



AN 834: Using the Intel[®] HLS Compiler Pro Edition with an IDE

Updated for Intel[®] Quartus[®] Prime Design Suite: **20.1**



AN-834 | 2020.05.29

Latest document on the web: [PDF](#) | [HTML](#)



Contents

1. Using the Intel® HLS Compiler Pro Edition with an IDE.....	3
2. Using Intel HLS Compiler Pro Edition with an Eclipse* IDE on Linux Operating Systems.....	4
2.1. Debugging i++ Executable Outputs in the Eclipse IDE	8
3. Using Intel HLS Compiler Pro Edition with Microsoft* Visual Studio* on Windows Operating Systems.....	10
3.1. Debugging i++ Executable Outputs in Microsoft Visual Studio.....	13
4. Document Revision History.....	14

1. Using the Intel[®] HLS Compiler Pro Edition with an IDE

Integrating the Intel[®] HLS Compiler with an IDE lets you use a native compiler and debugger to work on the functional correctness of your testbench and HLS component before optimizing your component for FPGA performance.

This document covers using the Intel HLS Compiler Pro Edition Version 20.1 with an IDE. For instructions on using the Intel HLS Compiler Standard Edition with an IDE, see [AN 918: Using the Intel HLS Compiler Standard Edition with an IDE](#).

Only the Intel HLS Compiler `i++` command generates the High-Level Design Reports that help you to optimize your component.

On Linux operating systems, the native compiler is `g++`, while on Windows operating systems the native compiler is Microsoft Visual C++ (MSVC).

At a high level, using a native compiler (`g++` or MSVC) involves the following steps:

1. Export the path to the Intel HLS Compiler include files to your environment:

```
<hls_installdir>/include
```

Where `<hls_installdir>` is the path to your Intel HLS Compiler. The default installation locations for the Intel HLS Compiler are as follows:

- Linux: `/home/<username>/intelFPGA_pro/20.1/hls`
- Windows: `C:\intelFPGA_pro\20.1\hls`

2. Export the path to the Intel HLS Compiler emulation library to the same environment:

- Linux:

```
<hls_installdir>/host/linux64/lib
```

- Windows:

```
<hls_installdir>/host/windows64/lib
```

3. In the environment set with the exported paths, run your IDE (or compiler) and compile your code with the emulation library and any other flags described in the ["Compiler Interoperability" in Intel HLS Compiler Pro Edition Version 20.1 Reference Manual](#).

You can also use your IDE to debug executables generated by the Intel HLS Compiler compiler, including your simulation testbench.

2. Using Intel HLS Compiler Pro Edition with an Eclipse* IDE on Linux Operating Systems

Before you use the Intel HLS Compiler Pro Edition Version 20.1 with an Eclipse* IDE, ensure that you have complete the following tasks:

- Install the Intel HLS Compiler Pro Edition.
For information about installing the Intel HLS Compiler, see [Intel High Level Synthesis Compiler Pro Edition Getting Started Guide](#).
- Install an Eclipse IDE and the Eclipse C/C++ Development Tooling (CDT)

To use the Intel HLS Compiler Pro Edition with an Eclipse IDE, start a terminal session and complete the following steps in that terminal session:

1. Run the following command:

```
source <hls_installdir>/init_hls.sh
```

Where <hls_installdir> is the path to your Intel HLS Compiler. The default installation location for Intel HLS Compiler Pro Edition is /home/<username>/intelFPGA_pro/20.1/hls.

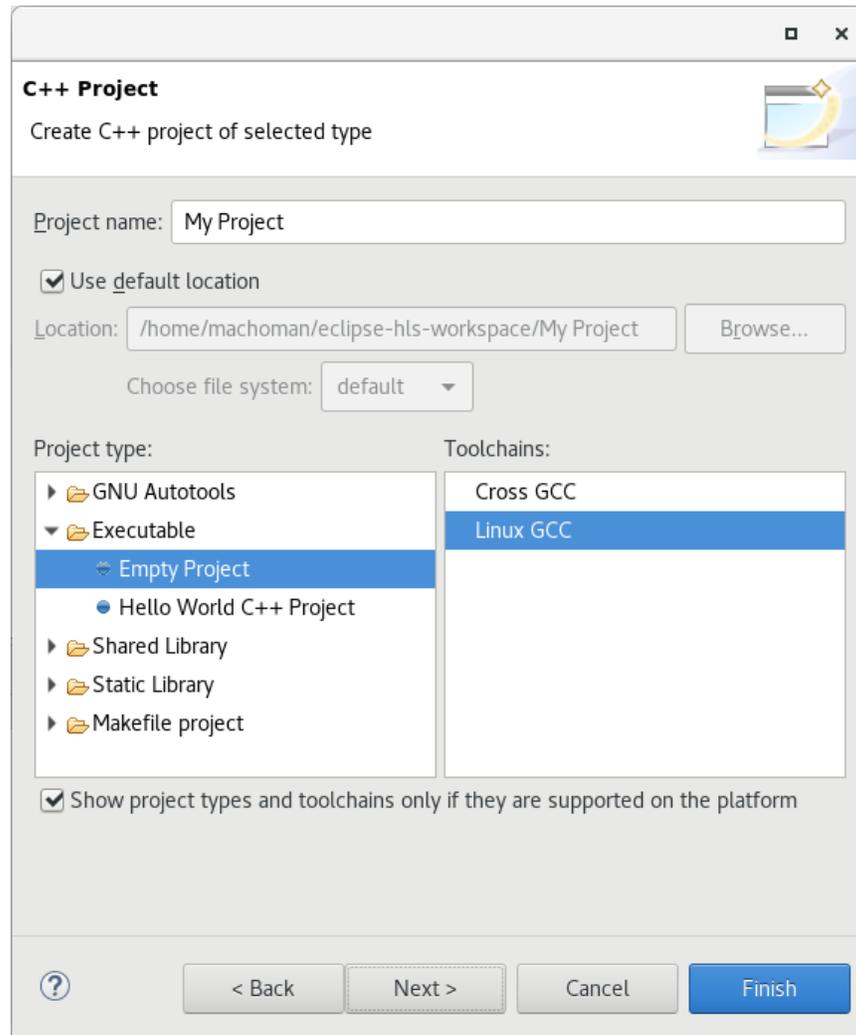
2. Set the CPATH and LIBRARY_PATH environment variables as follows:

```
export "CPLUS_INCLUDE_PATH=<hls_installdir>/include"  
export "LIBRARY_PATH=<hls_installdir>/host/linux64/lib"
```

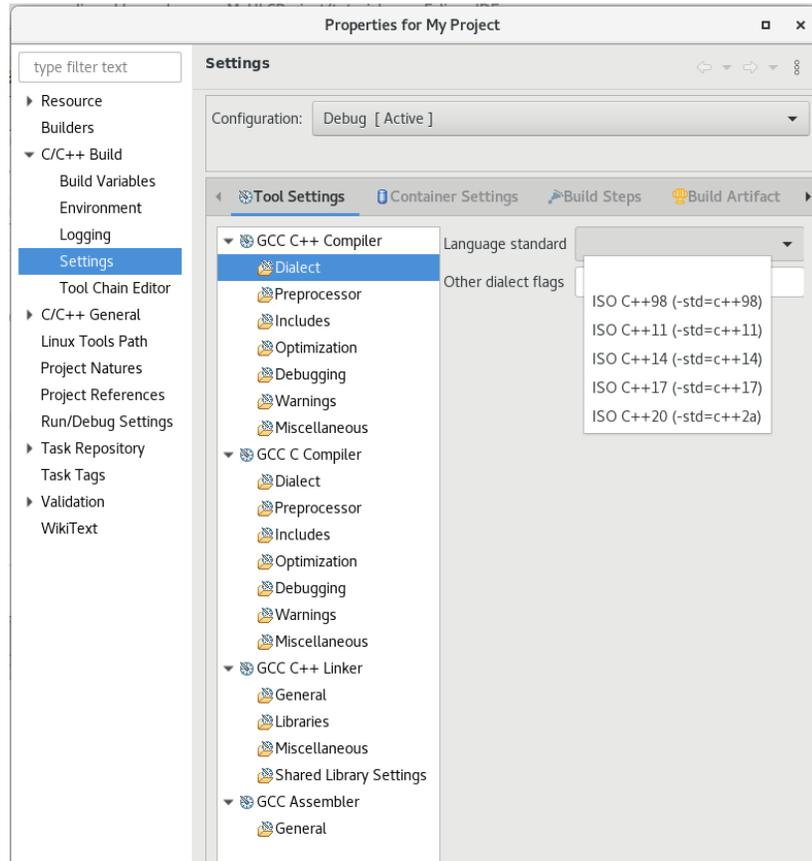
3. In the same terminal session, start your Eclipse IDE.

If you have an HLS project that you have compiled before, you are ready to run and debug your HLS component, and you can skip the next step.

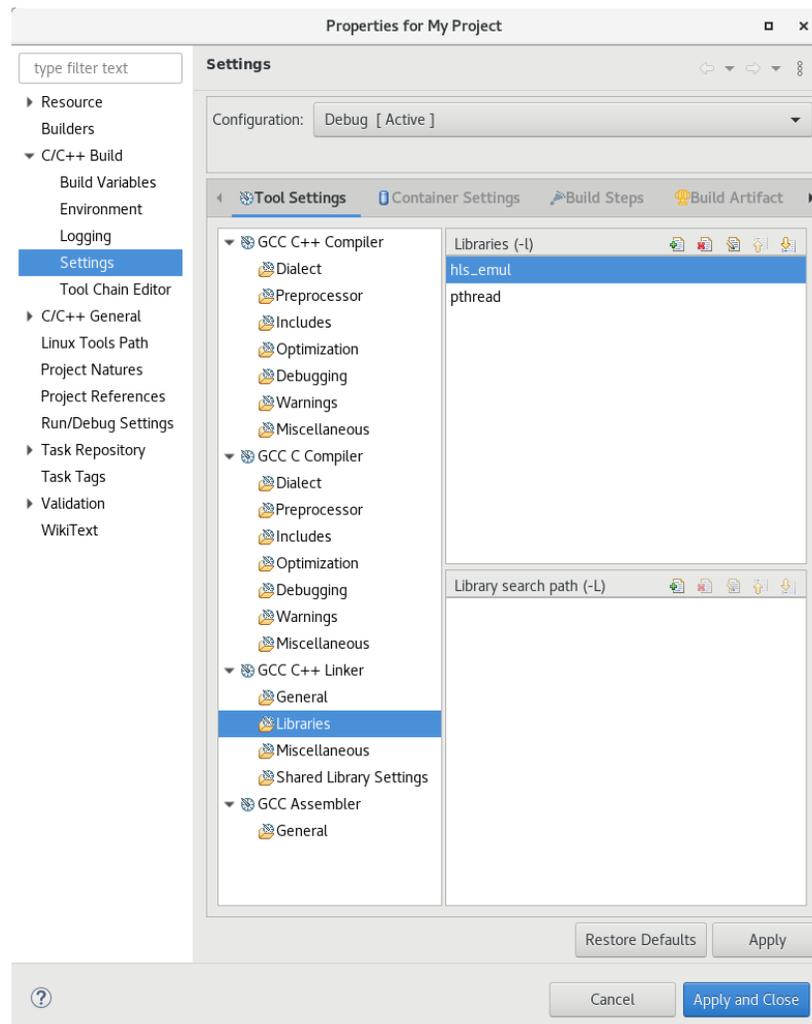
4. Create and configure an Eclipse project for your HLS component:
 - a. Create a new C++ project in Eclipse (**File > New > C++ Project**).
 - b. Select a **C++ Managed Build** template for the project and select the following properties:
 - **Project type: Executable > Empty Project**
 - **Toolchains: Linux GCC**



- c. Click **Next** and click **Advanced Settings**.
You can also access these settings later through your project properties:
 - i. In the **Project Explorer** view, right-click your project and select **Properties**.
 - ii. In the **Properties** window for your project, go to **C/C++ Build > Settings**.
- d. Set the C++ dialect and libraries call properties for the project:
 - i. On the **Tool Settings** tab, go to **GCC C++ Compiler > Dialect** and set the **Language standard** value to ISO C++17 (`-std=c++17`).



- ii. On the **Tool Settings** tab, go to **GCC C++ Linker > Libraries** and add the `hls_emul` and `pthreadlibraries` to the list of libraries.



- e. Click **Apply and Close**.
- f. Click **Finish** to finish creating your project.
- g. Import or create your HLS source files.
- h. If there are problems with the header files, rebuild the index.

To rebuild the index, right-click your project in the **Project Explorer** view and select **Index > Rebuild**.

You can now develop, run, and debug your HLS component using your Eclipse IDE.

After you have confirmed the functional correctness of your component, you can start to optimize your HLS component by using the information in the Intel HLS Compiler High-Level Design Reports. These reports are generated when you compile your component with the `i++` command.

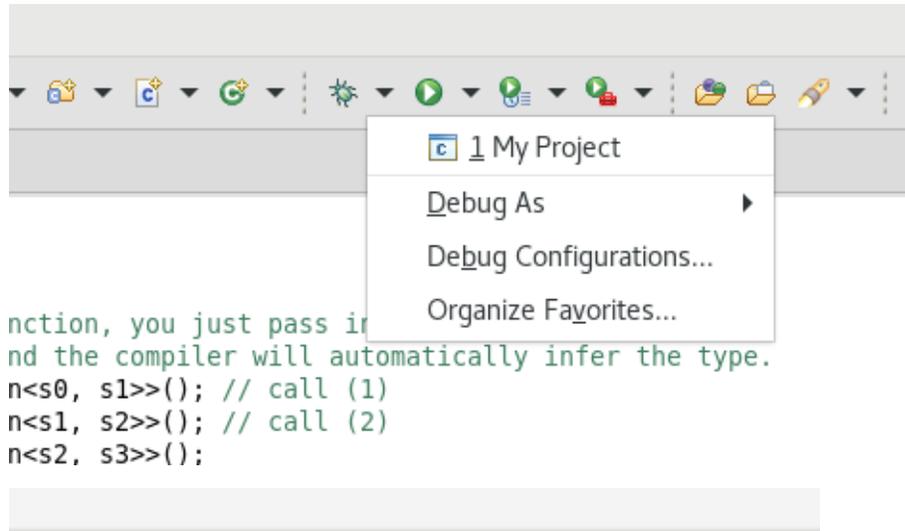
You can also use your Eclipse IDE to debug executable files that are produced by compiling your component with the `i++` command

2.1. Debugging i++ Executable Outputs in the Eclipse IDE

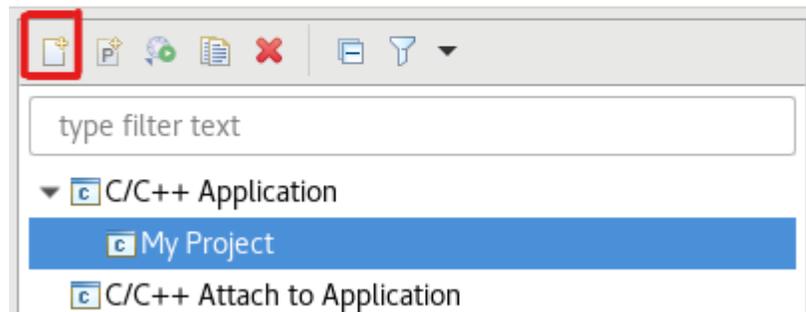
You can use your Eclipse IDE to debug executable files that are produced by compiling your component with the `i++` command by creating and using a debug configuration for the executable.

To debug an executable file produced by the `i++` command:

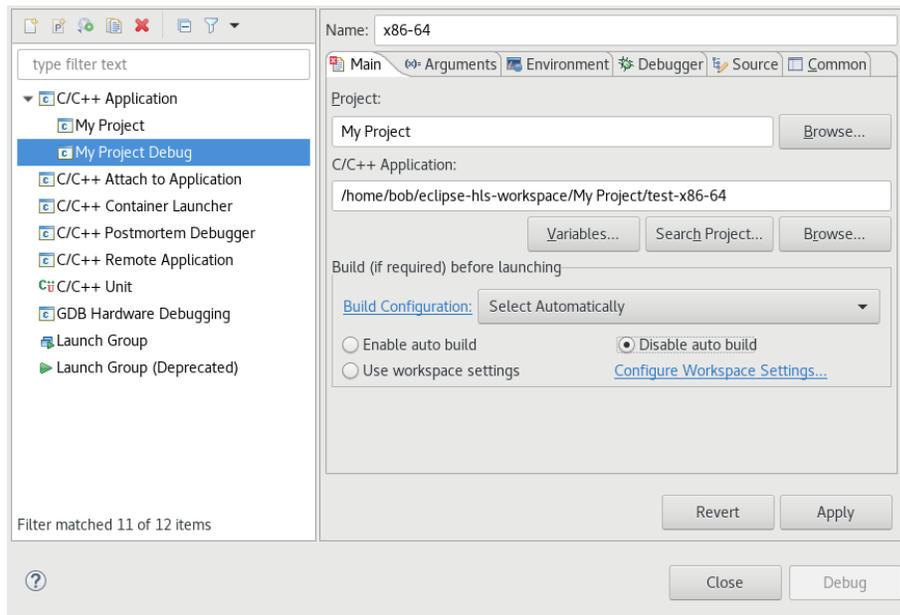
1. Open **Debug Configurations** and create a new debug configuration for the executable:



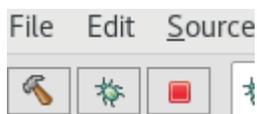
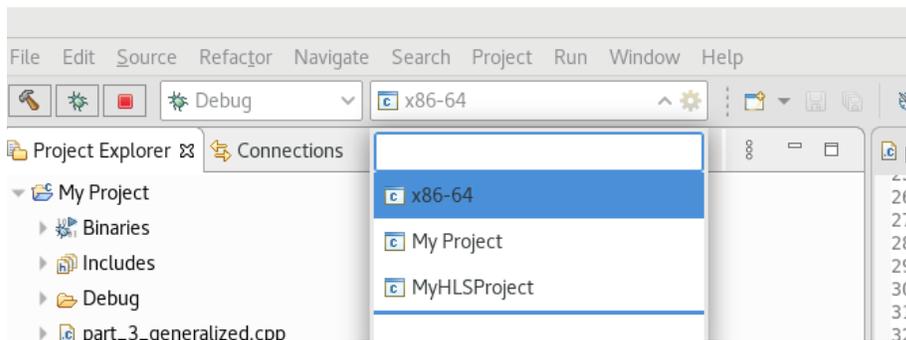
Create, manage, and run configurations



2. Load the executable file produced by the `i++` command and disable auto build:



3. Debug your executable by selecting your new debug configuration and clicking the debug button:





3. Using Intel HLS Compiler Pro Edition with Microsoft* Visual Studio* on Windows Operating Systems

Before you use the Intel HLS Compiler Pro Edition Version 20.1 with Microsoft* Visual Studio*, ensure that you have complete the following tasks:

- Install the Intel HLS Compiler Pro Edition.
For information about installing the Intel HLS Compiler, see [Intel High Level Synthesis Compiler Pro Edition Getting Started Guide](#).
- Install a version of Microsoft Visual Studio* that is supported by the Intel HLS Compiler Pro Edition.
For supported versions of Microsoft Visual Studio*, see "Intel High Level Synthesis Compiler Pro Edition Prerequisites" in [Intel High Level Synthesis Compiler Pro Edition Getting Started Guide](#).

To use the Intel HLS Compiler Pro Edition with a Visual Studio IDE:

1. Start a Windows Command Prompt session.
2. In your command prompt session, run the following command:

```
<hls_installdir>\init_hls.bat
```

Where *<hls_installdir>* is the path to your Intel HLS Compiler installation. For example, `C:\intelFPGA_pro\20.1\hls`.

3. Start your Visual Studio IDE:

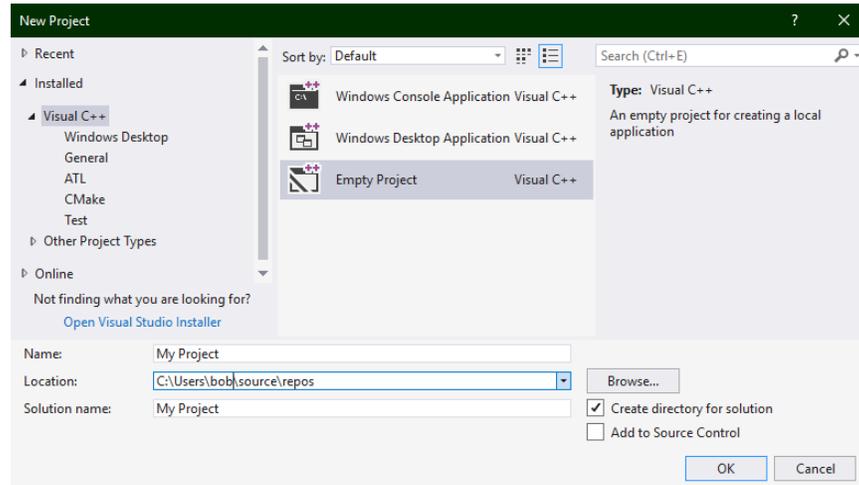
```
devenv /useenv
```

If you have an HLS project that you have compiled before, you are ready to run and debug your HLS component, and you can skip the next step.

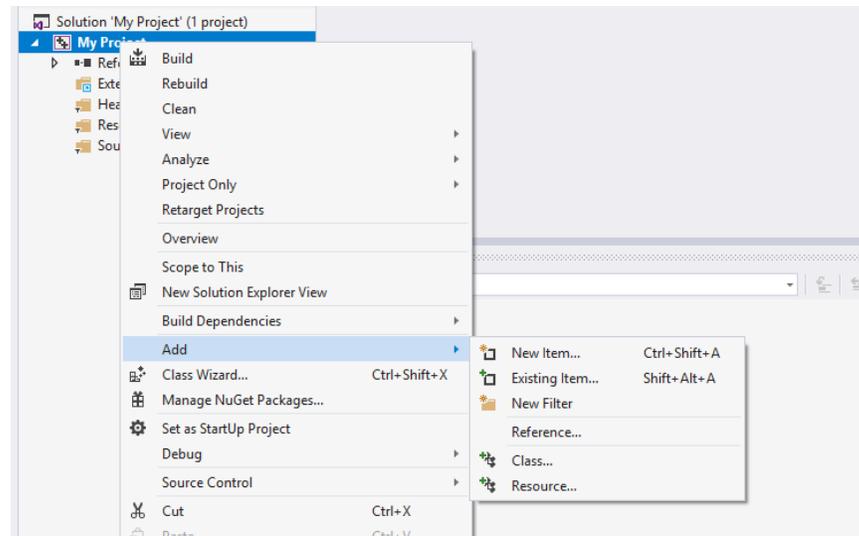
4. Create and configure a Visual Studio project for your HLS component:
 - a. Create a new project:

3. Using Intel HLS Compiler Pro Edition with Microsoft* Visual Studio* on Windows Operating Systems

AN-834 | 2020.05.29



b. Add your sources:



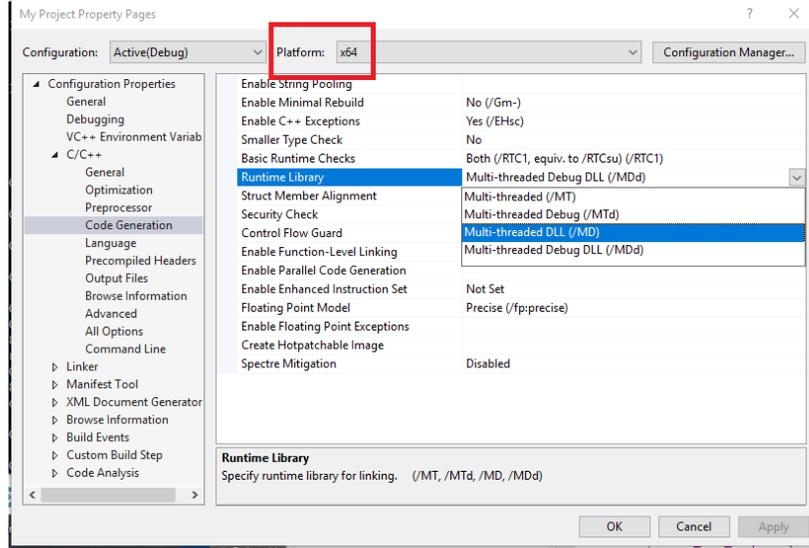
c. Disable debug symbols for the C++ runtime library:

- i. Right-click your project and click **Properties**.
- ii. In the **Property Pages** dialog box, go to **Configuration Properties** > **C/C++** > **Code Generation** and change the **Runtime Library** to **Multi-threaded DLL (/MD)**:
Ensure that you have **x64** selected in the **Platform** field.



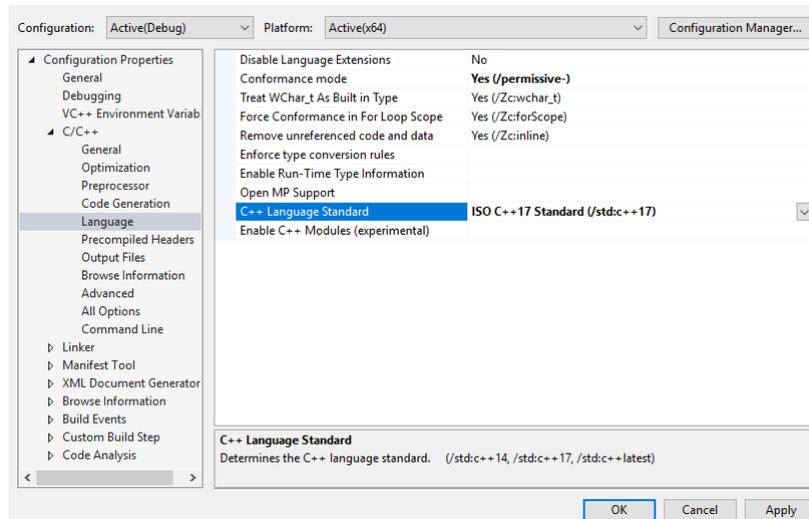
3. Using Intel HLS Compiler Pro Edition with Microsoft* Visual Studio* on Windows Operating Systems

AN-834 | 2020.05.29



- iii. In the **Property Pages** dialog box, go to **Configuration Properties** > **C/C++** > **Language** and change the **c++ Language Standard** to **ISO C++17 Standard (/std:c++17)**:

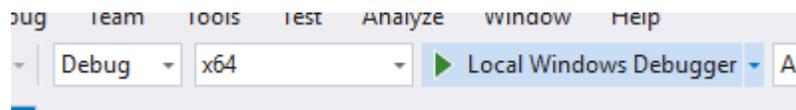
Ensure that you have **x64** selected in the **Platform** field.



- iv. Click **OK**.

You can now develop, run, and debug your HLS component using your Visual Studio IDE.

To run your component in Visual Studio, choose the 64-bit target:



After you have confirmed the functional correctness of your component, you can start to optimize your HLS component by using the information in the Intel HLS Compiler High-Level Design Reports. These reports are generated when you compile your component with the `i++` command.

3.1. Debugging i++ Executable Outputs in Microsoft Visual Studio

You can also use your Visual Studio IDE to debug output from the `i++` command.

For details, see [Debug an app that isn't part of a Visual Studio solution](#) in the Microsoft Visual Studio documentation. When you follow the Microsoft instructions, ensure that you debug the `.exe` file generated by the `i++` command.

4. Document Revision History

Table 1. Document Revision History for AN 834: Using the Intel HLS Compiler with an IDE

Date	Version	Changes
May 2020	2020.05.29	<ul style="list-style-type: none"> Changes in the Intel HLS Compiler since this document was last published have changed the process for using an IDE with the Intel HLS Compiler significantly. As such, this document has been largely rewritten. Changed the title from <i>AN 834: Developing for the Intel HLS Compiler with an IDE</i> to <i>AN 834: Using the Intel HLS Compiler with an IDE</i>. Removed information about running the <code>i++</code> command from within an Eclipse* IDE. That information is outside of the scope of this document Removed Standard Edition information from this document. For information about using Intel HLS Compiler Standard Edition with an IDE, see AN 918: Using the Intel HLS Compiler Standard Edition with an IDE.
July 2019	2019.09.19	<ul style="list-style-type: none"> Revised references to supported versions of Microsoft Visual Studio Updated example paths to a more recent version Fixed broken links to other Intel HLS Compiler documentation
December 2017	2017.12.01	<ul style="list-style-type: none"> Initial release