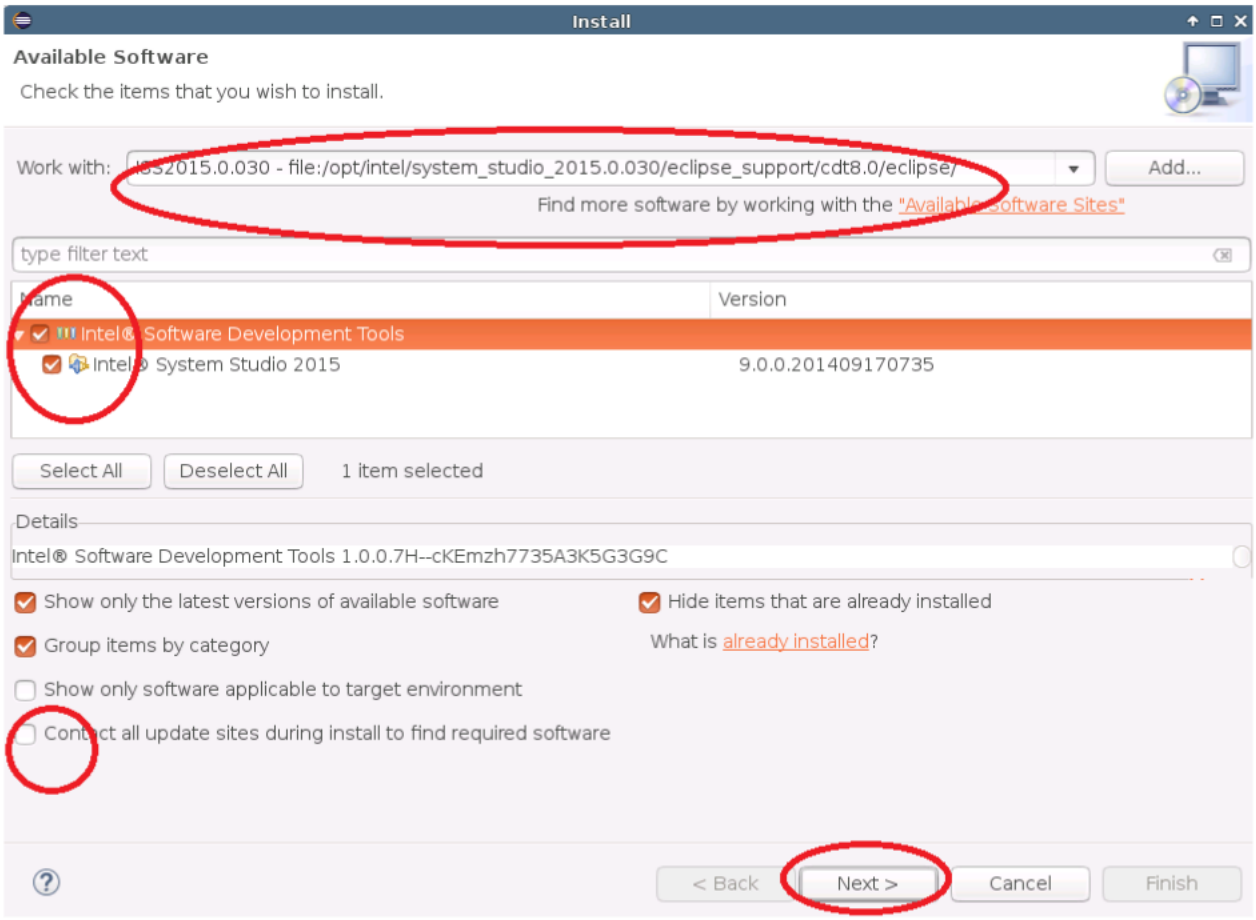
For example:

ISS2015.0.030 - file:/opt/intel/system_studio_2015.0.030/eclipse_support/cdt8.0/eclipse/

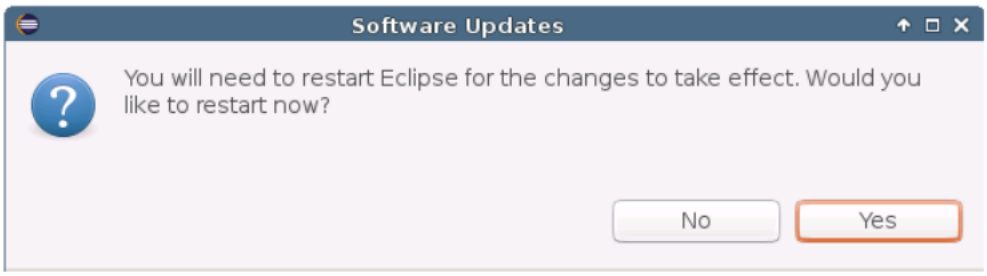
OR:

ISS2015.0.030 - file:/C:/Program Files (x86)/Intel/System Studio 2015.0.027/eclipse_support/cdt8.0/eclipse/

After typing, **press enter** and you will see the contents of this plugin, check the checkbox of “Intel® Software Development Tools”, and uncheck “contact all update sites during install to find required software” (without this, it requires internet connection, or it may hang with the message “Calculating requirements and dependencies.”), and see below:



Now, you can click the “Next” button and follow the prompt to accept the license agreement and click “Finish” button to start the installation. After the installation finished, you will see pop-up window as below:

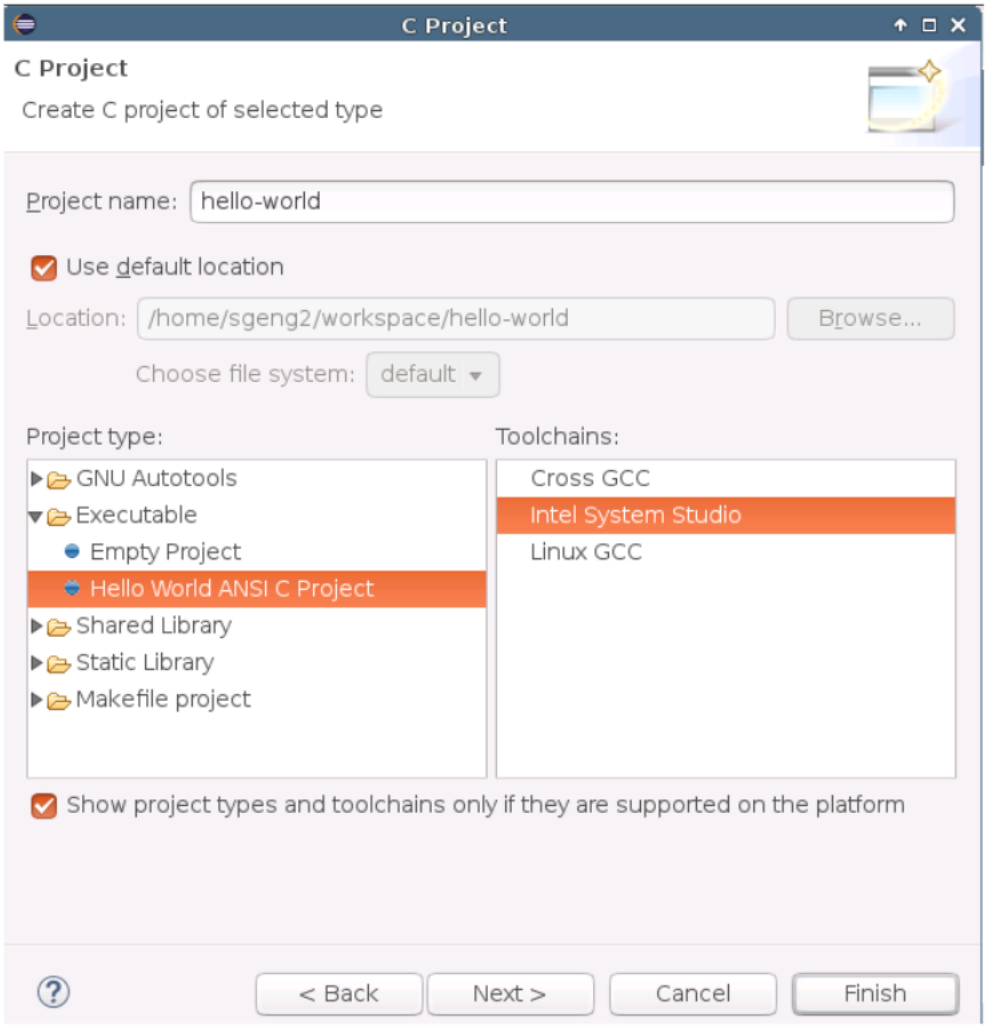


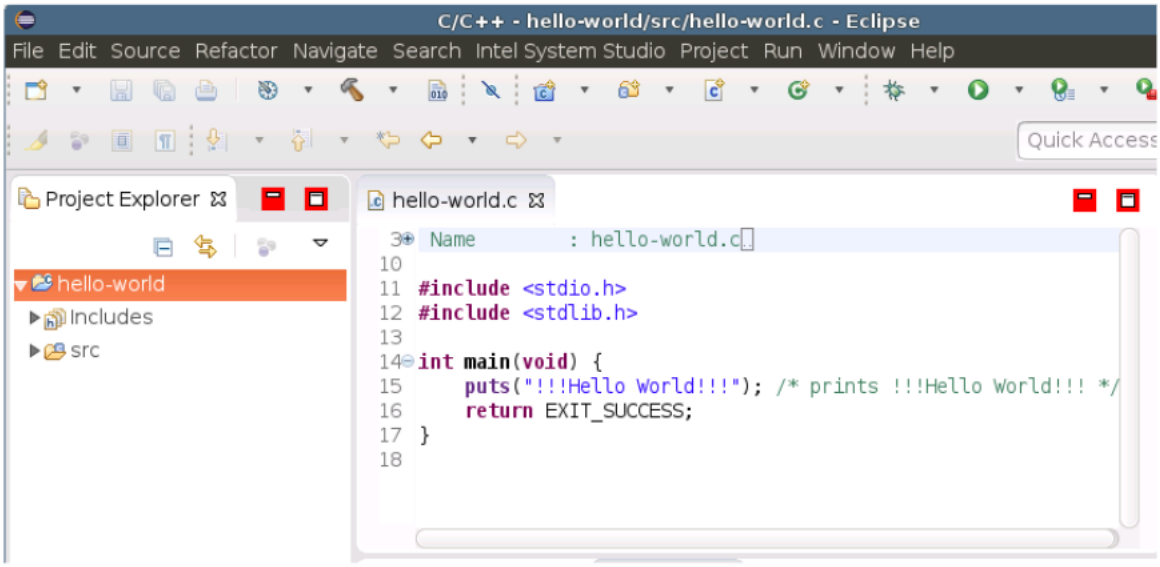
Click “Yes” to restart Eclipse and the integration is done.

*Create a project with Intel® compiler in Eclipse**

You can easily create a project and use Intel® tool chain to build. You only need to select corresponding Intel® Toolchain while creating the project. First, create a project with menu item “File->New->Project”, in the "New Project" window, select "C/C++ -> C Project" and click "Next", then, you

can select “Intel System Studio” as the Toolchains for your project (hello-world as “Project Name” and “Hello World ANSIC C Project” as the “Project type”), click “Finish” and the project is created.





Setting up the cross-build options for your embedded Linux Targets*

By default, the created project will build your application for your host machine, we need to set up the cross-build options for the embedded Linux* targets (Yocto Project*, Wind River* Linux*, CE Linux*).

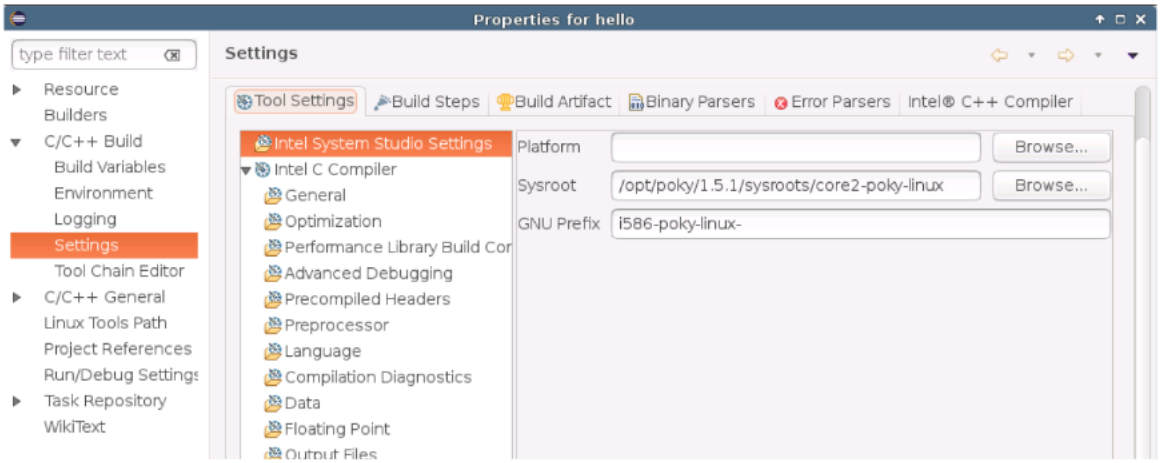
You are suggested to read below article first to understand how to use Intel Compiler for cross compilation:

https://software.intel.com/en-us/articles/using-intel-c-compiler-for-embedded-system

Using gcc compatible sysroot option

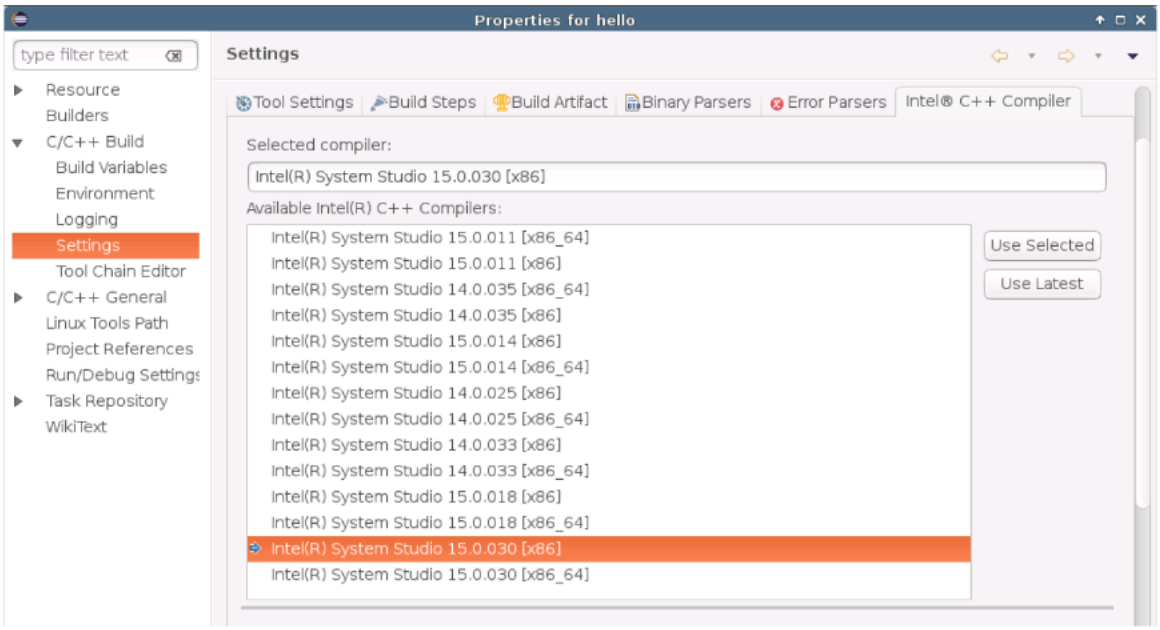
This is introduced in Intel C++ Compiler 14.0 from Intel System Studio 2014, and suggested to be used for newer embedded Linux* versions, such as Yocto Project* 1.3 or above, Wind River* Linux* 4 or above and CE Linux* PR32 or above.

Basically, you need to set --sysroot and --gnu-prefix options for Intel Compiler. Click "Project -> Properties" menu item and setting it in pop-up window as below (take Yocto Project* 1.5.1 as an example):



It is similar for other embedded Linux, for example, for Wind River* Linux 5.x, the GNU Prefix value may be "i686-wrs-linux-gnu-".

By default, the created project will use 64bit target compiler, if you are using 32bit target as above example, you need to switch the compiler in "Intel(R) C++ Compiler" tab in above screenshots, see below:



You can select the compiler marked as “x86” and click “Use Selected” button for 32bit targets.

Using platform configuration

This is introduced from Intel C++ Compiler 13.0 from Intel System Studio 2013, and still available in Intel C++ Compiler 15.0 from Intel System Studio 2015, to support the old embedded Linux* versions, such as Yocto Project* 1.2 or 1.3, CE Linux* PR28, Wind River* Linux* 5.

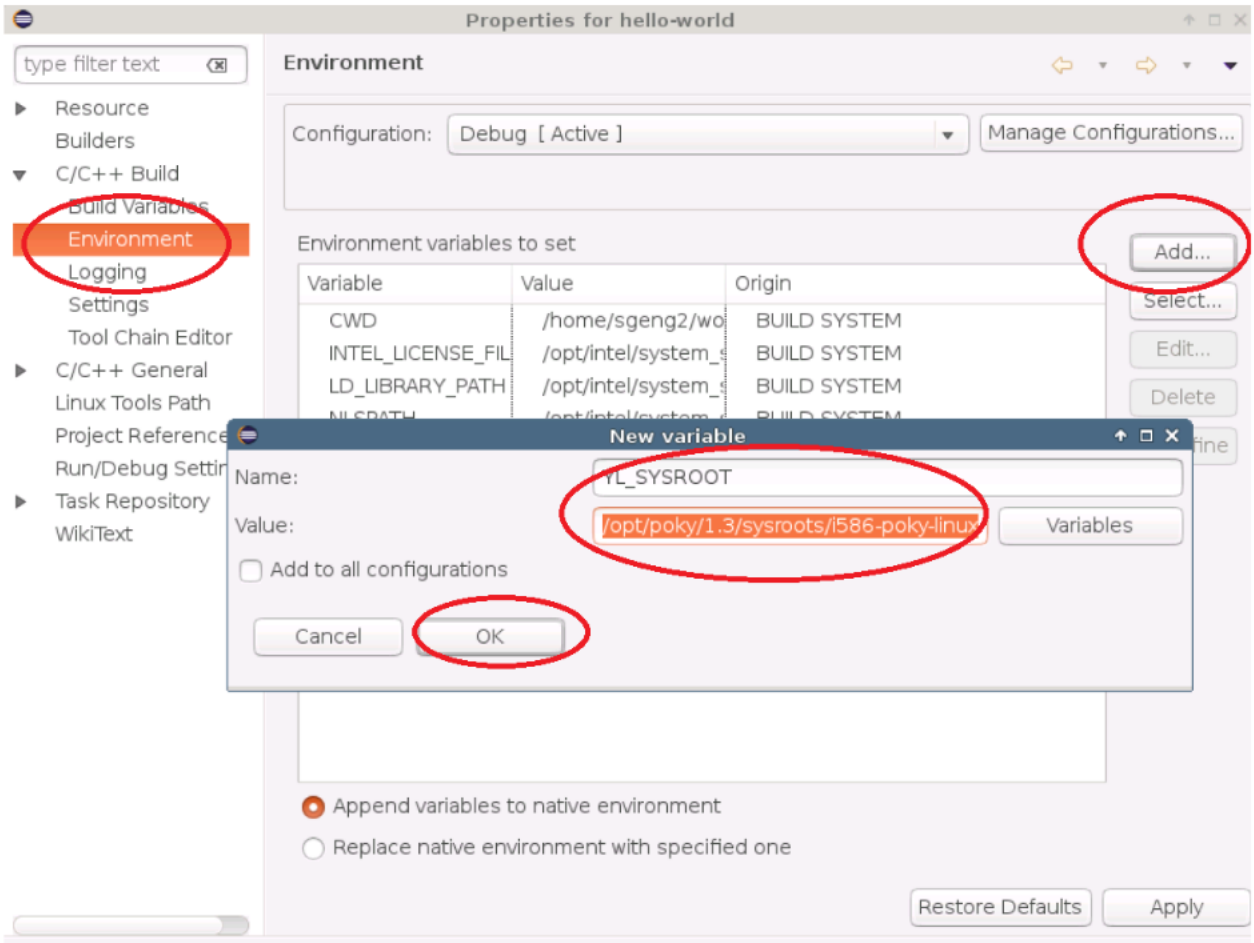
Note: it is suggested to use the sysroot and GNU prefix options if the target Linux* is supported by them, such as Yocto Project* 1.3, Wind River* Linux* 5.

First of all, you need to set the environment variables based on your target Linux*.

For Yocto Project* based target, an example as below:

```
YL_SYSROOT=/opt/poky/1.3/sysroots/i586-poky-linux
YL_TOOLCHAIN=/opt/poky/1.3/sysroots/i686-pokysdk-linux/usr/bin/
```

To add these environment variables in Eclipse*, click “Project -> Properties” menu item and click “Add…” button in pop-up window as below:

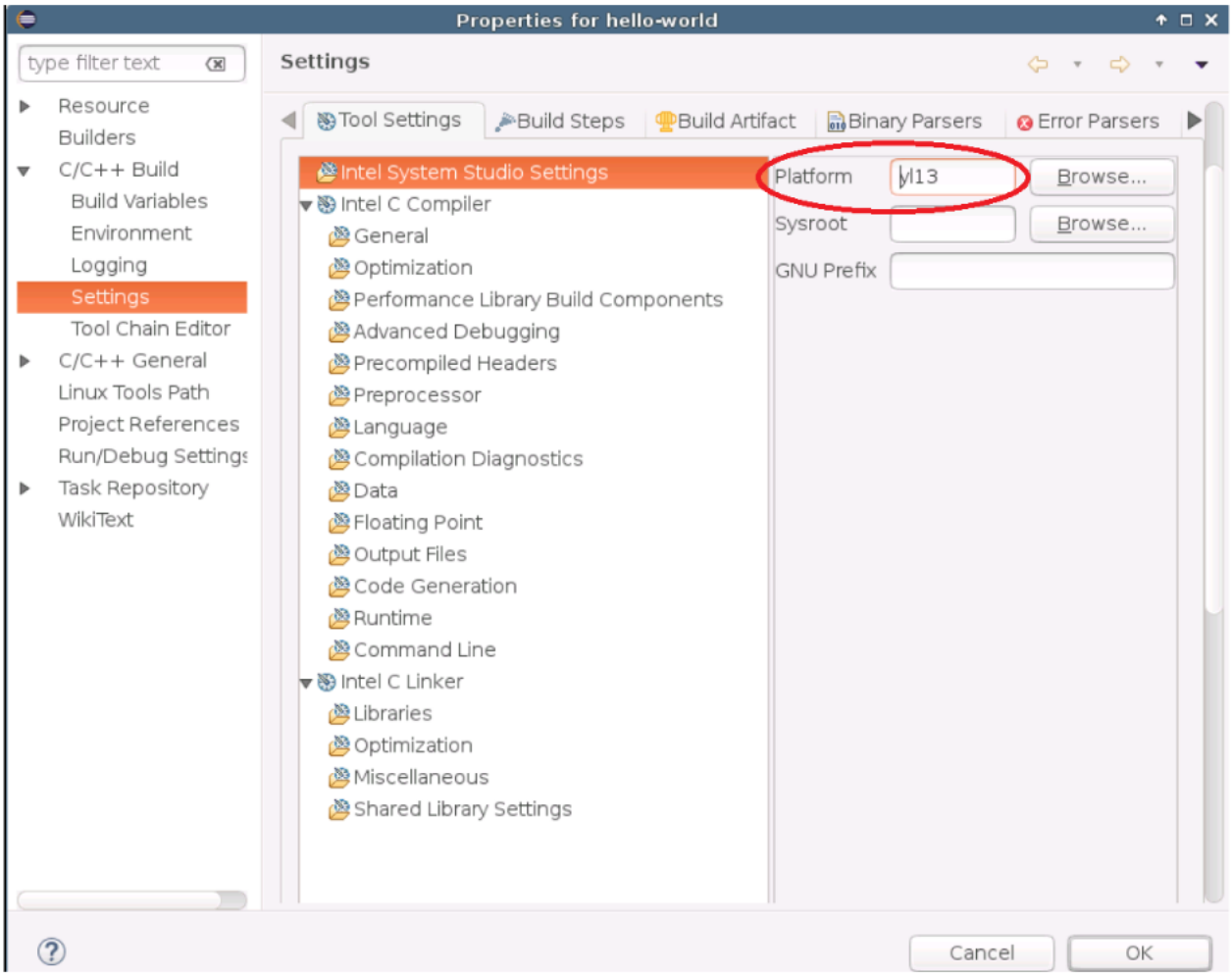


After adding the 2 environment variables, you will see below:

Environment variables to set

Variable	Value
CWD	/home/sgeng2/workspace/hello-world/Debug
INTEL_LICENSE_FILE	/opt/intel/system_studio_2015.0.030/licenses:/home/s
LD_LIBRARY_PATH	/opt/intel/system_studio_2015.0.030/compiler/lib/intel6
NLSPATH	/opt/intel/system_studio_2015.0.030/compiler/lib/intel6
PATH	/opt/intel/system_studio_2015.0.030/bin/intel64:/opt/p
PWD	/home/sgeng2/workspace/hello-world/Debug
YL_SYSROOT	/opt/poky/1.3/sysroots/i586-poky-linux/
YL_TOOLCHAIN	/opt/poky/1.3/sysroots/i686-pokysdk-linux/usr/bin/

Next, you need to set the -platform option based on your target Linux*, for example, for Yocto Project* 1.3, you will need to set it as “yl13”, see below:

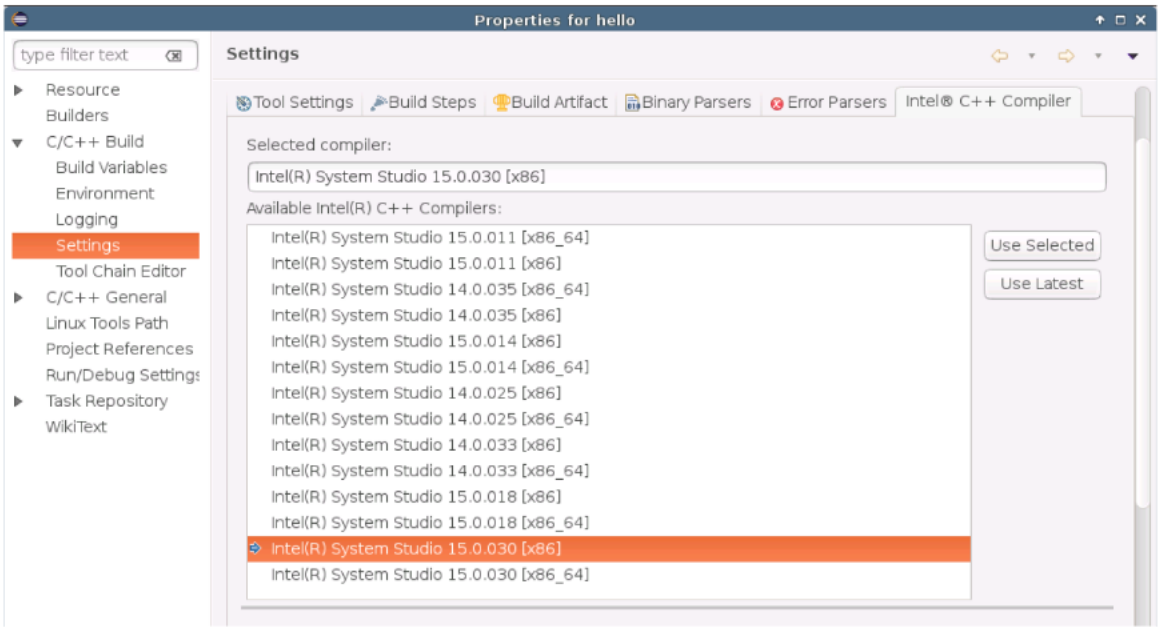


The supported values for -platform are as below:

32bit Targets: celpr28, yl12, yl13, wrl43, wrl50

64bit Targets: wrl43, wrl50

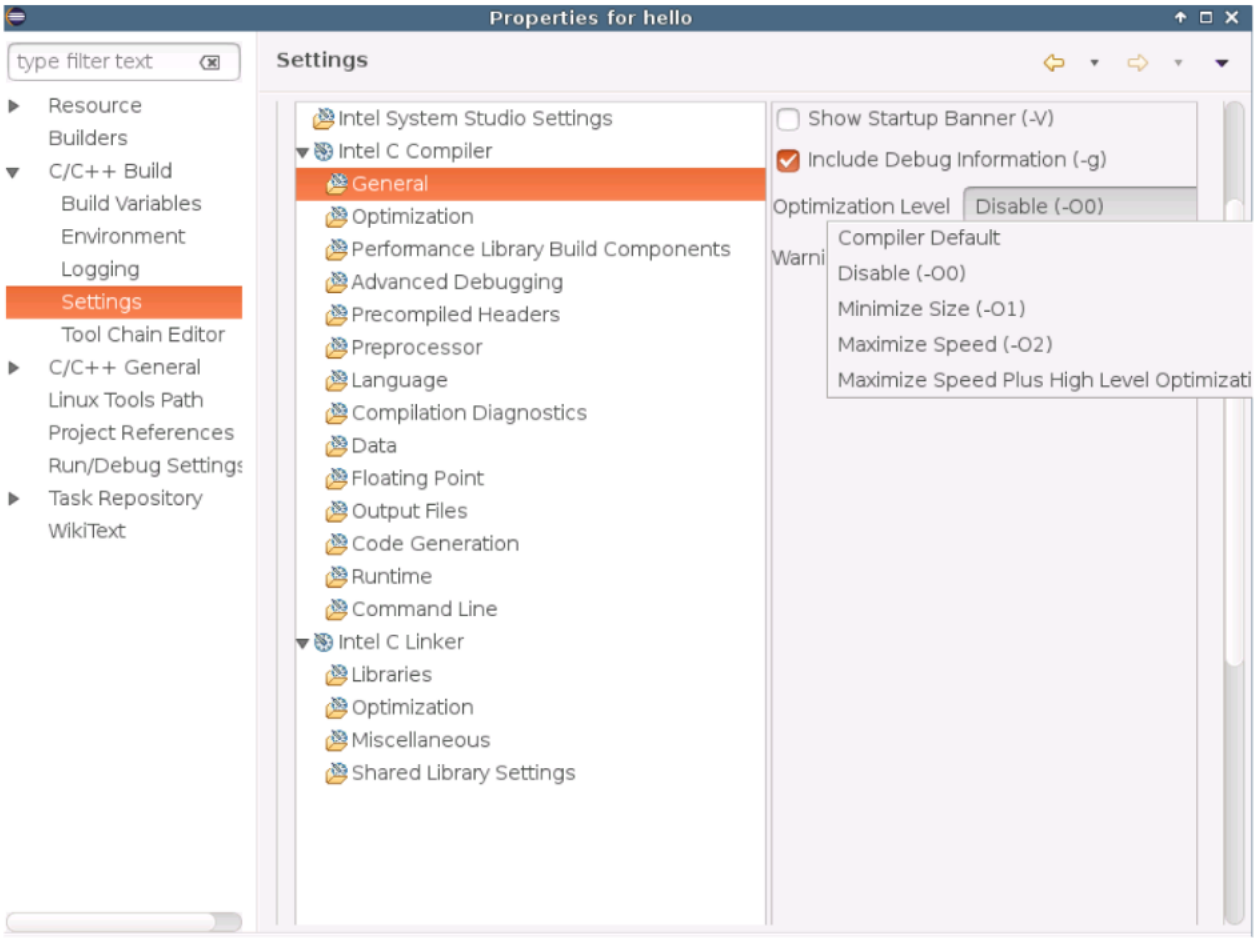
Again, the default created project will use 64bit as target, for 32bit targets, you need to switch it as below:



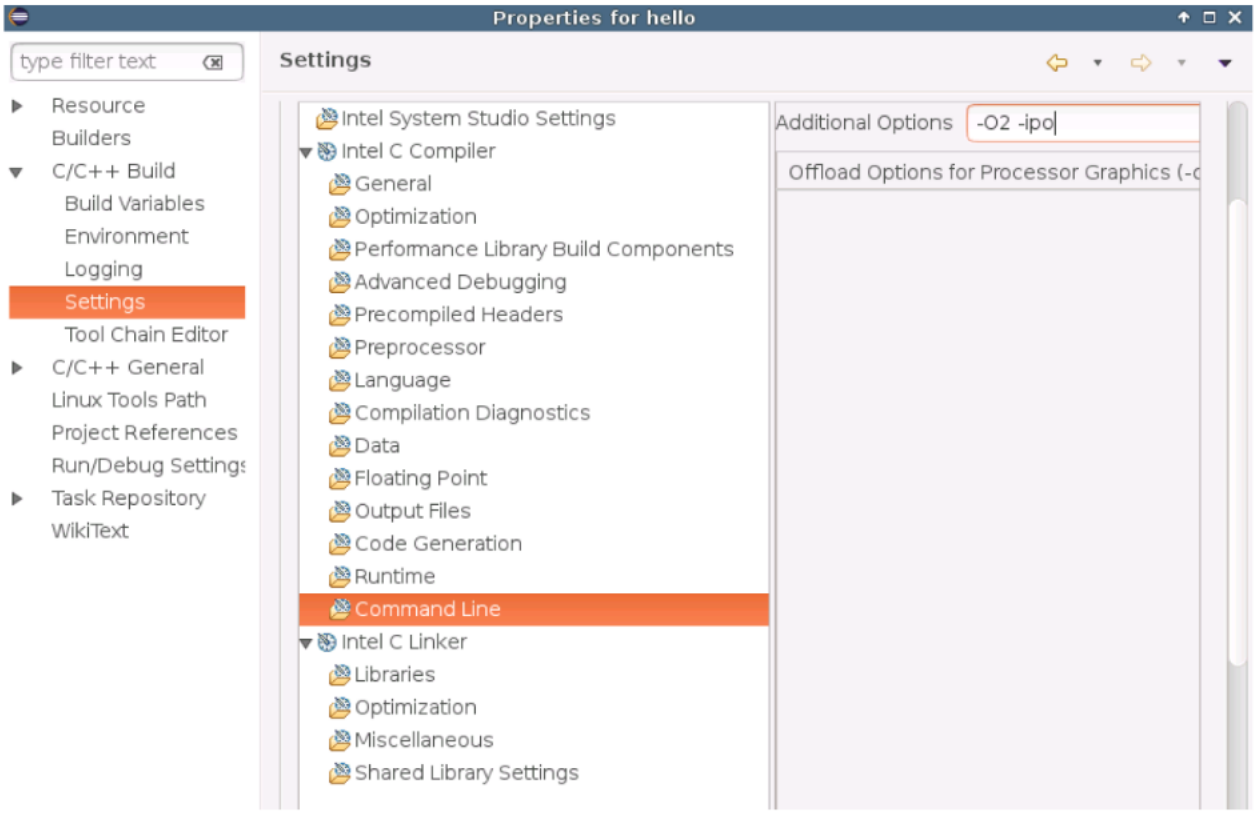
You can select the compiler marked as “x86” and click “Use Selected” button for 32bit targets.

*Using Intel® compiler options in Eclipse**

ICC is a compiler helping to improve performance and it provides lots of optimization options. You may need to set the compilation options in Eclipse. You can do this from "Project->Properties" menu items. See below example about setting the optimization level (O1/2/3):



The “Intel C Compiler” contains the settings for compiler options, and the “Intel C Linker” contains the settings for linker options. All these settings of Intel Compiler and Linker options are grouped. You can easily find the options you want. If you cannot find specific option from these groups, you can also add it using the "Command Line" group, which contains an edit box to enter the additional options you need, see below:



Build the project

Now, you can build the project and the build log as below: