Intel® System Studio Product Guide

May 30, 2019
Contents

Introduction ............................................................................................................................................................... 4

Core Components and Extensions................................................................. 4
Complementary Toolkits and SDKs......................................................... 4

Get Started ............................................................................................................. 6

Install Intel System Studio ................................................................. 6
Create a Simple Project .............................................................................. 6
Getting Started Page (Linux Target) ............................................................ 7
Getting Started Guide (Windows Target) .................................................... 9

Integrated Development Environment (IDE) .............................................. 10

Toolbar ........................................................................................................ 10
Enhanced Perspectives .................................................................................. 10
Sensor Support View .................................................................................... 11
Cloud Support View ..................................................................................... 11
MKL Browser View ....................................................................................... 12
Switching Perspectives .................................................................................. 12

Use Case: Debug a System Bring Up Problem in the BIOS ........................................... 13

Problem Statement .................................................................................... 13
Solution ....................................................................................................... 13
Instructions ................................................................................................. 13
Next Steps ................................................................................................. 24

Use Case: Develop Computer Vision and Deep Learning Optimized Applications ................. 25

Using the OpenVINO™ Toolkit with Intel® System Studio ................................. 25
Solution ....................................................................................................... 25
Instructions ................................................................................................. 25
Next Steps ................................................................................................. 25

Getting Help and Support ........................................................................ 26

Priority Support .......................................................................................... 26
Forums ......................................................................................................... 26
Documentation ............................................................................................ 26
Legal Information

Intel, the Intel logo, VTune, OpenCL, OpenVINO, Atom, Puma, Core, and Xeon are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others

**Copyright 2015-2019 Intel Corporation**

This software and the related documents are Intel copyrighted materials, and your use of them is governed by the express license under which they were provided to you (License). Unless the License provides otherwise, you may not use, modify, copy, publish, distribute, disclose or transmit this software or the related documents without Intel's prior written permission.

This software and the related documents are provided as is, with no express or implied warranties, other than those that are expressly stated in the License.
For device manufacturers, system integrators, and embedded application developers, the Intel® System Studio 2019 tool suite provides a comprehensive set of software development tools that can be used to create solutions based on Intel® architectures. It includes a unified and easy-to-use Integrated Development Environment (IDE) based on Eclipse®, with extensions to make your experience consistent across all phases of the development cycle.

## Core Components and Extensions

The following diagram shows where Intel System Studio components and extensions come into play across development phases.

**NOTE:** You might not have all of the tools and libraries mentioned in this guide. To download additional Intel System Studio tools and libraries, visit the product page on the [Intel Developer Zone](https://www.intel.com/developer).

The core Intel System Studio components include:

- Debug and trace capabilities to isolate complex defects
- Power and performance analyzers to improve performance and power efficiency
- Compilers and performance libraries to build in better performance

The suite also includes extensions for complementary toolkits and SDKs. For information, see the next section.

## Complementary Toolkits and SDKs

To extend the power of your development environment, you can use complementary Intel® toolkits and SDKs within the Intel® System Studio Eclipse*-based IDE. This gives you access to the Intel System Studio workflows, tools, and libraries.

This section contains examples of how this type of "better together" development can work.
Intel® Distribution of OpenVINO™ Toolkit

You can use the Intel® Distribution of OpenVINO™ toolkit with the Intel® System Studio Eclipse*-based Integrated Development Environment (IDE) to develop optimized Computer Vision and Deep Learning applications.

For example, you can:

- Run and debug your OpenVINO toolkit application on a remote target using Docker* container-based workflows in the Intel System Studio IDE.
- Debug and tune using leading compilers, performance and parallel runtime libraries, and debug and analysis tools available with Intel System Studio.

Get started quickly with instructions to create, run, and debug a Security Barrier Camera or an Image Classification sample application in Intel System Studio.

OpenCL™ Developer Tools Component of Intel® System Studio

Use OpenCL™ software technology for offloading compute functions to Intel® processors and to Intel® Processor Graphics. The technology streamlines creating, building, debugging, and analyzing OpenCL applications, and enables customized kernel code with computer vision and media software tools from Intel.

To get started, click here.

Arduino Create*: From Prototype to Production

Use advanced debugger and trace tools to take an Arduino Create* prototype to production.

Seamlessly export a prototype project from Arduino Create and import into Intel® System Studio. From there, you can use the powerful build, analysis, and profiling tools, as well as the debuggers in the Eclipse* IDE to develop the prototype into a production project.

Intel® Distribution for Python*


For instructions to create a Python application in the Intel System Studio IDE, click here.
Get Started

Install Intel System Studio

Intel System Studio supports system and application development for various host and target systems. For detailed requirements, see the System Requirements.

1. To download and install Intel System Studio, click here.
2. If you plan to develop applications for remote Linux targets, you will be prompted to install Docker*. For instructions, see Install Docker for Intel System Studio.
3. Launch Intel System Studio, as described in the table below.

<table>
<thead>
<tr>
<th>Host system OS</th>
<th>Target system OS</th>
<th>How to launch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>Linux</td>
<td>Run this script: /opt/intel/system_studio_2019/iss_ide_eclipse-launcher.sh</td>
</tr>
<tr>
<td>macOS</td>
<td>Linux</td>
<td>Double-click the Intel System Studio icon in the Applications folder.</td>
</tr>
<tr>
<td>Windows</td>
<td>Linux</td>
<td>Double-click the Intel System Studio shortcut on your desktop.</td>
</tr>
<tr>
<td>Windows</td>
<td>Windows</td>
<td>Click Start &gt; Intel &gt; Intel System Studio for Windows Target &gt; Documentation and Getting Started.</td>
</tr>
</tbody>
</table>

4. If you discover problems with the install, check the Intel System Studio community forum or the Online Service Center.

Create a Simple Project

Next, you can create a simple project based on a sample. For instructions, see the quick start information that appears in the IDE after you launch.

What is displayed depends on which host and target systems you are using, as summarized in the table below.

<table>
<thead>
<tr>
<th>Host system</th>
<th>Target system</th>
<th>When you launch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>Linux</td>
<td>The Intel System Studio IDE opens with the Quick Start page, as shown in Quick Start Page (Linux Target).</td>
</tr>
<tr>
<td>macOS</td>
<td>Linux</td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td>Windows</td>
<td>The Getting Started Guide appears, as shown in Getting Started Guide (Windows Target). It explains how to create a project using Microsoft Visual Studio* IDE.</td>
</tr>
</tbody>
</table>
Getting Started Page (Linux Target)

The Getting Started page appears in the IDE when you launch Intel System Studio for a Linux target.

Quick Start tab

First, select one of the tabs on the left side of the page. Then, click the Quick Start links for instructions to create a simple project or analyze your own code.

For detailed information about creating projects, see the Get Started Guides and User Guides on the Intel Developer Zone.
Samples tab
Select the Samples tab to browse the library of samples and tutorials.

Get Started tab
Select the Get Started tab for links to a wide range of documentation.
Getting Started Guide (Windows Target)

The Getting Started guide appears when you launch Intel System Studio for a Windows target.

To get started, expand **Build a Hello World Application** and follow the instructions to create a simple project.
Integrated Development Environment (IDE)

Intel System Studio includes an Eclipse*- based Integrated Development Environment (IDE). The IDE has been customized to work with Intel System Studio, as described in the following sections.

Toolbar

Below is an example of the IDE toolbar.

**Intel System Studio IDE Toolbar**

![Intel System Studio IDE Toolbar](image)

Enhanced Perspectives

An Eclipse perspective is a collection of work areas called views. For example, the C/C++ perspective includes views to edit code, explore files and folders in a project and consoles to view results.

For training and documentation about Eclipse perspectives, see the [Workbench User Guide](#) on the Eclipse website.

We added functionality to certain Eclipse perspectives; for example:

- New Sensor Explorer and Cloud Explorer views in the C/C++ perspective.
- New perspectives for features such as OpenCL™ tools or Energy Analysis.
Sensor Support View

With the Sensor Support view, you can:

- View details about supported sensors
- Add a sensor to your source file
- Add sensor headers to your project

![Sensor Support View Diagram]

Cloud Support View

Cloud support categories, such as Amazon Web Services (AWS) and Microsoft Azure. Sample for the selected cloud support. Click to create a new project using the sample.

![Cloud Support View Diagram]
MKL Browser View

Switching Perspectives

To switch to a different perspective:

1. Select Window from the main toolbar.

2. Select Perspective > Open Perspective > Other.

3. Select a perspective from the list and click Open.
Use Case: Debug a System Bring Up Problem in the BIOS

Problem Statement

You are developing a system on an Intel® Core™ processor. When you attempt to boot the system, it hangs in the BIOS.

Solution

You can use the Intel System Debugger System Trace and System Debug features to identify the location of the problem in the assembly code.

The following table summarizes the steps you need to complete. For detailed instructions, click a step number in the first column.

<table>
<thead>
<tr>
<th>Steps</th>
<th>Tasks</th>
<th>Tools Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Establish a connection to the target board</td>
<td>Target Connection Assistant (TCA) tool¹</td>
</tr>
<tr>
<td>Step 2</td>
<td>Find the BIOS POST code that corresponds to the problem</td>
<td>System Trace¹</td>
</tr>
<tr>
<td>Step 3</td>
<td>Identify the function name and location in the assembly code</td>
<td>System Debug¹</td>
</tr>
</tbody>
</table>

NOTE: If you have not installed the Intel System Debugger, click here to get started. Make sure you include the Intel System Studio IDE when you install.

Instructions

The instructions are written for a Windows* development system and Linux* target system.

Step 1. Establish a connection to the target board

Target Connection Assistant (TCA)
The Target Connection Assistant (TCA) is a tool designed to help you connect to targets and troubleshoot issues regarding connection configuration and connectivity. Once the connection is established and functional, TCA notifies clients, such as System Debug and System Trace, of changes in the connection status.

TCA also allows you to connect to a target using an IPC provider already configured and running on the host. Once the connection is established, TCA forwards the IPC provider information to System Debug and System Trace.
Follow these steps:

1. Launch the Intel System Studio IDE and click the **New Connection** drop-down list on the main toolbar.

2. In the **Wizards** list, select **JTAG Connection for System Debug & Trace** and click **Next**.

3. Select **Create a new connection** and click **Next**.
4. Select your **Target Type** and **Connection Method** and click **Next**.

5. The Intel System Debugger components supported by the selected target type and connection method are listed under **Supported Features**. Verify that **System Debug** and **System Trace** are listed.

6. Enter a **Connection Name**, specify the **Connection configuration details**, and select **Connect on finish**.

7. Click **Finish**.
The connection is automatically established. The red Disconnect button becomes active and the word (Connected) appears in title bar of the Debug Targets Connection tab.

Step 2. Find the BIOS POST code that corresponds to the problem

System Trace feature of Intel System Debugger
If your system or board does not have a POST code display, you can use the System Trace feature of Intel System Debugger to find the POST code message that corresponds to the problem.

About BIOS POST codes
The BIOS normally writes POST codes to the I/O port 0x80. Below is a summary of the BIOS POST codes for a 7th generation Intel® Core™ Customer Reference Board (CRB).

For the latest information for Intel desktop boards, see Port 80h POST Codes for Intel® Desktop Boards.

<table>
<thead>
<tr>
<th>Range</th>
<th>Category/Subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 – 0F</td>
<td>Debug codes: Can be used by any PEIM/driver for debug</td>
</tr>
<tr>
<td>10 – 1F</td>
<td>Host Processors</td>
</tr>
<tr>
<td>20 – 2F</td>
<td>Memory/Chipset</td>
</tr>
<tr>
<td>30 – 3F</td>
<td>Recovery</td>
</tr>
<tr>
<td>40 – 4F</td>
<td>Reserved for future use</td>
</tr>
<tr>
<td>50 – 5F</td>
<td>I/O Busses: PCI, USB, ISA, ATA, and so on</td>
</tr>
<tr>
<td>60 – 6F</td>
<td>Not currently used</td>
</tr>
<tr>
<td>70 – 7F</td>
<td>Output Devices: All output consoles</td>
</tr>
<tr>
<td>80 – 8F</td>
<td>Reserved for future use (new output console codes)</td>
</tr>
</tbody>
</table>
Use Case: Debug a System Bring Up Problem in the BIOS

<table>
<thead>
<tr>
<th>Range</th>
<th>Category/Subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>90 – 9F</td>
<td>Input devices: Keyboard/Mouse</td>
</tr>
<tr>
<td>A0 – AF</td>
<td>Reserved for future use (new input console codes)</td>
</tr>
<tr>
<td>B0 – BF</td>
<td>Boot Devices: Includes fixed media and removable media.</td>
</tr>
<tr>
<td>C0 – CF</td>
<td>Reserved for future use</td>
</tr>
<tr>
<td>D0 – DF</td>
<td>Boot device selection</td>
</tr>
<tr>
<td>E0 – FF</td>
<td>E0 - EE: Miscellaneous codes</td>
</tr>
<tr>
<td></td>
<td>F0 – FF: FF processor exception</td>
</tr>
</tbody>
</table>

Follow these steps:

1. To open the Trace perspective, select Window on the menu bar and then select Perspective > Open Perspective > Other.

2. Select System Trace and click Open.
3. Select **Help me setup System Trace** and click **OK**.

4. Enter a **Project name** and specify a location. Click **Next**.

5. Review the connection settings and click **Next**.
6. Select the configuration folder. Change the default Trace Configuration File name if you wish. Click Finish.

7. Specify configuration settings for the System Trace, as shown on the following screen.
8. In the **Trace Capture** view, click the **Begin capturing trace data from target** button to start live tracing.

![Trace Capture view](image)

9. Locate the BIOS POST code messages in the trace view.

In the highlighted area on the sample screen below, the last two BIOS POST code messages are:

- **0x69 (POSTCODE=<00000069)**
- **0x78 (POSTCODE=<00000078)**

These codes likely correspond to the boot problem.

![Trace View](image)
Step 3. Identify the function name and location in the assembly code

System Debug feature of Intel System Debugger
Use the System Debug feature to set an I/O breakpoint using the POST codes found in the previous step. Then, step through the process to find out what the process does just before it hangs.

Follow these steps:

1. To launch System Debug, select Tools on the menu bar and then select System Debugger > Launch System Debugger - System Debug.
2. Click Yes.
3. To open the Breakpoints view, select View on the menu bar and then select Breakpoints.
4. Before you create a breakpoint, click the yellow Suspend Execution (Pause) button.
5. To open the Create Breakpoint screen, select **Debug** on the menu bar and then select **Create Breakpoint**.

6. To create an I/O breakpoint at the I/O port 0x80, select the **Data** tab and enter specifications as shown below:
   a. **Location**: Enter 0x80
   b. **Condition**: Enter (SYSREG(eax)&0xFF==0x78
   c. **Skip**: 0
   d. **Length**: 0
   e. **IO Read&Write**: Select this option

7. Click **OK**.
8. Select **View** on the menu bar and then select **Instruction Trace**.

![Instruction Trace](image1.png)

9. To start the Trace, on the **Instruction Trace [LBR]** console, click the blue **Enable Trace** camera icon.

![Enable Trace](image2.png)

Instruction tracing is disabled. Use context menu or *trace* command to enable trace. Trace data collection occurs while the target is running.
10. View the Trace results.

The process will pause when one of the POST codes (0x78) is written to port 0x80.

The screen will display the instructions that were executed just before the system paused, as shown below.

You can use this information to identify the root cause. If you have symbols loaded, the System Debugger identifies the function names and provides their source code.

Next Steps
To learn more about debugging with Intel® System Studio, check out our collection of Case Studies and the following documentation:

Getting Started Guide
Intel System Debugger 2019 - System Trace User Guide
Intel Tech Decoded
Use Case: Develop Computer Vision and Deep Learning Optimized Applications

Using the OpenVINO™ Toolkit with Intel® System Studio

You can use the OpenVINO™ toolkit with Intel® System Studio to enhance development of computer vision and deep learning applications. The Intel System Studio IDE includes OpenVINO sample projects to help you get started. And, you can take advantage of the remote debugging capabilities in the IDE.

Solution

This use case explains how to develop and remotely debug sample OpenVINO applications:

- **Security Barrier Camera**: Showcases Vehicle Detection, followed by Vehicle Attributes and License Plate Recognition applied on top of Vehicle Detection.

- **Interactive Face Detection**: This demo showcases the Object Detection task applied to face recognition using a sequence of neural networks.

- **Image Classification**: Performs inference using image classification networks, like AlexNet* and GoogLeNet*. The sample application reads command line parameters and loads a network and an image to the Inference Engine plugin. When inference is complete, the application creates an output image and outputs data to the standard output stream.

- **Hello Infer**: This sample shows you how to use the new Infer Request API of the Inference Engine in applications.

Instructions

The following table summarizes the process for debugging in the Intel® System Studio IDE – from installing prerequisites to running or debugging the sample applications.

For step-by-step instructions, see Get Started Using the OpenVINO™ Toolkit with Intel® System Studio.

Next Steps

To learn more about the OpenVINO toolkit and Intel System Studio, see:

- Get Started Using the OpenVINO™ Toolkit with Intel® System Studio
- Intel System Studio 2019
- Intel Tech.Decoded
Getting Help and Support

Priority Support

Purchased licenses include free updates and direct confidential access to Intel engineers for a year through the Online Service Center.

Forums

Public forums are available for Intel System Studio and component tools and libraries.

- Intel System Studio
- Intel® C++ Compiler
- Intel® Integrated Performance Primitives
- Intel® Math Kernel Library
- Intel® Threading Building Blocks
- Intel® VTune™ Amplifier
- Intel® Graphics Performance Analyzers
- Intel® Inspector
- Intel® System Debugger

Documentation

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Samples and Tutorials</td>
<td>Intel provides downloadable sample code and tutorials that guide you through the basic product features. The sample code illustrates common scenarios, and the corresponding tutorials show how the build tools can be used to create and optimize code.</td>
</tr>
<tr>
<td>Intel System Studio Get Started</td>
<td>Links to Get Started Guides for all Intel System Studio components</td>
</tr>
<tr>
<td>Intel System Studio Featured</td>
<td>Links to articles that address a wide range of Intel System Studio topics</td>
</tr>
<tr>
<td>Documentation</td>
<td>A downloadable ZIP file containing all Intel® System Studio documentation is available for offline use.</td>
</tr>
<tr>
<td>Intel System Studio Product Page</td>
<td>Support and the latest online documentation</td>
</tr>
<tr>
<td>Intel® Tech.Decoded</td>
<td>An information hub designed for developers</td>
</tr>
</tbody>
</table>