

Intel® System Studio 2016 Update 1 Installation Guide and Release Notes

Installation Guide and Release Notes for Linux* Host

16 November 2016

Contents

1	Introduction	5
2	What's New	6
	Intel® System Studio 2016 Update 1	6
3	Intel® Software Manager	6
4	Product Contents	7
5	Getting Started	8
6	Technical Support and Documentation	8
6.1	Release Notes Location	8
6.2	Article & Whitepaper Locations	10
6.3	Support	10
6.4	Support for native code generation for Intel® Graphics Technology	10
7	System Requirements	11
7.1	Supported Host Platforms	11
7.2	Eclipse* Integration Prerequisites	12
7.3	Host Prerequisites and Resource Requirements	12
7.3.1	Host Space Requirements by Component	12
7.3.2	Intel® Integrated Performance Primitives (Intel® IPP) Details	12
7.3.3	Intel® C++ Compiler	13
7.4	Target Software Requirements	13
7.5	Target Prerequisites and Resource Requirements	13
7.5.1	Target Space Requirement by Component	13
7.5.2	Intel® VTune™ Amplifier target OS kernel configuration	14
7.5.3	Intel® VTune™ Amplifier Feature vs. Resource Matrix	15

7.6	Hardware Requirements	15
7.7	Intel® Graphics Technology development specific requirements	16
	Up-to-date information on hardware, operating system and driver requirements for offloading computations to the integrated processor graphics can be found on the following page:.....	16
	https://software.intel.com/en-us/articles/getting-started-with-compute-offload-to-intelr-graphics-technology	16
8	Installation Notes	17
8.1	Installing the Tool Suite.....	17
8.2	Installation directory structure	17
8.3	Development target package installation.....	18
8.3.1	Intel® Inspector Command line interface installation.....	20
8.3.2	Preparing a Target Android* System for Remote Analysis	20
8.3.3	Intel® VTune™ Amplifier Collectors Installation on Remote Linux* Systems	20
8.3.4	Intel® VTune™ Amplifier Sampling Enabling Product Installation on Remote Linux* Systems	21
8.3.5	Intel® Integrated Performance Primitives redistributable shared object installation 21	
8.3.6	Intel® Math Kernel Library redistributable shared object installation	21
8.3.7	Intel® C++ Compiler dynamic runtime library installation	21
8.4	Eclipse* IDE Integration	22
8.4.1	Installation	22
8.4.2	Launching Eclipse for Development with the Intel C++ Compiler	22
8.4.3	Editing Compiler Cross-Build Environment Files	22
8.4.4	Cheat Sheets.....	23
8.4.5	Integrating the Intel® System Debugger into Eclipse*	23
8.5	Wind River* Workbench* IDE Integration	23
8.5.1	Documentation.....	23
8.5.2	Installation	23
8.5.3	Manual installation	24
8.5.4	Uninstall.....	24
8.6	Installing Intel® XDP3 JTAG Probe.....	24

Ordering JTAG Device for Intel® System Debugger	24
8.7 Removing the Product	24
9 Issues and Limitations.....	25
9.1 General Issues and Limitations.....	25
9.1.1 Use non-RPM installation mode with Wind River* Linux* 5 and 6	25
9.1.2 Intel® Software Manager unsupported on Wind River* Linux* 5.0, Ubuntu* 12.04. 25	
9.1.3 Installation into non-default directory on Fedora* 19 may lead to failures.....	25
9.1.4 Running online-installer behind proxy server fails	25
9.1.5 The online-installer has to be run with sudo or root access	25
9.1.6 Some hyperlinks in HTML documents may not work when you use Internet Explorer.	26
9.2 Wind River* Linux* 7 Support.....	26
9.2.1 Windows* host is currently not supported for Wind River* Linux* 7 targeted development	26
9.2.2 No integration into Wind River* Workbench* IDE is currently available for Wind River* Linux* 7 target	26
9.2.3 Remote event-based sampling with Intel® VTune™ Amplifier Limitations.....	26
9.3 Intel® Energy Profiler.....	26
9.3.1 /boot/config-‘uname -r’ file must be present on platform.	26
9.3.2 Power and Frequency Analysis support for Intel® Atom™ Processor covers Android* OS only.	26
9.4 Intel® VTune™ Amplifier Usage with Yocto Project*	27
9.4.1 Building Sampling Collector (SEP) for Intel® VTune™ Amplifier driver on host Linux* system	27
9.4.2 Remote Intel® VTune™ Amplifier Sampling on Intel® 64 Yocto Project* Builds ..	27
9.4.3 Building 64bit Sampling Collector against Yocto Project* targeting Intel® Atom™ Processor E38xx requires additional build flags	27
9.5 Intel® System Studio System Analyzer	27
9.5.1 Supported Linux* Distributions.....	27
9.5.2 The path for the Intel® System Studio System Analyzer does not get set up automatically.....	27

9.5.3	Support for Intel® Atom™ Processor Z3560 and Z3580 code-named “Moorefield” missing	28
9.6	Intel® System Debugger	28
9.6.1	Intel® Puma™ 6 Media Gateway Firmware Recovery Tool not available	28
9.6.2	Connecting to Intel® Quark™ SoC may trigger error message that can be ignored	28
9.6.3	Using the symbol browser on large data sets and large symbol info files not recommended	28
9.6.4	Limited support for Dwarf Version 4 symbol information	28
9.7	GDB* - GNU* Project Debugger	28
9.7.1	Eclipse* integration of GDB* requires Intel® C++ Compiler install	28
9.8	Intel® C++ Compiler	28
9.8.1	“libgcc_s.so.1” should be installed on the target system	28
10	Attributions	29
11	Disclaimer and Legal Information	30

1 Introduction

This document provides a brief overview of the Intel® System Studio 2016 and provides pointers to where you can find additional product information, technical support, articles and whitepapers.

It also explains how to install the Intel® System Studio product. Installation is a multi-step process and may contain components for the development host and the development target. Please read this document in its entirety before beginning and follow the steps in sequence.

The Intel® System Studio consists of multiple components for developing, debugging, tuning and deploying system and application code targeted towards embedded, Intelligent Systems, Internet of Things and mobile designs.

The tool suite covers several different use cases targeting development for embedded intelligent system platforms ranging from Intel® Atom™ Processor based low-power embedded platforms to 3rd, 4th, 5th and 6th generation Intel® Core™ microarchitecture based designs. Please refer to the Intel® System Studio User's Guide for guidance on how to apply Intel® System Studio to the various use case scenarios that are available with this versatile product.

Due to the nature of this comprehensive integrated software development tools solution, different Intel® System Studio components may be covered by different licenses. Please see the licenses included in the distribution as well as the [Disclaimer and Legal Information](#) section of these release notes for details.

2 What's New

Intel® System Studio 2016 Update 1

1. Intel® C++ Compiler:
 - Enhancements for offloading to Intel® Graphics Technology
2. Intel® Energy Profiler (SoC Watch):
 - Added support for collection of gfx-cstate and ddr-bw metrics on platforms based on Intel® Core™ architecture.
3. Intel® System Debugger:
 - New options for the debugger's "Restart" command
 - System Trace Viewer:
 - New "Event Distribution View" feature
 - Several improvements in the Trace Viewer GUI.

3 Intel® Software Manager

The installation now provides an Intel® Software Manager to provide a simplified delivery mechanism for product updates and provide current license status and news on all installed Intel software products.

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 Product Contents

The product contains the following components:

1. Intel® C++ Compiler 16.0 Update 1 (64-bit host only)
2. Intel® Integrated Performance Primitives 9.0 Update 1 for Linux*
3. Intel® Math Kernel Library 11.3 Update 1 for Linux*
4. Intel® Threading Building Blocks 4.4 Update 1
5. Intel® System Debugger 2016 (64-bit host only)
 - 5.1. Intel® System Debugger notification module xdbntf.ko (Provided under GNU General Public License v2)
6. OpenOCD 0.8.0 library (Provided under GNU General Public License v2+) (64-bit host only)
 - 6.1. OpenOCD 0.8.0 source (Provided under GNU General Public License v2+)
7. GNU* GDB 7.8.1 (Provided under GNU General Public License v3) (64-bit host only)
 - 7.1. Source of GNU* GDB 7.8.1 (Provided under GNU General Public License v3)
8. Intel® VTune™ Amplifier 2016 Update 1 for Systems with Intel® Energy Profiler
9. Intel® Inspector 2016 for Systems
10. Intel® Graphics Performance Analyzers 2015 R3

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 optimizations 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 User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

5 Getting Started

Please refer to the Getting Started Guide and Intel® System Studio User's Guide for guidance on Intel® System Studio use cases and supported usage models.

Intel® System Studio User's Guide

- `<install-dir>/documentation_2016/en/iss2016/iss_ug.pdf`

Intel® System Studio Getting Started Guide

- `<install-dir>/documentation_2016/en/iss2016/iss_gsg_lin.htm`

6 Technical Support and Documentation

6.1 Release Notes Location

The release notes and getting started guide for the tools components making up the Intel® System Studio product can be found at the following locations after unpacking `system_studio_2016.1.xxx.tgz` and running the `install.sh` installation script or running the online installer `system_studio_2016.1.xxx_online.sh`.

The paths are given relative to the installation directory `<install-dir>`. The default installation directory is `/opt/intel` for (sudo)root installations and the user home directory `($HOME)/intel` for user installations unless indicated differently.

Intel® System Studio Release Notes and Installation Guide

- `<install-dir>/documentation_2016/en/iss2016/all-release-install.pdf`

Intel® C++ Compiler

- `<install-dir>/documentation_2016/en/compiler_c/ReleaseNotes_ISS_Compiler.pdf`

Intel® Integrated Performance Primitives

- `<install-dir>/documentation_2016/en/ipp/iss2016/ReleaseNotes.htm`

Intel® Math Kernel Library

- `<install-dir>/documentation_2016/en/mkl/common/ReleaseNotes.htm`

Intel® Threading Building Blocks

- `<install-dir>/documentation_2016/en/tbb/common/Release_Notes.txt`

Intel® System Debugger

- `<install-dir>/documentation_2016/en/debugger/iss2016/system_debugger/sysdebug-release-install.pdf`

GDB

- `<install-dir>/documentation_2016/en/debugger/iss2016/gdb/GDB_Release_notes.pdf`
- `<install-dir>/documentation_2016/en/debugger/iss2016/gdb/GDB.pdf`

Intel® VTune™ Amplifier for Systems

- `<install-dir>/documentation_2016/en/vtune_amplifier_for_systems/release_notes_amplifier_for_systems_linux.pdf`

Intel® Inspector for Systems

- `<install-dir>/documentation_2016/en/inspector_for_systems/Release_Notes_Inspector_Linux.pdf`

Intel® VTune™ Amplifier Sampling Enabling Product

- The user's guide explaining the usage of the SEP command line tool for hardware event-based sampling collection on embedded devices can be found at `<install-dir>/documentation_2016/en/vtune_amplifier_for_systems/SEP_User_Guide.pdf`

WakeUp Watch for Android*

- The user's guide explaining usage can be found at `../system_studio_target/wuwatch_android/WakeUpWatchForAndroid.pdf` after unpacking the `<install-dir>/system_studio_2016.1.xxx/targets/system_studio_target.tgz` package.

SoC Watch for Android*

- The user's guide explaining usage can be found at `../system_studio_target/socwatch_android_v1.x.x/SoCWatchForAndroid_v1_x_x.pdf` after unpacking the `<install-dir>/system_studio_2016.1.xxx/targets/system_studio_target.tgz` package.

Intel® System Studio System Analyzer

- <http://software.intel.com/en-us/articles/intel-graphics-performance-analyzers-for-android-os>
- <http://software.intel.com/en-us/articles/intel-gpa-online-help>

6.2 Article & Whitepaper Locations

Intel® System Studio Tutorials and Samples

- `<install-dir>/documentation_2016/en/iss2016/samples-and-tutorials.html`

Intel® System Studio Articles and Whitepapers

- <https://software.intel.com/en-us/articles/intel-system-studio-articles>
- For a list of all available articles, whitepapers and related resources please visit the Intel® System Studio product page at <http://software.intel.com/en-us/intel-system-studio> and look at the Support tab.

6.3 Support

If you did not register your compiler during installation, please do so at the [Intel® Software Development Products Registration Center](#). Registration entitles you to free technical support, product updates and upgrades for the duration of the support term.

To submit issues related to this product please visit the [Intel Premier Support](#) webpage and submit issues under the product **Intel(R) System Studio**.

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

For information about how to find Technical Support, product documentation and samples, please visit <http://software.intel.com/en-us/intel-system-studio>.

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

6.4 Support for native code generation for Intel® Graphics Technology

By default, the compiler will generate virtual ISA code for the kernels to be offloaded to Intel® Graphics Technology. The vISA is target independent and will run on processors that have the Intel graphics processor integrated on the platform and that have the proper Intel® HD Graphics driver installed. The Intel HD Graphics driver contains the offload runtime support and a Jitter (just-in-time compiler) that will translate the virtual ISA to the native ISA at runtime for the platform on which the application runs and do the offload to the processor graphics. The Jitter gets the current processor graphics information at runtime. The new feature allows generation of native ISA at link time by using the new option `/Qgpu-arch:<arch>` for Windows and `-mgpuarch=<arch>` for Linux. The option is described in detail in the User's Guide.

7 System Requirements

7.1 Supported Host Platforms

One of the following Linux distributions (this is the list of distributions supported by all components; other distributions may or may not work and are not recommended - please refer to Technical Support if you have questions).

In most cases Intel® System Studio 2016 will install and work on a standard Linux* OS distribution based on current Linux* kernel versions without problems, even if they are not listed below. You will however receive a warning during installation for Linux* distributions that are not listed

- Red Hat Enterprise* Linux* 6, 7
- Ubuntu* 10.04 LTS, 12.04 LTS, 14.04 LTS, 15.04 LTS
- Fedora* 20
- Wind River* Linux* 5, 6
- openSUSE* 12.1
- SUSE LINUX Enterprise Server* 11 SP2, 12

Individual Intel® System Studio 2016 components may support additional distributions. See the individual component's release notes after you unpacked and ran the installer for the tool suite distribution

```
> tar -zxvf system_studio_2016.1.xxx.tgz
```

for details.

Sudo or Root Access Right Requirements

- Integration of the Intel® C++ Compiler into a Yocto Project* Application Development Toolkit installed to /opt/poky/ requires the launch of the tool suite installation script install.sh as root or sudo user.
- Installation of the hardware drivers for the Intel® ITP-XDP3 probe to be used with the Intel® System Debugger requires the launch of the tool suite installation script install.sh as root or sudo user.

Environment Setup

To setup the environment for the Intel® C++ Compiler and integrate it correctly with the build environment on your Linux host, execute the following command:

```
> source <install-dir>/bin/compilervars.sh <arch> [vs] [-arch <arch>]  
[-platform <platform>]
```

where `<install-dir>` is the top-level Intel® System Studio installation directory, and `<arch>` is one of the following architectures:

`ia32`: Compilers and libraries for IA32 architectures only

`intel64`: Compilers and libraries for Intel® 64 architectures only

`[vs]` is the Microsoft Visual Studio* version and `<platform>` can be either `linux` or `android`.

7.2 Eclipse* Integration Prerequisites

When asked point the installer to the installation directory of your Eclipse* install. Usually this would be `/opt/eclipse/`.

The prerequisites for successful Eclipse integration are:

1. Eclipse* 3.8 (Juno) – Eclipse* 4.5 (Mars)
2. Eclipse* CDT 8.0 – 8.3
3. Java Runtime Environment (JRE) version 6.0 (also called 1.6) update 11 or later.

7.3 Host Prerequisites and Resource Requirements

7.3.1 Host Space Requirements by Component

	Minimum RAM	Recommended RAM	Disk Space
Intel® System Studio	2Gb	4Gb	7Gb
Intel® C++ Compiler	1Gb	2Gb	2.5Gb
Intel® Integrated Performance Primitives	1Gb	4Gb	1-2Gb
Intel® Math Kernel Library	1Gb	4Gb	2.3Gb
Intel® VTune™ Amplifier for Systems	2Gb	4Gb	650Mb
Intel® Inspector for Systems	2Gb	4Gb	350Mb
GDB	1Gb	2Gb	200Mb
Intel® System Debugger	1Gb	2Gb	300Mb

7.3.2 Intel® Integrated Performance Primitives (Intel® IPP) Details

Intel® Integrated Performance Primitives (Intel® IPP) for IA-32 Hardware Requirements:

- 1800MB of free hard disk space, plus an additional 400MB during installation for download and temporary files.

Intel® Integrated Performance Primitives (Intel® IPP) for Intel® 64 Hardware Requirements:

- 1900MB of free hard disk space, plus an additional 700MB during installation for download and temporary files.

7.3.3 Intel® C++ Compiler

Cross-build for Wind River Linux* target currently requires an existing Wind River* Linux 4.x, 5.x, 6.x or 7.x installation that the compiler can integrate into.

7.4 Target Software Requirements

- Yocto Project* 1.4, 1.5, 1.6, 1.7, 1.8 based environment
- CE Linux* PR35 based environment
- Tizen* IVI 3.x
- Wind River* Linux* 4, 5, 6, 7 based environment
- Android* 4.1.x through 5.1

Note:

The level of target OS support by a specific Intel® System Studio component may vary.

7.5 Target Prerequisites and Resource Requirements

7.5.1 Target Space Requirement by Component

	Minimum RAM	Dependencies	Disk Space
Intel® C++ Compiler	application dependent	Linux kernel 1.26.18 or newer glibs-2.5 or compatible libgcc-4.1.2 or compatible libstdc++-3.4.7 or compatible	13Mb (IA-32) 15Mb (Intel® 64)
Intel® VTune™ Amplifier CLI (Command-Line Interface)	4Gb	Specific kernel configuration reqs. Details below.	200Mb
Intel® VTune™ Amplifier SEP (Sampling Enabling Product)	(# logical cores+2) Mb	specific kernel configuration reqs. Details below.	8Mb
SoC Watch	(# logical cores+2) Mb	Specific kernel configuration reqs. See SoCWatch documentation	8Mb
WakeUp Watch	(# logical cores+2) Mb	Specific kernel configuration reqs. See WuWatch documentation	8Mb

Intel® Inspector for Systems CLI	2Gb	4Gb	350Mb
gdbserver	negligable	none	1.5Mb
xdbntf.ko	<1Mb	kernel build environment	<1Mb

7.5.2 Intel® VTune™ Amplifier target OS kernel configuration

For Intel® VTune™ Amplifier performance analysis and Intel® Energy Profiler there are minimum kernel configuration requirements. The settings below are required for different analysis features.

- For event-based sampling (EBS) sep3_x.ko and pax.ko require the following settings:
CONFIG_PROFILING=y
CONFIG_OPROFILE=m (or CONFIG_OPROFILE=y)
CONFIG_HAVE_OPROFILE=y
- For EBS with callstack information vtsspp.ko additionally needs the following settings:
CONFIG_MODULES=y
CONFIG_SMP=y
CONFIG_MODULE_UNLOAD=y
CONFIG_KPROBES=y
CONFIG_TRACEPOINTS=y (optional but recommended)
- For power analysis, required by apwr3_x.ko
CONFIG_MODULES=y
CONFIG_MODULE_UNLOAD=y
CONFIG_TRACEPOINTS=y
CONFIG_FRAME_POINTER=y
CONFIG_COMPAT=y
CONFIG_TIMER_STATS=y
CONFIG_X86_ACPI_CPUFREQ=m (or CONFIG_X86_ACPI_CPUFREQ=y)
CONFIG_INTEL_IDLE=y

7.5.3 Intel® VTune™ Amplifier Feature vs. Resource Matrix

	Event based sampling (EBS) analysis	EBS analysis with stacks	Algorithmic analysis (PIN-based)	Intel Energy Profiler	Remote collection from host	Result view on target	Requirements:
SEP "VTune Amplifier hardware event-based sampling collector for performance analysis"	X						~8 MB disk space (Number of logical cores +2) Mb RAM
amplxe-cl -target "VTune Amplifier collector for power and performance analysis on Embedded Linux systems"	X	X	X	X	X		~25 MB disk space ~64 Mb RAM
amplxe-cl "VTune Amplifier command line interface for text-based power and performance analysis"	X	X	X	X	X	X	~200MB disk space >= 4Gb RAM

7.6 Hardware Requirements

- IA32 or Intel® 64 architecture based host computer
- Development platform based on the Intel® Atom™ processor Z5xx, N4xx, N5xx, D5xx, E6xx, N2xxx, D2xxx, E3xxx, Z2xxx, Z3xxx, C2xxx, or Intel® Atom™ processor CE4xxx, CE53xx and the Intel® Puma™ 6 Media Gateway
- Intel® Edison development platform
- Alternatively development platform based on 2nd, 3rd, 4th, 5th or 6th generation Intel® Core™ processor.
- Xeon® processors based on 2nd, 3rd, 4th or 5th generation Intel® Core™ architecture.

7.7 Intel® Graphics Technology development specific requirements

Up-to-date information on hardware, operating system and driver requirements for offloading computations to the integrated processor graphics can be found on the following page:

<https://software.intel.com/en-us/articles/getting-started-with-compute-offload-to-intelr-graphics-technology>

8 Installation Notes

8.1 Installing the Tool Suite

The installation process as well as prerequisites for using the different Intel® System Studio components are documented online and can be found here:

<https://software.intel.com/en-us/articles/system-studio-install-prerequisites>

The default base installation, in the following referred to as <install-dir> directory is:

- For sudo/root installation: /opt/intel/
- For user mode installation: \$HOME/intel/

Important Note: As indicated in the installation process, Intel® System Studio 2015 customers need to upgrade their license either by entering an Intel® System Studio **2016** serial number directly or by obtaining the new license file from Intel® Registration Center. More information on this can be found on the following page:

<https://software.intel.com/en-us/articles/required-license-upgrade-for-intel-system-studio-2016-and-intel-parallel-studio-xe-2016>

8.2 Installation directory structure

Intel® System Studio 2016 installs components which are unique to System Studio into <install-dir>/system_studio_2016.1.xxx/ and components which share subcomponents (such as documentation) with other Intel® Software Development Products into <install-dir>.

The Intel® System Studio 2016 installation directory contains tools and directories as well as links to shared components into the parent directory as follows:

- <install-dir>/system_studio_2016.1.xxx/compiler_libraries -> ../compilers_and_libraries_2016.1.xxx
- <install-dir>/system_studio_2016.1.xxx/debugger
- <install-dir>/system_studio_2016.1.xxx/documentation -> ../documentation_2016
- <install-dir>/system_studio_2016.1.xxx/FrameAnalyzerOGL -> ../FrameAnalyzerOGL
- <install-dir>/system_studio_2016.1.xxx/ide_support -> ../ide_support_2016
- <install-dir>/system_studio_2016.1.xxx/inspector_for_systems -> ../inspector_2016_for_systems
- <install-dir>/system_studio_2016.1.xxx/licensing
- <install-dir>/system_studio_2016.1.xxx/man -> ../man
- <install-dir>/system_studio_2016.1.xxx/PlatformAnalyzer -> ../PlatformAnalyzer
- <install-dir>/system_studio_2016.1.xxx/samples -> ../samples_2016
- <install-dir>/system_studio_2016.1.xxx/SystemAnalyzer -> ../SystemAnalyzer
- <install-dir>/system_studio_2016.1.xxx/targets

- `<install-dir>/system_studio_2016.1.xxx/uninstall`
- `<install-dir>/system_studio_2016.1.xxx/uninstall_GUI.sh`
- `<install-dir>/system_studio_2016.1.xxx/uninstall.sh`
- `<install-dir>/system_studio_2016.1.xxx/vtune_amplifier_for_systems -> ..vtune_amplifier_2016_for_systems`
- `<install-dir>/system_studio_2016.1.xxx/wr-iss-2016`

for Intel® C++ Compiler, Intel® Integrated Performance Primitives, Intel® Math Kernel Library, Intel® Threading Building Blocks, Intel® System Debugger, Intel® System Studio System Analyzer, Intel® VTune™ Amplifier, Intel® Inspector and the Wind River Linux* development environment integration respectively.

The Intel® System Studio contains components under GNU* General Public License (GPL) in addition to commercially licensed components. This includes the GNU* Project Debugger – GDB and the kernel module used by the Intel® System Debugger to export Linux* dynamically kernel module memory load information to host.

The Intel® VTune™ Amplifier, Intel® Energy Profiler and Intel® Inspector are available for power and performance tuning as well as memory and thread checking on the installation host.

For additional installation of command-line only versions of Intel® VTune™ Amplifier and Intel® Inspector on the development target, please follow the sub-chapter on the command line interface (CLI) installations below.

Furthermore a targets directory contains Intel® C++ Compiler runtime libraries, the Intel® VTune™ Amplifier Sampling Enabling Product (SEP) , target components for the Intel® VTune™ Amplifier Data Collector, the kernel module used by the Intel® System Debugger to export Linux* dynamically kernel module memory load information to host, and prebuilt gdbserver target debug agents for GDB.

Sudo or Root Access Right Requirements

- Integration of the Intel® C++ Compiler into the Yocto Project* Application Development Toolkit requires the launch of the tool suite installation script `install.sh` as root or sudo user.
- Installation of the hardware drivers for the Intel® ITP-XDP3 probe to be used with the Intel® System Debugger requires the launch of the tool suite installation script `install.sh` as root or sudo user.

8.3 Development target package installation

The targets directory contains Intel® C++ Compiler runtime libraries, the Intel® VTune™ Amplifier Sampling Enabling Product (SEP) , target components for the Intel® VTune™ Amplifier Data Collector, target components for the Intel® Inspector, the `xdbnf.ko` used by the

Intel® System Debugger to export Linux* kernel module memory load information to host, and prebuilt gdbserver target debug agents for GDB.

To install it follow the steps below

1. Copy the contents of the `/opt/intel/system_studio_2016.1.xxx/targets` directory to your target platform and unpack the `system_studio_target.tgz` and `debugger_kernel_module.tgz` files contained in this directory there.
2. Add the compiler runtime libraries that you find in `../system_studio_target/compiler/lib/ia32` to your target environment search path.
3. For the dynamic kernel module load export feature follow the instructions found at `../debugger_kernel_module/debugger/xdb/kernel-modules/xdbntf/read.me`.
This is also detailed in the Intel® System Debugger Installation Guide and Release Notes `jtag-release-install.pdf`.
4. For the GDB* Debugger remote debug agent `gdbserver` pick the executable that describes your target system from `..<install-dir>/system_studio_2016.1.xxx/targets/<arch>/<target>/bin`, where `arch` and `target` can have the following values
 - o `arch: ia32, intel64, Quark`
 - o `target: WindRiverLinux5, WindRiverLinux4, TizenIVI, CELinuxPR32, Yocto1.3, Yocto1.4, Yocto1.5, Galileo`Run `gdbserver` on the target platform to enable remote application debug. During the Intel® System Studio product install you can also choose to install the `gdbserver` sources if support for additional target platforms is needed.
5. For the Intel® VTune Amplifier Sampling Enabling Product (SEP) pick `../system_studio_target/vtune_amplifier_2016_for_systems_target/linux<arch>/vtune_amplifier_target_sep_x86[_64].tgz`
 - o `arch: 32, 64`
6. For the Intel® VTune Amplifier for Systems target package pick `../system_studio_target/vtune_amplifier_2016_for_systems_target/linux<arch>/vtune_amplifier_target_x86[_64].tgz`
 - o `arch: 32, 64`
7. For WakeUp Watch for Android* follow the instructions at `../system_studio_target/wuwatch_android/WakeUpWatchForAndroid.pdf`
8. For SoC Watch for Android* follow the instructions at `../system_studio_target/socwatch_android_v1.4d/SoCWatchForAndroid_v1_4_0.pdf`
9. For the Intel® Inspector for Systems follow the instructions in `../system_studio_target/inspector_2016_for_systems/documentation/en/Release_Notes_Inspector_Linux.pdf`

8.3.1 Intel® Inspector Command line interface installation

If you would like to install the Intel® Inspector command line interface only for thread checking and memory checking on a development target device, please follow the steps outlined below:

1. From `../inspector_2016_for_systems/` on the target execute the environment configuration script `inspxe-genvars.sh`.
2. Source the script `inspxe-vars.sh` generated by `inspxe-genvars.sh`.
3. The fully functional command-line Intel® Inspector installation can be found in the `bin32` and `bin64` subdirectories for IA32 and Intel® 64 targets respectively.

8.3.2 Preparing a Target Android* System for Remote Analysis

If you would like to install the Intel® VTune™ Amplifier data collectors for power tuning and performance tuning on an Android* target device, please follow the steps outlined below:

1. You will find SoC Watch at

`../system_studio_target/socwatch_android_v1.4d/`
on the target.

- `../system_studio_target/socwatch_android_v1.4d/WakeUpWatchForAndroid.pdf`
- The “[Preparing a Target Android* System for Remote Analysis](#)” chapter of the Intel® VTune™ Amplifier User’s Guide.

2. You will find WakeUp Watch at

`../system_studio_target/wuwatch_android`

on the target.

Please follow the instructions for installation and usage in

- `../system_studio_target/socwatch_android/SoCWatchForAndroid_v1_4_0.pdf`
- The “[Preparing a Target Android* System for Remote Analysis](#)” chapter of the Intel® VTune™ Amplifier User’s Guide.

8.3.3 Intel® VTune™ Amplifier Collectors Installation on Remote Linux* Systems

If you would like to install the Intel® VTune™ Amplifier data collector for power tuning and performance tuning on a development target device, please follow the steps outlined below:

3. You will find the Intel® VTune™ Amplifier data collectors at

```
../system_studio_target/vtune_amplifier_2016_for_systems_target/linux<arch>/vtune_amplifier_target_x86[_64].tgz  
arch: 32, 64  
on the target.
```

4. Data collection on both IA32 and Intel® 64 targets is supported.
5. Follow the instructions in Help document in section “User’s guide->Running analysis remotely” for more details, on how to use this utility.

8.3.4 Intel® VTune™ Amplifier Sampling Enabling Product Installation on Remote Linux* Systems

If you would like to install the Intel® VTune™ Amplifier Sampling Enabling Product (SEP), please follow the steps outlined below:

1. You will find the Intel® VTune Amplifier Sampling Enabling Product at

```
../system_studio_target/vtune_amplifier_2016_for_systems_target/linux<arch>/vtune_amplifier_target_sep_x86[_64].tgz
```
2. After unpacking this zip file follow the instructions in

```
../vtune_amplifier_2016_for_systems.0.x.xxxxxx/sepdk/src/README.txt
```

8.3.5 Intel® Integrated Performance Primitives redistributable shared object installation

If you are using dynamic linking when using the Intel® Integrated Performance Primitives, you will need to copy the relevant Linux* shared objects to the target device along with the application.

A list of the redistributable shared objects can be found at

```
/opt/intel/system_studio_2016.1.xxx/documentation_2016/en/ipp
```

8.3.6 Intel® Math Kernel Library redistributable shared object installation

If you are using dynamic linking when using the Intel® Math Kernel Libraries, you will need to copy the relevant Linux* shared objects to the target device along with the application.

A list of the redistributable shared objects can be found at

```
/opt/intel/system_studio_2016.1.xxx/documentation_2016/en/mkl
```

8.3.7 Intel® C++ Compiler dynamic runtime library installation

After unpacking system_studio_target.tgz on the target platform you will find the Intel® C++ Compiler runtime libraries at `../system_studio_target/compiler/lib/<arch>`,

where `<arch>` is `ia32` or `intel64`.

8.4 Eclipse* IDE Integration

8.4.1 Installation

The Intel® C++ Compiler for Embedded OS and GDB can be automatically integrated into a preexisting Eclipse* CDT installation. The Eclipse* CDK, Eclipse* JRE and the Eclipse* CDT integrated development environment are not shipped with this package of the Intel® System Studio. The Eclipse* integration is automatically offered as one of the last steps of the installation process if you run `install.sh` or `install_GUI.sh`. If you decide against integration during an earlier install, simply rerun the Intel® System Studio installer.

When asked point the installer to the installation directory of your Eclipse* install. Usually this would be `/opt/eclipse/`.

The prerequisites for successful Eclipse integration are:

1. Eclipse* 3.8 (Juno) – Eclipse* 4.4 (Luna)
2. Eclipse* CDT 8.0 – 8.3
3. Java Runtime Environment (JRE) version 6.0 (also called 1.6) update 11 or later.

Note: The Eclipse* integration of the GDB* GNU Project Debugger requires that the Intel® C++ Compiler installation is selected during Intel® System Studio installation as well.

8.4.2 Launching Eclipse for Development with the Intel C++ Compiler

Since Eclipse requires a JRE to execute, you must ensure that an appropriate JRE is available to Eclipse prior to its invocation. You can set the `PATH` environment variable to the full path of the folder of the `java` file from the JRE installed on your system or reference the full path of the `java` executable from the JRE installed on your system in the `-vm` parameter of the Eclipse command, e.g.:

```
eclipse -vm /JRE folder/bin/java
```

Invoke the Eclipse executable directly from the directory where it has been installed. For example:

```
<eclipse-install-dir>/eclipse/eclipse
```

8.4.3 Editing Compiler Cross-Build Environment Files

Environment File Support appears under “Intel® System Studio - Intel® System Studio Tools Environment File” on the menu bar.

For details on the Environment File Editor, please check the Intel® System Studio User’s Guide at

```
<install-dir>/documentation_2016/en/iss2016/iss_ug.pdf
```

8.4.4 Cheat Sheets

The Intel® C++ Compiler Eclipse* Integration additionally provides Eclipse* style cheat sheets on how to set up a project for embedded use cases using the Intel® C++ Compiler. In the Eclipse* IDE see

Help > Cheat Sheets > Intel® C/C++ Compiler

8.4.5 Integrating the Intel® System Debugger into Eclipse*

Remote debugging with GDB using the Eclipse* IDE requires installation of the C/C++ Development Toolkit (CDT) (<http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/junosr2>) as well as Remote System Explorer (RSE) plugins (<http://download.eclipse.org/tm/downloads>). In addition RSE has to be configured from within Eclipse* to establish connection with the target hardware.

To add Intel® System Debugger Eclipse* integration after full Intel® System Studio installation or to add the Intel® System Debugger launcher into Wind River* Workbench* this can be done from within Eclipse* by following these steps:

1. Navigate to the "Help > Install New Software" entry in the pulldown menu
2. Select "Add" and "Local" in the following menus ...
3. Browse to `<ISS_INSTALL_PATH>/debugger/xdb/ide_plugins`, where the default for `ISS_INSTALL_PATH` is `/opt/intel/system_studio_2016.1.xxx/`

8.5 Wind River* Workbench* IDE Integration

8.5.1 Documentation

1. You will find a detailed README file on the integration particulars of Intel® System Studio in the `wr-iss-2016` subdirectory of the Wind River* Workbench* installation directory. This README also goes into the use of the Intel® C++ Compiler as a secondary toolchain layer and adding Intel® System Studio recipes to target platforms for both Wind River* Linux* and Yocto Project*.
2. Additionally there is a Wind River* Workbench integration feature and usage description in the "[Using Intel® System Studio with Wind River* Linux* Build Environment](#)" article.

8.5.2 Installation

Intel® System Studio provides Wind River* Linux* build environment integration and platform recipes for Intel® C++ Compiler, Intel® Performance Primitives, Intel® Math Kernel Library and Intel® VTune™ Amplifier Sampling Collector.

It also integrated IDE launchers for Intel® VTune™ Amplifier for Systems and Intel® System Debugger.

This is offered automatically as a step in the Intel® System Studio product installation.

As part of the installation the following steps are taken implicitly:

1. Create folder `wr-iss-2016` in both the Intel® System Studio installation directory and the Wind River* Workbench* installation directory.
2. In the `wr-setup` subdirectory, execute the script `postinst_wr_iss.sh` `<install-dir>`, providing the Intel® System Studio installation directory as a parameter. This script will register the platform recipes for different Intel® System Studio components and also the IDE integration of Intel® System components such as Intel® C++ Compiler, Intel® VTune™ Amplifier and Intel® System Debugger.

8.5.3 Manual installation

1. Change into the Wind River* Workbench* installation directory and there into the `../wr-iss-2016/wr-setup` subdirectory.
2. In the `wr-setup` subdirectory, execute the script `postinst_wr_iss.sh`. This script will register the platform recipes for different Intel® System Studio components and also the IDE integration of Intel® System components such as Intel® C++ Compiler, Intel® VTune™ Amplifier and Intel® System Debugger.

8.5.4 Uninstall

3. Change into the Wind River* Workbench* installation directory and there into the `../wr-iss-2016/wr-setup` subdirectory.
4. In the `wr-setup` subdirectory, execute the script `uninst_wr_iss.sh`

8.6 Installing Intel® XDP3 JTAG Probe

If the `install.sh` installation script is executed using root access, `su` or `sudo` rights, the required drivers will be installed automatically. Root, `su` or `sudo` rights are required for the installation.

Ordering JTAG Device for Intel® System Debugger

To order the Intel® ITP-XDP3 device, please

1. either log into your account at <https://designintools.intel.com/>, select the Debug Tools product category and add ITP-XDP BR3 to your cart.
2. or contact the Hibbert Group* at Intelvtg@hibbertgroup.com and request the VTG order form.

We will also gladly assist with the ordering process. If you have any questions please submit an issue in the Intel® System Studio product of Intel® Premier Support <https://premier.intel.com> or send an email to IntelSystemStudio@intel.com.

8.7 Removing the Product

To uninstall the Intel® System Studio, change to the `/opt/intel/system_studio_2016.1.xxx` directory and run the `uninstall.sh` uninstall script.

9 Issues and Limitations

9.1 General Issues and Limitations

For known issues of individual Intel® System Studio components please refer to the individual component release notes. Their location in the installed product can be found in chapter 2:

[Technical Support and Documentation](#)

9.1.1 Use non-RPM installation mode with Wind River* Linux* 5 and 6

RPM package access on Wind River* Linux* 5 may be slow and cause the Intel® System Studio installation to take a long time. On Wind River* Linux* 5 host it is recommended to invoke the installation script in non-RPM mode instead

```
$ ./install.sh --INSTALL_MODE NONRPM
```

or

```
$ ./install-GUI.sh --INSTALL_MODE NONRPM
```

9.1.2 Intel® Software Manager unsupported on Wind River* Linux* 5.0, Ubuntu* 12.04.

The Intel® Software Manager is not supported on Wind River* Linux* 5.0 and Ubuntu* 12.04.

9.1.3 Installation into non-default directory on Fedora* 19 may lead to failures

If Intel® System Studio is installed into a folder other than /opt/intel/system_studio_2016.1.xxx/ on Fedora 19, the installation of optional components and of the Python* source for GDB* may fail.

The workaround is to only install default components and deselect the Python* source component of GDB* during install.

Please refer to Bugzilla* report 1001553 entered against RPM tool 4.11 for details.

https://bugzilla.redhat.com/show_bug.cgi?id=1001553

9.1.4 Running online-installer behind proxy server fails

Running online-installer behind proxy server produces error: "Connection to the IRC site cannot be established". Please see the [Installation Notes](#) for more details

9.1.5 The online-installer has to be run with sudo or root access

If the online-installer is run as a regular user the installation will hang in step 6 of the installation. Please see the [Installation Notes](#) for more details.

9.1.6 Some hyperlinks in HTML documents may not work when you use Internet Explorer.

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

9.2 Wind River* Linux* 7 Support

9.2.1 Windows* host is currently not supported for Wind River* Linux* 7 targeted development

Windows* host is currently not supported for Wind River* Linux* 7 targeted development

9.2.2 No integration into Wind River* Workbench* IDE is currently available for Wind River* Linux* 7 target

No integration into Wind River* Workbench* IDE is currently available for Wind River* Linux* 7 target

9.2.3 Remote event-based sampling with Intel® VTune™ Amplifier Limitations

When targeting Wind River* Linux* 7 with the Sampling Collector (SEP) for Intel® VTune™ Amplifier for Systems sampling with callstack using vtss.ko can lead to target freeze-up. The recommendation is to not use callstacks sampling when targeting Wind River* Linux* 7 until the next release update.

9.3 Intel® Energy Profiler

9.3.1 /boot/config-`uname -r` file must be present on platform.

In order to enable CPU power data collection for Intel® VTune™ Amplifier please make sure your environment does have a file named `/boot/config-`uname -r`` located in your `/boot/config` directory

If there is no such file you should run the following command:

```
$ cat /proc/config.gz | gunzip - > /boot/config-`uname -r`
```

9.3.2 Power and Frequency Analysis support for Intel® Atom™ Processor covers Android* OS only.

Power and frequency analysis currently requires at least a 2nd generation Intel® Core™ Processor Family based platform or an Intel® Atom™ Processor Z2xxx or Z3xxx running Android* OS

9.4 Intel® VTune™ Amplifier Usage with Yocto Project*

9.4.1 Building Sampling Collector (SEP) for Intel® VTune™ Amplifier driver on host Linux* system

For Yocto Project* targeted development additional kernel utilities required for building drivers and kernel modules need to be present in the kernel source tree. The following utilities need to be manually added to the standard Yocto Project* 1.x kernel build tree: viz, recordmcount, fixdep, and modpost.

9.4.2 Remote Intel® VTune™ Amplifier Sampling on Intel® 64 Yocto Project* Builds

The GNU linker ld is installed in a non-standard path on Yocto Project* 1.5 for Intel® 64 (x86_64). For remote sampling with amplxe-runss to work correctly "/lib64/ld-linux-x86-64.so.2 " has to be added as a symlink to /lib/ld-linux-x86-64.so.2 on the target filesystem.

9.4.3 Building 64bit Sampling Collector against Yocto Project* targeting Intel® Atom™ Processor E38xx requires additional build flags

Building the Intel® VTune™ Amplifier for Systems Sampling Collector driver SEPDK against the x86_64 version of Yocto Project 1.6 (Daisy) for Intel® Atom™ Processor E38xx requires a modification of the Makefile in ../sepdk/src and ../sepdk/pax.

In both cases the EXTRA_CFLAGS entry needs to be amended with the option -DCONFIG_COMPAT:

```
EXTRA_CFLAGS += -I$(LDDINCDIR) -I$(LDDINCDIR1) -DCONFIG_COMPAT
```

9.5 Intel® System Studio System Analyzer

9.5.1 Supported Linux* Distributions

The Intel® System Studio System Analyzer is currently only supported on Ubuntu.

9.5.2 The path for the Intel® System Studio System Analyzer does not get set up automatically

To launch the Intel® System Studio System Analyzer it is necessary to change into the /opt/intel/usr/bin directory and run the scripts or add this directory to the PATH.

The run scripts for the System Analyzer binaries are located at:

- System Analyzer: /opt/intel/usr/bin/gpa-system-analyzer
- GPA console client: /opt/intel/usr/bin/gpa-console-client
- Platform Analyzer: /opt/intel/usr/bin/gpa-platform-analyzer
- Frame Analyzer for OpenGL: /opt/intel/FrameAnalyzerOGL/FrameAnalyzerOGL.sh

9.5.3 Support for Intel® Atom™ Processor Z3560 and Z3580 code-named “Moorefield” missing

Support for Intel® Atom™ Processor Z3560 and Z3580 code-named “Moorefield” is currently not available

9.6 Intel® System Debugger

9.6.1 Intel® Puma™ 6 Media Gateway Firmware Recovery Tool not available

The start_xdb_firmware_recovery.sh / start_xdb_firmware_recovery.bat utility to allow recovery of corrupted flash memory on the Intel® Puma 6 Media Gateway is not functional in the Intel® System Debugger 2016.

9.6.2 Connecting to Intel® Quark™ SoC may trigger error message that can be ignored

Establishing a connection with the Intel® System Debugger to an Intel® Quark™ soC based platform using the ITP-XDP3 device will trigger a console message “MasterFrame.HostApplication Application Error”. The connection will be established but target is not stopped. To stop target execution press "Suspend Execution (Pause)".

9.6.3 Using the symbol browser on large data sets and large symbol info files not recommended

It is recommended to use the source files window to browse to the function to debug instead of the symbol browser as the use of the symbol browser on large data sets and large symbol information files (e.g. Android* kernel image) can lead to debugger stall.

9.6.4 Limited support for Dwarf Version 4 symbol information

If when debugging binaries generated with GNU* GCC 4.8 or newer the line information and variable resolution in the debugger is unsatisfactory, please try to rebuild your project using the `-gdwarf-3` option instead of simply `-g`.

9.7 GDB* - GNU* Project Debugger

9.7.1 Eclipse* integration of GDB* requires Intel® C++ Compiler install

The Eclipse* integration of the GDB* GNU Project Debugger requires that the Intel® C++ Compiler installation is selected during Intel® System Studio installation as well.

9.8 Intel® C++ Compiler

9.8.1 “libgcc_s.so.1” should be installed on the target system

By default the Intel® C++ Compiler links the compiled binary with the library “libgcc_s.so.1”. Some embedded device OSs, for example Yocto-1.7, don't have it in default

10 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.

11 Disclaimer and Legal Information

The Intel® C++ Compiler, Intel® Integrated Performance Primitives, Intel® Math Kernel Library, Intel® VTune™ Amplifier, and Intel® Inspector for Systems are provided under Intel's End User License Agreement (EULA).

The GNU* Project Debugger, GDB is provided under the General GNU General Public License GPL V3.

The Intel® System Debugger kernel module xdbntf.ko is provided under the General GNU General Public License GPL V2.

Please consult the licenses included in the distribution for details.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: <http://www.intel.com/design/literature.htm>

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. Go to: http://www.intel.com/products/processor_number/

MPEG-1, MPEG-2, MPEG-4, H.261, H.263, H.264, MP3, DV, VC-1, MJPEG, AC3, AAC, G.711, G.722, G.722.1, G.722.2, AMRWB, Extended AMRWB (AMRWB+), G.167, G.168, G.169, G.723.1, G.726, G.728, G.729, G.729.1, GSM AMR, GSM FR are international standards promoted by ISO, IEC, ITU, ETSI, 3GPP and other organizations. Implementations of these standards, or the standard enabled platforms may require licenses from various entities, including Intel Corporation.

BunnyPeople, Puma, Celeron, Celeron Inside, Centrino, Centrino Inside, Cilk, Core Inside, i960, Intel, the Intel logo, Intel AppUp, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside, Intel Inside logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel Sponsors of Tomorrow., the Intel Sponsors of Tomorrow. logo, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, InTru, the InTru logo, InTru soundmark, Itanium, Itanium Inside, MCS, MMX, Moblin, Pentium, Pentium Inside, skool, the skool logo, Sound Mark, The Journey Inside, vPro Inside, VTune, Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and other countries.

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

Microsoft, Windows, Visual Studio, Visual C++, and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Java is a registered trademark of Oracle and/or its affiliates.

Copyright (C) 2008–2015, Intel Corporation. All rights reserved.