Intel® System Studio 2020 Initial Release
Release Notes

4 December 2019

Contents

1   Introduction

This document provides an overview of the **Intel® System Studio 2020** product and provides pointers to where you can find new features and changes, the release history, installation instructions additional product information and references to articles and white papers.

The Intel® System Studio has separate download packages for Linux* and Windows* hosts.

The target audience is the performance-oriented C/C++ embedded/mobile/wearable/IoT developer who is developing on Linux*, Windows*, and/or macOS* host environments for Yocto Project* embedded Linux*, Wind River* Linux*, and/or Android* targets.

For full product information for the previous release, as well as a link to renewable 90-day community license, please refer to Intel® System Studio product webpage https://software.intel.com/intel-system-studio.

For licensing information, please refer to the Intel End User Licensing Agreement (EULA) available at https://software.intel.com/articles/end-user-license-agreement.

2   System Requirements and What's New

System requirements and what's new in Intel® System Studio 2020 are available online: https://software.intel.com/en-us/articles/intel-system-studio-release-notes-whats-new

3   Product Contents and Cross Reference

The following table outlines which versions of the Intel® Software Development Tools are present in Intel® System Studio 2020.

Component	Version
Docker* based build system	2020.0
Eclipse* IDE	2.2.8 (2019-09)
Yocto Project* Compatible Application Development Plugins	2.2.8
GNU* GDB and source	8.3.0
Intel® C/C++ Compiler	19.1.0
Intel® Data Analytics Acceleration Library (Intel® DAAL)	2020.0.0
Intel® Integrated Performance Primitives (Intel® IPP)	2020.0.0
Intel® Math Kernel Library (Intel® MKL)	2020.0.0
Intel® Threading Building Blocks (Intel® TBB)	2020.0.0
OpenCL™ Tools	2020
IoT Connection Tools (UPM / MRAA / Cloud Connectors)	2020
Sample Applications	N/A
Intel® Advisor 2020	2020.0.0
Intel® Inspector 2020	2020.0.0
Intel® SoC Watch for Android* targets	2020.1
Intel® SoC Watch for Linux* targets	2020.1
Intel® SoC Watch for Windows* targets	2020.1
Intel® VTune™ Profiler 2020	2020.0.0
Intel® VTune™ Platform Profiler	2020.0.0
Intel® Debug Extensions for WinDbg*	2020.0
Intel® System Debugger (System Debug & System Trace)	2020.0

Starting with Intel® C++ Compiler 18.0 Gold Release, Intel® Cilk™ Plus will be marked as deprecated and eventually removed in a future release. To learn how to migrate to OpenMP* or Intel® Threading Building Blocks, see this article:

https://software.intel.com/en-us/articles/migrate-your-application-to-use-openmp-or-intelr-tbb-instead-of-intelr-cilktm-plus

Release notes for individual components are linked to from the main release notes page:

https://software.intel.com/en-us/articles/intel-system-studio-release-notes-whats-new

This document uses this label to represent the directory path where the product is installed: *<INSTALL_DIR>*

Where the *<INSTALL_DIR>* is by default:

Windows* Host:

      Windows Target: C:\Program Files (x86)\IntelSWTools\system_studio_2020

      Linux Target: C:\IntelSWTools\system_studio_2020

Linux* Host:

      sudo/root install: /opt/intel/system_studio_2020

macOS*:

      /opt/intel/system_studio_2020

3.1 Intel® Software Manager

The Intel® Software Manager, automatically installed with the Intel® System Studio product on Windows* host systems, is a graphical tool that provides a simplified delivery mechanism for product updates, current license status, and news on all installed Intel Software Development tools.

It can also be manually started as well from these locations:

- Windows* 8.x/10: Launch the *Intel® Software Manager* application for the start screen
- Windows* 7: *Start / Intel System Studio 2020 / Intel Software Manager*

The software manager from this release replaces any previous installed software manager and manages all installed Intel® Software Development Tools licenses on the system.

When you install Intel® System Studio, we collect information that helps us understand your installation status and environment. Information collected is anonymous and is not shared outside of Intel. See https://software.intel.com/en-us/articles/data-collection for more information on what is collected and how to opt-out.

You can also volunteer to provide Intel anonymous usage information about these products to help guide future product design. This option, the Intel® Software Improvement Program, is not enabled by default – you can opt-in during installation or at a later time, and may opt-out at any time. For more information please see http://intel.ly/SoftwareImprovementProgram

4 Technical Support and Documentation

4.1 Technical Support

Registration entitles you to free technical support, free product updates, and free product upgrades for the duration of the support term.

Technical support is provided through [Intel Online Service Center](#). You will receive private support for questions on the product. Select the "Intel® System Studio" product when submitting questions on the product. See [this article](#) for step-by-step guidance on submitting a support request.

Additionally, you may submit questions and browse issues in the [Intel® System Studio User Forum](#).

For additional information about how to find Technical Support, please visit: https://software.intel.com/intel-system-studio-support.

Note: If your distributor provides technical support for this product, please contact them for support rather than Intel.

4.2 Documentation Locations

The main page for additional information, to download the package, the latest technical documentation, and information about finding technical support can be found here: https://software.intel.com/en-us/system-studio/documentation

5   Installation Notes and Log Files

Please refer to the System Requirements to check the prerequisites for installing the Intel®
System Studio 2020 product.

If you run into issues installing the tools, you can refer to the README provided with the
installer for more information.

Additionally, if you would like to see the install logs (helpful for reporting issues) you can find
them here:

Linux* Host / Linux Target:

The Intel System Studio installer writes log files to /tmp, one for the user and one for root
(when the installer is run with sudo). These log file names start with intel.pset, end with a
timestamp and have the extension .log.

Windows* Host:

The Intel System Studio installer writes log files to %TEMP%\Intel\. The primary installer log
file names begin with intel.pset and have the extension .log, and other Intel System Studio
component installation log files have the extension .log. Additionally, if Intel System Studio
components are integrated into Visual Studio*, VSIXInstaller_*.log and dd_setup_*.log files are
written to the %TEMP% directory.

macOS* Host / Linux* Target:

The Intel System Studio installer writes log files to the system temp directory (echo $TMPDIR),
one for the user and one for root (when the installer is run with sudo). These log file names
start with intel.pset, end with a timestamp and have the extension .log.

6   Known Issues and Limitations

For the complete list of known issues of individual Intel® System Studio 2020 components, please refer to the individual component release notes: https://software.intel.com/en-us/articles/intel-system-studio-release-notes-whats-new

6.1   General Known Issues and Limitations

6.1.1   Intel® System Studio 2020 Windows* target cannot co-exist with older versions

Intel® System Studio 2020 Windows* target contains new installation technology which is not fully compatible with Intel System Studio 2019 or Intel® Parallel Studio XE 2019, and earlier. To install Intel System Studio 2020 on a system where Intel System Studio 2019 or Intel Parallel Studio XE 2019 or earlier has been installed, first uninstall the older version.

6.1.2   Intel® System Studio installation on Windows* host may cause a system restart

While installing Intel® System Studio Ultimate Edition on a Windows* host, the system may try to reboot. If the reboot is cancelled, the Intel System Studio installer may display some error messages. To work around this issue, re-run the Intel System Studio installation.

6.1.3   Some hyperlinks in HTML documents may not work with Internet Explorer*

Try using another browser, such as Google Chrome* or Mozilla Firefox*, or right-click the link, select Copy shortcut, and paste the link into a new Internet Explorer* window.

6.1.4   Host Operating System Requirements

Intel® VTune™ Profiler, Intel® Advisor, and Intel® Inspector graphical user interfaces may require newer versions of host operating systems. See the individual components' release notes for details.

6.1.5   End User License Agreement

The top-level third-party-programs.txt file in the Intel® System Studio Composer and Professional Edition packages incorrectly states that Intel® System Debugger is distributed under the "End User License Agreement for the Intel(R) Software Development Products (Version October 2018)" license. The Intel System Debugger is distributed under the "Intel End User License Agreement for Developer Tools (Version September 2019)" license.

NOTE: The Intel System Debugger tool is distributed as part of the Ultimate Edition package of Intel System Studio. The license statement regarding Intel System Debugger in the top-level third-party-programs.txt file, in the Ultimate Edition package, is correct.

6.2   Issues and Limitations by Component

Component	Description	Implication	Workaround
Application launch (ISS-3029)	The local run configuration for native cmake projects is not auto created.	Users will be unable to launch a local application by clicking run and choosing Local.	Users can add a manual local run configuration and set the field value of "C/C++ Application" to one of the project binaries. Then, click Run.
Docker* based build system	Docker is not provided with Intel® System Studio installation.	Container based project will fail on these systems without additional setup.	Users must follow the Installing Docker* for Intel System Studio guide to build their projects: • https://software.intel.com/en-us/intel-system-studio-docker-install-why
Docker* based build system	Installing Intel System Studio onto a virtual machine is only supported by and has only been tested for Ubuntu* Desktop 16.04 and 18.04 Linux* guest virtual machines. Other guest operating systems (Windows* and macOS*) have not been tested and are not supported.	Some Intel System Studio features (e.g., Docker* and Intel VTune Amplifier) require access to low-level CPU features that are not supported by all virtual machine managers or hypervisors or are not enabled by default.	Install Intel System Studio onto a "real" Windows* or macOS* system, not a guest VM. If you must use a VM we recommend using an Ubuntu 16.04 or 18.04 guest VM with "nested virtualization" enabled and with at least 4GB of RAM dedicated to the VM
Docker* based build system	Removal or manual modification of Docker images and containers created by Intel System Studio (especially while Intel System Studio is running) may result in errors.	Existing projects may stop working. New projects in an existing workspace will not work.	To recover after removing a container or image, restart Intel System Studio. When both image and container have been removed, Intel System Studio prompts the user to redownload the image. The container will be recreated after the image has been downloaded.
Docker* based build system	Each release of Intel System Studio comes with a standard Docker* image that has been updated since the previous release. When a new release of Intel System Studio is installed, the updated image is not automatically installed.	The user can either continue using existing Intel System Studio Docker* image (e.g., if the image has been customized), or install the new Docker image.	To install a new Docker* image, see Installing a New Intel System Studio Docker Image at https://software.intel.com/en-us/intel-system-studio-docker-update-image
Docker* based build system (ISS-2091)	When the proxy bypass includes entries other than localhost and 127.0.0.1, the project creation internet connectivity check may fail even when a valid connection is available.	When this condition occurs, new projects for "building in a container" cannot be created.	Temporarily remove non-default entries from proxy bypass list when creating a new project. After project creation internet connectivity check succeeds, add the entries back to proxy bypass list.
Docker* based build system (ISS-2823)	Installing a Docker image in Intel System Studio on Windows* may cause a "Failed to update the includes" popup.		Click "OK" and continue to use the Docker image.

Docker* based build system (ISS-3552)	The first time a project is built in a workspace using an Ubuntu 18.04 container, the Eclipse* C++ indexer cannot correctly identify the C++ standard library paths and reports indexing errors, even though the project can be built without errors as can be verified in the console.	This problem is specific to the first project in the workspace and won't happen for subsequent projects.	To resolve this issue, close Eclipse and reopen it.

Eclipse* based IDE	On Red Hat* and CentOS* Hosts the welcome screen in the Intel System Studio IDE may be empty, non-responsive or display an error message.	Users will be unable to read welcome screen content.	User must install "epel-release" and "webkitgtk" packages using the following commands at a bash shell (in a terminal window):
			$ sudo yum install epel-release
			$ sudo yum install webkitgtk

Eclipse* based IDE	The Sensor View is not supported in the projects imported from Arduino* Create.	Users will not see the sensor view for projects imported from Arduino* Create.	Add sensors library code manually. Create a new C/C++ Docker*-based project, and use it to access the Sensor View.
Eclipse* based IDE (ISS-1877)	Creating a new project with the same name as an existing project folder results in an "Invalid thread access" error message.	The new project may fail to be created or files may be added and/or replaced in the existing project folder.	Do not use the name of an existing folder in the workspace. Open the workspace folder using OS file explorer to see all folders present, as some folders / projects may not appear in the Eclipse file / project explorer.
Eclipse* based IDE (ISS-1946)	Connecting to an application development target from Intel System Studio IDE freezes at 25% progress after initial connection succeeded.	The project cannot be launched on the target.	To work around this issue, reboot the target.
Eclipse* based IDE (Linux* host) (ISS-1953)	By default, Intel System Studio displays web content using the Eclipse internal browser.	The internal browser uses global proxy settings configured via Network Manager, not the Eclipse proxy configuration. When global proxy settings are not configured to allow internet access, external web content cannot be displayed in the Eclipse internal browser.	If it is not possible to configure global proxy settings (e.g., Network Manager is not available), web content can be displayed in an external browser. Open Preferences->General->Web Browser, then select "Use external web browser." Ensure the selected browser has internet access. When web content links are selected in Intel System Studio, the content will be displayed in the selected browser.

Eclipse* based IDE (ISS-1563)	Wind River* Linux* Connection Wizard may not have field to enter IP address or name.	User may not be able to create connection.	Resize the connection wizard dialog box. The UI should update itself and the field to enter IP address and name will become available.
Eclipse* based IDE (ISS-2060)	Intel System Studio has Intel Advisor samples. These samples require advisor variable script to be sourced.	Advisor samples will not work without sourcing the advixe-var.sh script.	Before launching Intel System Studio, open a terminal and follow these commands: 1. source <install_dir>/advisor_2020/advixe-vars.sh 2. launch eclipse from same terminal using ./iss_ide_eclipse-launcher.sh
Eclipse* based IDE (ISS-2129)	Remote target connection setup fails silently on some Linux systems.	After creating a new run or debug configuration using the "C/C++ Remote Application" wizard, no new connection appears in the list. This affects the normal workflow for projects imported from Arduino* Create.	In Intel System Studio, select Settings > General > Security > Secure Storage, uncheck "Linux integration" master password provider and restart Intel System Studio.
Eclipse* based IDE (ISS-2217)	Connecting to remote target fails because secure storage fails silently on some Linux systems.	When connecting to a remote target, the user is repeatedly prompted for login credentials, even after providing correct user name and password.	In Intel System Studio, select Settings > General > Security > Secure Storage, uncheck "Linux integration" master password provider and restart Intel System Studio.
Eclipse* based IDE (ISS-2732)	When a Python* project is created using Intel Application Development -> Python Project, creating additional Python source files fails.	The user is unable to add new Python source file in projects created automatically based on provided samples.	When creating a new file, use basic "File" action instead of "Python Module" and use the file extension ".py".
Eclipse* based IDE (ISSDEV-2455)	"Save credentials" checkbox in the Connection Login dialog does not work on some Linux systems.	After entering username and password and checking the "Save credentials" checkbox, credentials are not actually saved.	In Intel System Studio, select Settings > General > Security > Secure Storage, uncheck "Linux integration" master password provider and restart Intel System Studio.

Eclipse* based IDE (ISS-2405)	"Intel Application Development" > "Project to cross compile C/C++ for Linux and Android targets" project setup dialog is confusing.	This workflow requires that pre-requisites have been installed. The pre-requisites are listed in the Intel® System Studio 2020 System Requirements document, but are not listed in Intel C++ Compiler 19.0 documentation referenced by the workflow.	gcc and binutils must be installed to use Intel C++ Compiler 19.0 (refer to system requirements at https://software.intel.com/en-us/articles/intel-system-studio-release-notes-whats-new). On Ubuntu, these can be installed via the build-essential package (`sudo apt-get install build-essential`). Other Linux distributions have similar packages. On Windows, a cross-compile environment such as Cygwin* or mingw* is required.
Eclipse* based IDE (ISS-2405)	By default, new C/C++ Remote Application Run configurations use the "Legacy Remote Create Process Launcher."	In some cases, this launcher is unable to connect to correctly configured targets.	The "Remote Launching via TCF/TE Launcher" is compatible with connections created using the Connection toolbar in Intel System Studio and may be used instead of the "Legacy Remote Create Process Launcher". To switch to the "Remote Launching via TCF/TE Launcher", click the "Select other" link at the bottom of the "Run Configurations" dialog. Click "Change Workspace Settings" in the "Select Preferred Launcher" dialog. Scroll down to "C/C++ Remote Application" and select [Run]. In the "Preferred Launcher" pane, pick "Remote Launching via TCF/TE Launcher." Click "Apply and Close." Click OK on the "Select Preferred Launcher" dialog. The Connection selection control has been updated to allow selection of connections created using the Connection toolbar.
Eclipse* based IDE (ISS-2815)	Internet connection is not detected if proxy is set to native on Linux* host systems.	Users cannot create new projects based on Intel System Studio Samples. The "Checking Internet Connection" dialog keeps attempting to connect.	There are two workarounds: Workaround 1: Use launcher script a. Shut down Intel System Studio. b. Launch a terminal window. Ensure proxy values are set in environment variables. Browse to Intel System Studio installation directory. c. Launch Intel System Studio from

			the terminal by using ./iss_ide_launcher_script.sh Workaround 2: Use manual proxy a. Click on "Change Network settings" b. Browse to General > Network Connections. c. Choose 'Manual' as Active Provider. Add the values of your proxy settings.
Eclipse* based IDE (ISS-3026)	Switching between multiple remote target systems within a project does not always run the application on the correct target system.	When switching among multiple remote target systems within a project, the project does not always run on the correct target system.	1) Delete the target connection. 2) Create the target connection. 3) Run the project.
Eclipse* based IDE (ISS-3597)	When trying to integrate ICC recipes with Wind River* Linux* (WRL) LTS 17 (and above), the build fails with the following error message: install: cannot stat \u2018/work/windriver_lts17/ layers/meta-intel-iss/../../ compilers_and_libraries/linux/ lib/intel64_lin/liboffload.so\ u2019: No such file or directory. This is because in the latest ICC compiler shipped with Intel® System Studio 2020, the following libraries were removed: liboffload.so, liboffload.so.5, offload_main.		1) Go to <iss-install-dir>/YoctoProject/meta-intel-iss/ recipes-icc/intel-iss-icc-runtime/ intel-iss-icc-runtime_17.0.bb 2) Remove instances of: liboffload.so, liboffload.so.5, and offload_main from the following line: icc_runtime_intel64="cilk_db.so …" 3) The modified line will look like: icc_runtime_intel64="cilk_db.so libchkp.so libcilkrts.so libcilkrts.so.5 libimf.so libintlc.so libintlc.so.5 libiomp5.so libiompstubs5.so libirc.so libirng.so libpdbx.so libpdbx.so.5 libsvml.so" ICC recipes will now successfully build with WRL LTS 17 (and above).
Energy Analysis Plugin	Failure to build Intel® SoC Watch drivers on the target.	Users cannot run an Energy Analysis collection without manually installing the driver.	Users must follow the Preparing a Target Linux* System for Energy Analysis guide to install required drivers: https://software.intel.com/en-us/energy-analysis-user-guide-preparing-a-target-linux-system-for-energy-analysis
GNU* GDB	By default the "Function call history" is empty after enabling reverse debugging option.	The user cannot see any history in the Function call history window.	Send "record btrace pt" command manually once debugging is started from the Debugger Console window.

Intel® Data Analytics Acceleration Library	Intel® DAAL Python API (a.k.a. pyDAAL) is provided as source.	When building it on Windows, users may see warning messages.	Ignore the warnings, the messages do not indicate critical issues and do not affect the library's functionality.
OpenCL™ Tools	The OpenCL™ CPU Kernel Debugger feature in this release of OpenCL™ Tools requires the latest update of Intel® CPU Runtime for OpenCL™ Tools, available for [Linux*](#) and [Microsoft Windows*](#) OSes.		For more information, please see [OpenCL™ Tools Release Notes](#) and for any questions, see the [OpenCL* Support Forum](#).
How-to-Code Samples	How-to-code samples may not work for all target operating systems.	Samples might not compile for all target operating systems, resulting in build errors.	None. Updates to these samples will be made over the next weeks/months and will be dynamically available.
Samples (ISS-1859)	Certain valid code statements, including some used in samples, result in false error messages. For example, the statement: std::cout << "Payload Length : " << payload.length() << std::endl; Results in the error: Invalid overload of 'std::endl' These false errors are caused by incorrectly configured static analyzer rules and the code should still compile.		This particular false-positive error can be resolved by converting the code into a multi-line statement: std::cout << "Payload Length : " << payload.length(); std::cout << std::endl;
Sensor Explorer (ISS-2504)	Using a right-click with the sensor explorer to remove sensors from Java* projects does not work.	Removing a sensor from a Java project using the sensor explorer by right-clicking sensor and selecting 'Remove import(s) from current source file' may fail.	Use the checkbox provided to the left of each sensor to add/remove sensors from the project.
Sensor Support (ISS-2551)	The Sensor Support feature is not compatible with Java language projects in this release of System Studio.	Users are unable to use the Sensor Support view to add or remove sensors to such projects and an error message is displayed, when open.	Users can add or remove sensors to projects manually by editing the source code files and by appending the UPM Java package dependencies to the project's pom.xml file.

Sensor Support (ISSDEV-2783)	The Sensor Support feature is not compatible with Python language projects in this release of System Studio.	Users are unable to use the Sensor Support view to add or remove sensors to such projects and an error message is displayed when open.	Users can add or remove sensors to projects manually by editing the source code files and by appending the UPM Python module installation directory to the project's PYTHONPATH.
Visual Studio*	Installation of Intel System Studio with Microsoft Visual Studio* 2017 integration hangs and fails on some systems.	User may see errors or the installation may complete successfully with no error/crashes, however, the integration to Visual Studio* 2017 is not installed.	Installing Visual Studio* 2017 version 15.3.3 may resolve this issue. For additional workaround options, see: https://software.intel.com/en-us/articles/intel-software-development-tools-integration-to-vs2017-issue
Yocto* compatible Linux* Platform Project Development	If the default browser is already open and Intel System Studio is installed as a root user, Toaster will not be launched in the default browser.	Users will be unable to configure their platform project via Toaster.	Close any running instance of the default browser before attempting to create a platform project in Intel System Studio, when Intel System Studio is installed as root.
Yocto* compatible Linux* Platform Project Development (ISS-2075)	The message "Network access disabled through BB_NO_NETWORK" appears while fetching linux-yocto.	The kernel source (which is included in the wrlinux layer) has been updated since Toaster was started and Toaster did not detect the change.	1. Remove the older wrlinux cloned repository (older kernel repository): $ rm –rf <installdir>/_toaster_clones/_git___lxgit.wrs.com_git_layers_wrlinux_WRLINUX_10_17_LTS 2. Open the project in Toaster via the KERNEL CONFIGURATION TOOL. 3. Click Update. 4. This will re-clone the repository and things will again be in sync.

7 Attributions

This product includes software developed at:

The Apache Software Foundation (http:\\www.apache.org\).

Portions of this software were originally based on the following:
- software copyright (c) 1999, IBM Corporation., http:\\www.ibm.com.
- software copyright (c) 1999, Sun Microsystems., http:\\www.sun.com.
- the W3C consortium (http:\\www.w3c.org) ,
- the SAX project (http:\\www.saxproject.org)
- voluntary contributions made by Paul Eng on behalf of the
Apache Software Foundation that were originally developed at iClick, Inc.,
software copyright (c) 1999.

This product includes updcrc macro,
Satchell Evaluations and Chuck Forsberg.
Copyright (C) 1986 Stephen Satchell.

This product includes software developed by the MX4J project
(http:\\mx4j.sourceforge.net).

This product includes ICU 1.8.1 and later.
Copyright (c) 1995-2006 International Business Machines Corporation and others.

Portions copyright (c) 1997-2007 Cypress Semiconductor Corporation.
All rights reserved.

This product includes XORP.
Copyright (c) 2001-2004 International Computer Science Institute

This product includes software from the book
"Linux Device Drivers" by Alessandro Rubini and Jonathan Corbet,
published by O'Reilly & Associates.

This product includes hashtab.c.
Bob Jenkins, 1996.

8 Disclaimer and Legal Information

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel, the Intel logo, VTune, Cilk, Atom, and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries.

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice Revision #20110804

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

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos.

© Intel Corporation.