Intel® System Studio 2017 Pre-Release (Beta) Installation Guide and Release Notes

Installation Guide and Release Notes for Linux* Host
30 March 2016

Contents

1 Introduction

This document provides a brief overview of the Intel® System Studio 2017 Beta 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

This section highlights important changes in the actual product release. More detailed information about new features and changes in the respective product release notes (s. also section ‘6.1 Release Notes, Installation Notes and User Guides Locations’).

Intel® System Studio 2017 Beta

This section highlights important changes since the latest Intel® System Studio 2016 Update 2 release.

1. **Intel® C++ Compiler:**
   - Annotated source listing
     - This feature annotates source files with compiler optimization reports. The listing format may be specified as either text or html.
   - New attribute, pragma, and compiler options for code alignment
   - Additional C++14 features supported
   - Additional C11 features supported
   - New and Changed Compiler Options

   View the full release notes for more details.

2. **Intel® Math Kernel Library (Intel® MKL):**
   - Introducing Deep Neural Networks (DNN) primitives including convolution, normalization, activation and pooling functions intended to accelerate convolutional neural networks (CNNs) and Deep neural networks (DNNs) on Intel® Architecture
   - Enabled Automatic Offload (AO) and Compiler Assisted Offload (CAO) modes for the second generation of Intel® Xeon Phi™ coprocessor on Linux* OS
   - The SP2DP interface library is removed
   - Removed pre-compiled BLACS library for MPICH v1; MPICH users can still build the BLACS library with MPICH support via Intel MKL MPI wrappers
   - Sparse BLAS:
     - Improved performance of parallel BSRMV functionality for processor supporting Intel® Advanced Vector Extensions 2 (Intel® AVX2) instruction set
   - Intel MKL PARDISO:
     - Added support for mkl_progress in Parallel Direct Sparse Solver for Clusters
   - DFT:
     - Improved performance of batched 1D FFT with large batch size on processor supporting Intel® Advanced Vector Extensions(Intel® AVX), Intel AVX2, Intel® Advanced Vector Extensions 512 (Intel® AVX512) and IntelAVX512_MIC instruction sets
   - Data Fitting:
     - Introduced 2 new storage formats for interpolation results (DF_MATRIX_STORAGE_SITES_FUNCS_DERS, DF_MATRIX_STORAGE_SITES_DERS_FUNCS)
   - **Known Issue:**

    - Intel® Xeon Phi™ coprocessor support is not a default install option in Intel® Parallel Studio XE 2017 Beta Cluster Edition for Linux*. Users will have to select the checkbox during Installation in the "Select components to install" window. This option will be made default option again in coming release

3. **Intel® Performance Primitives (Intel® IPP):**
    - Added new APIs (Intel® IPP 64x functions) to support 64-bit data length in the image and signal processing domains:
        - This release provides the 64x functions for memory allocation, image addition, subtraction, multiplication, division, resizing, and filtering operations.
        - The Intel® IPP 64x functions are implemented as wrappers over Intel® IPP functions operating on 32-bit sizes by using tiling and multithreading. The 64x APIs support external threading for Intel® IPP functions, and are provided in the form of source and pre-built binaries.
    - Added integration wrappers for some image processing and computer vision functions. The wrappers provide the easy-to-use C and C++ APIs for Intel® IPP functions, and they are available as a separate download in the form of source and pre-built binaries.
    - Performance and Optimization:
        - Extended optimization for Intel® Advanced Vector Extensions 512 (Intel® AVX-512) instruction set on Intel® Many Integrated Core Architectures (Intel® MIC Architectures). Please see the Intel® IPP Functions Optimized for Intel® AVX-512 article for more information.
        - Extended optimization for Intel® AVX-512 instruction set on Intel® Xeon® processors.
        - Extended optimization for Intel® Advanced Vector Extensions 2 (Intel® AVX2) instruction set on the 6th Generation Intel® Core™ processors. Please see the Intel® IPP Functions Optimized for Intel® AVX2 article for more information.
        - Extended optimization for Intel® Streaming SIMD Extensions 4.2 (Intel® SSE4.2) instruction set on Intel® Atom™ processors.
    - Signal Processing:
        - Added the ippsIIRIIR functions that perform zero-phase digital IIR filtering.
        - Added 64-bit data length support to the ippsSortRadixAscend functions.
    - Image Processing:
        - Added the ippiScaleC functions to support image data scaling and shifting for different data types.
    - Data Compression:
        - Added the patch files for the zlib compression and decompression functions. The patches provide drop-in optimization with Intel® IPP functions, and support zlib version 1.2.5.3, 1.2.6.1, 1.2.7.3 and 1.2.8.
    - Removed the tutorial from the installation package, and its sample code and documentation are now provided online (https://software.intel.com/en-us/product-code-samples).
        - Threading Notes: Though Intel® IPP threaded libraries are not installed by default, these threaded libraries are available by the custom installation, so the code written with these libraries will still work as before. However, the multi-threaded libraries are deprecated and moving to external threading is recommended. Your feedback on this is welcome

4. **Intel® Threading Building Blocks:**
   - Modified parallel_sort to not require a default constructor for values and to use iter_swap() for value swapping.
   - Added support for creating or initializing a task_arena instance that is connected to the arena currently used by the thread.

     Preview Features:
   - Added template class opencl_node to the flow graph API. It allows a flow graph to offload computations to OpenCL* devices.
   - Extended join_node to use type-specified message keys. It simplifies the API of the node by obtaining message keys via functions associated with the message type (instead of node ports).
   - Added static_partitioner that minimizes overhead of parallel_for and parallel_reduce for well-balanced workloads.
   - Improved template class async_node in the flow graph API to support user settable concurrency limits.
   - Class global_control supports the value of 1 for max_allowed_parallelism.
   - Please see complete list of changes for TBB 4.4 U2 and TBB 4.4 U3 for reference and details.

5. **Intel® System Debugger:**
   - Support for Eclipse* 4.5 (Mars) for the trace viewer. The package is also included in the Intel® System Studio installation package for optional installation.
   - Support for debug format Dwarf4
   - SMM support for Intel® Core™ based pocessors debugging.
   - Improved UEFI debug script

6. **Intel® VTune™ Amplifier for Systems**
   - **Disk Input and Output** analysis used to monitor utilization of the disk subsystem, CPU and processor buses and to identify long latency of I/O requests and imbalance between I/O and compute operations
   - **GPU Hotspots** analysis targeted for GPU-bound applications and providing options to analyze execution of OpenCL™ kernels and Intel Media SDK tasks
   - **Basic Hotspots** analysis extended to **support Python* applications** running via the Launch Application or Attach to Process modes
   - **Application Performance Snapshot** tool (part of Intel Performance Snapshot tool set) providing a quick look at your application performance and helping you understand whether your application will benefit from tuning. It identifies how effectively your application uses the hardware platform and displays basic performance enhancement opportunities.
   - Detection of the **OpenCL™ 2.0 Shared Virtual Memory (SVM) usage types** per kernel instance
   - Driverless event-based sampling collection for uncore events enabled for Memory Access analysis.

7. **Intel® Inspector**

- Variable name detection for threading analysis (global, static and stack variables)
- Bug fixes

8. **Intel® Graphics Performance Analyzer (Intel® GPA)**
   - **New Features for Analyzing Microsoft DirectX* Applications**
     Intel GPA now provides alpha-level support for DirectX* 12 application profiling. This version has limited profiling and debug capabilities and might work unstable on some workloads. You can find more details regarding the supported features below.
     - Graphics Frame Analyzer provides detailed GPU hardware metrics for Intel® graphics. For third-party GPUs, GPU Duration and graphics pipeline statistics metrics are available.
     - DirectX states, Geometry, Shader code, Static and dynamic textures, Render targets resources are available for frame-based analysis in Graphics Frame Analyzer.
     - Simple Pixel Shader, Disable Erg(s) performance experiments, Highlighting and Disable draw calls visual experiments are available in Graphics Frame Analyzer
     - Time-based GPU metrics for Intel graphics, CPU metrics, Media and Power metrics in System Analyzer.
     - System Analyzer HUD includes support for hotkeys, the same set of metrics as in System Analyzer, messages and settings.

     Note: In order to capture DirectX 12 application frames, enable the **Force DirectX12 injection** option in the Graphics Monitor **Preferences** dialog box.
     Note: System memory consumption is expected to be high in this release at both time of capture and during playback. Needed memory is related to workload and frame complexity and varies greatly. 8GB is minimum, 16GB is recommended, with some workloads requiring more.
   - **New Features for Analyzing OpenGL/OpenGL ES* Applications**
     - Enabled support for GPU hardware metrics in System Analyzer and Graphics Frame Analyzer on the 6th Generation Intel® Core™ Processors for Ubuntu* targets.
     - Several OpenGL API calls (e.g. glTexImage2D, glReadPixels, glCopyTexImage2D, etc.) are now represented as ergs in Graphics Frame Analyzer, which allows measuring GPU metrics for them and see the used input and output.
     - Resource History was implemented in Graphics Frame Analyzer. When you select a particular texture or program in the Resource viewer, colored markers appear in the bar chart, indicating the ergs where these resources are used. The color of these markers corresponds to the type of the resource: input, execution, or output.

     View the full release notes for more details.

9. **New Usability Features**
   - **Online Installer**
     - Download for later installation on the same or another computer is now available.
     - The online installer is a full installer agent now including install scripts and first-use documentation.
   - **Eclipse* IDE**
     - All Intel® System Studio start-up documentation (getting started guides, tutorials, samples) is available now from the Eclipse* Welcome window.

  - The Intel® System Debugger, Intel® VTune™ for Systems and Intel® Inspector (which have their own standalone Eclipse framework) can be started directly from within the Eclipse* Welcome window.
- **Eclipse* package provided with Intel® System Studio**
  - The target installation directory for the System Studio provided Eclipse package is now version specific, starting with Eclipse* Mars. The installation directory (if you choose to install the package in the installation dialog) will be: `<INSTALLDIR>/eclipse_<version_codename>`, for example `/opt/intel/eclipse_mars/`

2.1 Version History

This section highlights important changes in previous Intel® System Studio 2016 product versions.

Intel® System Studio 2016 Update 2

1. Intel® C++ Compiler:
   - Support for Microsoft Visual Studio* 2015 Update 1
   - Intrinsics for the Short Vector Random Number Generator (SVRNG) Library
     - The Short Vector Random Number Generator (SVRNG) library provides intrinsics for the IA-32 and Intel® 64 architectures running on supported operating systems. The SVRNG library partially covers both standard C++ and the random number generation functionality of the Intel® Math Kernel Library (Intel® MKL). Complete documentation may be found in the Intel® C++ Compiler 16.0 User and Reference Guide.
   - Intel® SIMD Data Layout Templates (Intel® SDLT)
     - Intel® SDLT is a library that helps you leverage SIMD hardware and compilers without having to be a SIMD vectorization expert.
     - Intel® SDLT can be used with any compiler supporting ISO C++11, Intel® Cilk™ Plus SIMD extensions, and #pragma ivdep
     - Intel® SIMD Data Layout Templates
   - New C++14 and C11 features supported
   - And many others ... For a full list of new features please refer to the Composer Edition product release notes

2. Intel® Math Kernel Library (Intel® MKL)
   - Introduced mkl_finalize function to facilitate usage models when Intel MKL dynamic libraries or third party dynamic libraries are linked with Intel MKL statically are loaded and unloaded explicitly
   - Introduced sorting algorithm
   - Performance improvements for BLAS, LAPACK, ScaLAPACK, Sparse BLAS
   - Several new features for Intel MKL PARDISO
   - Added Intel® TBB threading support for all and OpenMP* for some BLAS level-1 functions.

3. Intel® Performance Primitives (Intel® IPP)

- Image Processing:
    - Added the contiguous volume format (C1V) support to the following 3D data processing functions: ipprWarpAffine, ipprRemap, and ipprFilter.
    - Added the ippiFilterBorderSetMode function to support high accuracy rounding mode in ippiFilterBorder.
    - Added the ippiCopyMirrorBorder function for copying the image values by adding the mirror border pixels.
    - Added mirror border support to the following filtering functions: ippiFilterBilateral, ippiFilterBoxBorder, ippiFilterBorder, ippiFilterSobel, and ippiFilterScharr.
    - Kernel coefficients in the ippiFilterBorder image filtering functions are used in direct order, which is different from the ippiFilter functions in the previous releases.
- Computer Vision:
    - Added 32-bit floating point input data support to the ippiSegmentWatershed function.
    - Added mirror border support to the following filtering functions: ippiFilterGaussianBorder, ippiFilterLaplacianBorder, ippiMinEigenVal, ippiHarrisCorner, ippiPyramidLayerDown, and ippiPyramidLayerUp.
- Signal Processing:
    - Added the ippsThreshold_LTAbsVal function, which uses the vector absolute value.
    - Added the ippsIIRIIR64f functions to perform zero-phase digital IIR filtering.
- The multi-threaded libraries only depend on the Intel® OpenMP* libraries; their dependencies on the other Intel® Compiler runtime libraries were removed

4. Intel® System Debugger:

- Unified installer now for all components of the Intel® System Debugger (for system debug, system trace and WinDbg* extension)
- Support for Eclipse* 4.4 (Luna) integration with Intel® Trace Viewer
- New ‘Trace Profiles’ feature for System Trace Viewer to configure the destination for streaming mode for:
    - BIOS Reserverd Trace Memory
    - Intel® Trace Hub Memory
    - Streaming to DCI-Closed Chassis Adapter (BSSB CCA)
- Tracing to memory support (Intel® Trace Hub or system DRAM memory) for 6th Gen Intel® Core™ processors (PCH) via Intel® XDP3 JTAG probe.
- Various stability bug fixes in Trace Viewer: Handling of decoder-instance-parameters. Crash on stop capture. Errors resulting from renaming capture files. Fix for persistent page up/down navigation. Decoding linked files containing spaces in path. Sporadic Eclipse error when switching target
- Trace Viewer improvements: Event distribution viewer. New progress bar when stopping a trace to memory. Rules are saved now in Eclipse workspace and restored during Eclipse restart. Improved memory download with wrapping enabled.

- Debugging support for Intel® Xeon® Processor D-1500 Product Family on the Grangeville platform.
- System Debugger improvements: Export memory window to text file.

5. Intel® Graphics Performance Analyzer (Intel® GPA)
   - Added support for 32-bit and 64-bit applications on Android M (6.0, Marshmallow).
   - Added support for OS X 10.11 El Capitan.
   - Implemented texture storage parameters modification experiment - you can now change dimensions and sample count parameters for input textures without recompiling your app.
   - Can now export textures in KTX/DDS/PNG file formats.
   - And much more….

View the full release notes for more details.

6. Intel® VTune™ Amplifier for Systems
   - Support for the ITT Counters API used to observe user-defined global characteristic counters that are unknown to the VTune Amplifier
   - Support for the Load Module API used to analyze code that is loaded in an alternate location that is not accessible by the VTune Amplifier
   - Option to limit the collected data size by setting a timer to save tracing data only for the specified last seconds of the data collection added for hardware event-based sampling analysis types
   - New Arbitrary Targets group added to create command line configurations to be launched from a different host. This option is especially useful for microarchitecture analysis since it provides easy access to the hardware events available on a platform you choose for configuration.
   - Source/Assembly analysis available for OpenCL™ kernels (with no metrics data)
   - SGX Hotspots analysis support for identifying hotspots inside security enclaves for systems with the Intel Software Guard Extensions (Intel SGX) feature enabled
   - Metric-based navigation between call stack types replacing the former Data of Interest selection
   - Updated filter bar options, including the selection of a filtering metric used to calculate the contribution of the selected program unit (module, thread, and so on)
   - DRAM Bandwidth overtime and histogram data is scaled according to the maximum achievable DRAM bandwidth

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 Intel® Software Manager, automatically installed with the Intel® System Studio product, is a graphical tool available under `<INSTALLDIR>/ism/ism` to provide a simplified delivery mechanism for product updates, current license status and news on all installed Intel software products. The default installation directory `<INSTALLDIR>` is `/opt/intel/` (for sudo/root installation) and `$HOME/intel/` (for user mode installation).

The software manager from this release replaces any previous installed software manager and manages all installed Intel® Software Development Tools licenses on the system.
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 17.0 Pre-Release (Beta) (64-bit host only)
2. Intel® Math Kernel Library 2017 Pre-Release (Beta)
3. Intel® Integrated Performance Primitives 2017 Pre-Release (Beta)
4. Intel® Threading Building Blocks 4.4 Update 3
5. Intel® Debugger for Heterogeneous Compute 2017 (64-bit host only)
   5.1. Intel® Debugger for Heterogeneous Compute 2017 Sources
   5.2. libelfdwarf.so library (Provided under GNU LGPL)
6. Intel® Inspector 2017 Beta
7. Intel® VTune™ Amplifier 2017 Beta for Systems with Intel® Energy Profiler
8. Intel® Graphics Performance Analyzers 2016 R1
9. Intel® System Debugger 2017 Pre-Release (Beta) (64-bit host only)
   9.1. Intel® System Debugger notification module xdbntf.ko (Provided under GNU General Public License v2)
10. OpenOCD 0.8.0 library and source (Provided under GNU GPL V2+)
11. GNU* GDB 7.10 (Provided under GNU General Public License v3) (64-bit host only)
    11.1. Source of GNU* GDB 7.10 (Provided under GNU General Public License v3)
    11.2. Python sources (Provided under Python Software Foundation License)
12. Eclipse* IDE 4.5 (Mars)

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.

The following paths are given relative to the installation directory `<INSTALLDIR>`. The default installation directory is:

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

Intel® System Studio User's Guide

- `<INSTALLDIR>/documentation_2017/en/iss2017/iss_ug.pdf`

Intel® System Studio Getting Started Guide

- `<INSTALLDIR>/documentation_2017/en/iss2017/iss_gsg_lin.htm`

6 Technical Support and Documentation

6.1 Release Notes, Installation Notes and User Guides Locations

Release and installation notes can be found after unpacking the full offline installer `system_studio_2017.x.xxx.tgz` or in the online installer `system_studio_2017.x.xxx_online.tgz` packages.

After installation, all release/installation notes and getting-started guides can be found under the following directories. The paths are given relative to the installation directory `<INSTALLDIR>`. 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

- `<INSTALLDIR>/documentation_2017/en/iss2017/all-release-install.pdf`

Intel® C++ Compiler

- `<INSTALLDIR>/documentation_2017/en/compiler_c/ReleaseNotes_ISS_Compiler.pdf`

Intel® Integrated Performance Primitives

- `<INSTALLDIR>/documentation_2017/en/ipp/iss2017/ReleaseNotes.htm`
- `<INSTALLDIR>/documentation_2017/en/ipp/common/get_started.htm`

Intel® Math Kernel Library

- `<INSTALLDIR>/documentation_2017/en/mkl/common/ReleaseNotes.htm`
- `<INSTALLDIR>/documentation_2017/en/mkl/iss2017/get_started.htm`

Intel® Threading Building Blocks

- `<INSTALLDIR>/documentation_2017/en/tbb/common/Release_Notes.txt`
- `<INSTALLDIR>/documentation_2017/en/tbb/common/get_started.htm`

Intel® System Debugger

- `<INSTALLDIR>/documentation_2017/en/debugger/iss2017/system_debugger/get_started.htm`
- `<INSTALLDIR>/documentation_2017/en/debugger/iss2017/system_debugger/system_debug/sysdebug-release-install.pdf`

GDB

- `<INSTALLDIR> /documentation_2017/en/debugger/iss2017/gdb/GDB_Release_notes.pdf`
- `<INSTALLDIR> /documentation_2017/en/debugger/iss2017/gdb/get_started.htm`
- `<INSTALLDIR>/documentation_2017/en/debugger/gdb-ia/gdb.pdf`
- `<INSTALLDIR>/documentation_2017/en/debugger/gdb-igfx/gdb.pdf`

Intel® VTune™ Amplifier for Systems

- `<INSTALLDIR>/documentation_2017/en/vtune_amplifier_for_systems/release_notes_amplifier_for_systems_linux.pdf`
- `<INSTALLDIR>/documentation_2017/en/vtune_amplifier_for_systems/amplsys_install_guide_linux.pdf`
- `<INSTALLDIR>/documentation_2017/en/vtune_amplifier_for_systems/emon_user_guide.pdf`
- `<INSTALLDIR>/documentation_2017/en/vtune_amplifier_for_systems/socwatch_android_release_notes.pdf`
- `<INSTALLDIR>/documentation_2017/en/vtune_amplifier_for_systems/socwatch_android_users_guide.pdf`
- `<INSTALLDIR>/documentation_2017/en/vtune_amplifier_for_systems/socwatch_linux_release_notes.pdf`
- `<INSTALLDIR>/documentation_2017/en/vtune_amplifier_for_systems/socwatch_linux_users_guide.pdf`
- 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 `<INSTALLDIR>/documentation_2017/en/vtune_amplifier_for_systems/SEP_Users_Guide.pdf`

Intel® Inspector

- `<INSTALLDIR>/documentation_2017/en/inspector/Release_Notes_Inspector_Linux.pdf`

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

- `<INSTALLDIR>/documentation_2017/en/iss2017/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 ISA 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 option `-mgpuarch=<arch>`. 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 2017 Beta 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* 14.04 LTS
- Fedora* 22, 23
- Wind River* Linux* 6, 7, 8 (Native Support)
- OpenSUSE* Leap 42

Individual Intel® System Studio 2017 Beta 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_2017.x.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 <INSTALLDIR>/bin/compilervars.sh ia32|intel64
```

where `<INSTALLDIR>` is the top-level Intel® System Studio installation directory, default:

`/opt/intel` ((sudo)root installation)
`$HOME/intel` (user installation)

7.2 Eclipse* Integration Prerequisites

If you decide to use an existing Eclipse* on the system for integration of System Studio components, point the installer to the installed Eclipse* directory. Usually this would be `/opt/eclipse/`.

The prerequisites for successful Eclipse integration are:

1. Eclipse* 3.8/4.2 (Juno) – Eclipse* 4.5 (Mars)
2. Eclipse* CDT 8.0 – 8.8
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, 2.0 based environment
- CE Linux* PR35 based environment
- Tizen* IVI 3.x
- Wind River* Linux* 6, 7, 8 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.

Note:

- The level of target hardware requirements by a specific Intel® System Studio component may vary.

7.7 Additional requirements for using Intel® C++ Compiler to offload application computation to Intel(R) Graphics Technology

Please see the online Getting Started With Compute Offload To Intel Graphics Technology (https://software.intel.com/en-us/articles/getting-started-with-compute-offload-to-intelr-graphics-technology) for complete host and target requirements.

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 `<INSTALLDIR>` directory is:

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

NOTE: The Intel® System Studio 2017 Beta product supports activation by serial number (requires online internet access) and license file (if computer is not connected to the internet) only.

You have the choice to use the online installer which is a small agent that downloads installation packages only you will choose for installation.

Alternatively you can use the full package offline installer which doesn't require an Internet connection for installation.

8.2 Product Installation (Online Installer)

If you only intend to install specific components of the Intel® System Studio you can reduce the package size that is downloaded for the actual install. Using the online installer requires to be connected to the internet and that https protocol based component downloads are permitted by your firewall.
To run the online installer proceed as follows:

- Unpack the downloaded installer agent package in a directory to which you have write access.
  ```
  > tar -zxvf system_studio_2017.x.xxx_online.tgz
  ```
- Change into the directory the tar file was extracted to
  ```
  cd ./ system_studio_2017.x.xxx_online/
  ```
- Execute one of the installation scripts for command line installation or using the GUI installer.
  ```
  >./install.sh
  >./install_GUI.sh
  ```

following all instructions. During installation you have to activate your product with serial number (xxxx-xxxxxxxx) or using a license file (.lic). See activation details under ch. 8.4.

8.3 Product Installation (Full Package Offline Installer)

Using the full package offline installer is suitable for systems where no Internet connection is available. You can perform a default installation (with a typical selection of components) or a custom installation where you configure your set of components to install.
The full package offline installer is available as a command line tool and a graphical installation Wizard.
To run the installer proceed as follows:

- Unpack the downloaded tool suite package in a directory to which you have write access.
  ```
  > tar -zxvf system_studio_2017.x.xxx.tgz
  ```
- Change into the directory the tar file was extracted to
  ```
  cd ./ system_studio_2017.x.xxx/
  ```
- Execute one of the installation scripts for command line installation or using the GUI installer.
  ```
  >./install.sh
  >./install_GUI.sh
  ```

following all instructions. During installation you have to activate your product with serial number (xxxx-xxxxxxxx) or using a license file (.lic). See activation details under ch. 8.4

8.4 Activating the Product

During installation of the Intel® System Studio 2017 Beta an activation dialog pops up. For Beta yuo have the following options

- **Use existing activation** (this option is visible when the product installer recognized an existing valid license on the system)
- **Activation with Serial Number**. ("Online Activation", requires Internet connection; the format of the serial number is: xxxx-xxxxxxxx)
- **Use license file** (license file .lic must be available on the install machine, no internet connection required)

8.5 Default / Customized Installation

When the Installation Summary dialog pops up, just click the 'Next' for a default installation or on 'Customize' button to modify the list of components to install.

8.6 Uninstalling the Tool Suite

To uninstall the product, execute the following

- Change into the System Studio base directory
  `cd <INSTALLDIR>/system_studio_2017.x.xxx/`

Execute one of the uninstallation scripts on command line or using the GUI uninstaller.

```
./uninstall.sh
./uninstall_GUI.sh
```

You need to uninstall the product with the same rights (user, (sudo) root) as you used for product installation

8.7 Installation directory structure

Intel® System Studio 2017 Beta installs components which are unique to System Studio into `<INSTALLDIR>/system_studio_2017.x.xxx/` and components which share subcomponents (such as documentation) with other Intel® Software Development Products into `<INSTALLDIR>`.

The Intel® System Studio 2017 Beta installation directory contains tools and directories as well as links to shared components into the parent directory as follows:

- `<INSTALLDIR>/system_studio_2017.x.xxx/compiler_libraries -> ../compilers_and_libraries_2017.x.xxx`
- `<INSTALLDIR>/system_studio_2017.x.xxx/debugger`
- `<INSTALLDIR>/system_studio_2017.x.xxx/debugger_2017 -> ../debugger_2017`
- `<INSTALLDIR>/system_studio_2017.x.xxx/documentation_2017 -> ../ documentation_2017`
- `<INSTALLDIR>/system_studio_2017.x.xxx/gpa -> ../gpa`
- `<INSTALLDIR>/system_studio_2017.x.xxx/ide_support_2017 -> ../ide_support_2017`
- `<INSTALLDIR>/system_studio_2017.x.xxx/inspector_2017 -> ../inspector_2017`
- `<INSTALLDIR>/system_studio_2017.x.xxx/licensing`
- `<INSTALLDIR>/system_studio_2017.x.xxx/man -> ../man`
- `<INSTALLDIR>/system_studio_2017.x.xxx/samples_2017 -> ../samples_2017`
- `<INSTALLDIR>/system_studio_2017.x.xxx/system_debugger_2017 -> ../system_debugger_2017`
- `<INSTALLDIR>/system_studio_2017.x.xxx/targets`
- `<INSTALLDIR>/system_studio_2017.x.xxx/uninstall`
- `<INSTALLDIR>/system_studio_2017.x.xxx/vtune_amplifier_2017_for_systems -> ../vtune_amplifier_2017_for_systems`
- `<INSTALLDIR>/system_studio_2017.x.xxx/vtune_amplifier_for_systems -> ../vtune_amplifier_2017_for_systems`
- `<INSTALLDIR>/system_studio_2017.x.xxx/uninstall_GUI.sh`
- `<INSTALLDIR>/system_studio_2017.x.xxx/uninstall.sh`
- `<INSTALLDIR>/system_studio_2017.x.xxx/wr-iss-2017`

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.

8.8 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 xdbntf.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 `<INSTALLDIR>\system_studio_2017.x.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. Add the proper compiler runtime libraries directory (IA32 or Intel64®, Android*, Linux* or QNX*) that you

find in the path
`../system_studio_target/compiler_and_libraries_2017.x.xxx/linux/compiler/lib/`
to your target environment search path. The following target architecture / OS versions are supported:

- `ia32/intel64` for Android* and Linux*
- `ia32` for QNX*
- `ia32/intel64` for Android* and Linux*
- `intel64` for Linux* x32

2. For the dynamic kernel module load export feature follow the instructions found at `../debugger_kernel_module/system_debug/kernel-modules/xdbntf/read.me.`
   This is also detailed in the Intel® System Debugger Installation Guide and Release Notes `sysdebug-release-install.pdf.`
3. For the GDB* Debugger remote debug agent `gdbserver` pick the executable that describes your target system from
   `../system_studio_target/debugger_2017/gdb/targets/<arch>/<target>/bin`.
   The following target architecture / OS versions are supported:
   - `arch: ia32`
     `target: Android, CELinuxPR35, ChromiumOS, KendrickCanyon, TizenIVI, WindRiverLinux4, 5, 6, 7 or 8, Yocto1.4, 1.5, 1.6, 1.7 or 2.0`
   - `arch: intel64`
     `target: Android, ChromiumOS, WindRiverLinux5, 6, 7 or 8, Yocto1.6, 1.7 or 2.0`
   - `arch: Quark`
     `target: Galileo/eglibc, Galileo/uclibc`

   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.
4. For the Intel® VTune Amplifier Sampling Enabling Product (SEP) pick
   `../system_studio_target/vtune_amplifier_2017_beta_for_systems_target/linux/vtune_amplifier_target_sep_x86[_64].tgz`
5. For the Intel® VTune Amplifier for Systems target package pick
   `../system_studio_target/vtune_amplifier_2017_beta_for_systems_target/linux/vtune_amplifier_target_x86[_64].tgz`
6. For SoC Watch follow the instructions at
   `../system_studio_target/socwatch_<OS>_vx.x.x/SoCWatchFor<OS>_vx_x_x.pdf` for the corresponding Android* or Linux* version.
7. For the Intel® Inspector follow the instructions in
   `../system_studio_target/inspector_2017_beta/documentation/en/Release_Notes_Inspector_Linux.pdf`

8.8.1 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 `../system_studio_target/inspector_2017_beta/` 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.

8.8.2 Intel® VTune™ Amplifier Collectors Installation on Remote 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:

1. You will find the Intel® VTune™ Amplifier data collectors at `../system_studio_target/vtune_amplifier_2017_beta_for_systems_target/linux/vtune_amplifier_target_[sep_]x86[_64].tgz` on the target.
2. Data collection on both IA32 and Intel® 64 targets is supported.
3. Follow the instructions in Help document in section “User’s guide->Running analysis remotely” for more details, on how to use this utility.

8.8.3 Preparing an Android* Target 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:

You will find SoC Watch for specific Android* OS versions documentation at `../system_studio_target/socwatch_android_vx.x.x/SoCWatchForAndroid[_vx_x_x].pdf`

8.8.4 Preparing a Linux* Target System for Remote Analysis

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

You will find SoC Watch for specific Linux* OS versions at `../system_studio_target/socwatch_linux_v2.x.x_x86[_64]/SoCWatchForLinux.pdf` on the target.

8.8.5 Intel® VTune™ Amplifier Sampling Enabling Product Installation

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_2017_beta_for_systems_target/linux/vtune_amplifier_target_sep_x86[_64].tgz`

2. After unpacking this zip file follow the instructions in

`../vtune_amplifier_2017_beta_for_systems.x.x.xxxxxx/sepdk/src/README.txt`

8.8.6 Intel® Integrated Performance Primitives runtime 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 for the respective target platform from `<INSTALLDIR>\system_studio_2017.x.xxx\compilers_and_libraries_2017\linux\ipp\lib` to the target device along with the application.

8.8.7 Intel® Math Kernel Library runtime 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 from `<INSTALLDIR>\system_studio_2017.x.xxx\compilers_and_libraries_2017\linux\mkl\lib` to the target device along with the application.

8.8.8 Intel® TBB Library runtime shared object installation

If you are using dynamic linking when using the Intel® Threading Building Blocks, you will need to copy the relevant Linux* shared objects from
`<INSTALLDIR>\system_studio_2017.x.xxx\compilers_and_libraries_2017\linux\tbb\lib`
to the target device along with the application.

8.8.9 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_and_libraries_2017.x.xxx/linux/compiler/lib`
for the corresponding platform and target OS.

8.9 Eclipse* IDE Integration

8.9.1 Installation

The Intel® C++ Compiler, Intel's enhanced GDB, Intel® System Debugger and Intel® VTune™ Amplifier for Systems can be automatically integrated into a preexisting Eclipse* CDT installation. The Eclipse* CDK, Eclipse* JRE and the Eclipse* CDT integrated development environment are 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.
You can choose to install the included Eclipse* package under the directory `<INSTALLDIR>/eclipse_mars` or point the installer to an existing Elipse* directory (usually `/opt/eclipse`) or skip the integration of Intel® System Studio components into Eclipse* completely.

The prerequisites for successful integration into an existing Eclipse* environment are:

1. Eclipse* 3.8/4.2 (Juno), Eclipse* 4.3 (Kepler), Eclipse* 4.4 (Luna) or Eclipse* 4.5 (Mars)
2. Eclipse* CDT 8.0 – 8.8
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.

8.9.2 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-INSTALLDIR>/eclipse/eclipse
```

8.9.3 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

```
<INSTALLDIR>/documentation_2017/en/iss2017/iss_ug.pdf
```

8.9.4 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

8.9.5 Integrating the provided GDB into Eclipse* for remote debug

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.

1. Copy the `gdbserver` provided by the product installation
   `<INSTALLDIR>/debugger_2017/gdb/targets/<arch>/<OS>/bin`
   to the target system and add it to the execution PATH environment variable on the target.
2. Configure Eclipse* to point to the correct GDB installation:

   a. Inside the Eclipse* IDE click on Window>Preferences from the pulldown menu.
   b. Once the preferences dialogue appears select C++>Debug>GDB from the treeview on the left.
   c. The GDB executable can be chosen by editing the “GDB debugger” text box. Point to `<INSTALLDIR>/debugger_2017/gdb/intel64/bin`.

8.9.6 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 steps:

1. Navigate to the `"Help > Install New Software "` entry in the pulldown menu
2. Select `"Add"` and `"Archive"` in the following menus …
3. Browse to `<INSTALLDIR/ system_studio_2017.x.xxx/ide_support_2017/eclipse`
4. Click on the `com.intel.iss.ide.integration.site.all-1.0.0-SNAPSHOT.zip` file.

8.10 Wind River* Workbench* IDE Integration

8.10.1 Documentation

1. You will find a detailed `README` file on the integration particulars of Intel® System Studio in the `wr-iss-2017` 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.

8.10.2 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, Intel® Threading Building Blocks 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-2017` 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 <INSTALLDIR>`, 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® Systeom components such as Intel® C++ Compiler, Intel® VTune™ Amplifier and Intel® System Debugger.

8.10.3 Manual installation

1. Change into the Wind River* Workbench* installation directory and there into the `../wr-iss-2017/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.

8.10.4 Uninstall

3. Change into the Wind River* Workbench* installation directory and there into the `../wr-iss-2017/wr-setup` subdirectory.
4. In the `wr-setup` subdirectory, execute the script `uninst_wr_iss.sh`.

8.11 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.

8.12 Removing the Product

To uninstall the Intel® System Studio, change to the `/opt/intel/system_studio_2017.x.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* 6, 7 or 8

RPM package access on Wind River* Linux* 6, 7 or 8 may be slow and cause the Intel® System Studio installation to take a long time. On Wind River* Linux* 6, 7, or 8 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 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

9.1.3 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.

9.1.4 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 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

9.2.2 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 2017 Beta.

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

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.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at Intel.com, or from the OEM or retailer.

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, Xeon, and Xeon Phi 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