Intel® System Studio 2016 Update 4
Installation Guide and Release Notes

Installation Guide and Release Notes for Windows* Host

28 September 2016

Contents

1 Introduction ................................................................. 4
2 What’s New ................................................................. 5
   2.1 Versions History ..................................................... 6
3 Intel® Software Manager ................................................. 12
4 Product Contents ......................................................... 13
5 Getting Started ............................................................ 13
6 Technical Support and Documentation ............................... 14
   6.1 Release / Installation Notes and User Guides Location ....... 14
   6.2 Articles and Whitepaper Locations .............................. 16
   6.3 Support .............................................................. 16
   6.4 Support for native code generation for Intel® Graphics Technology ......................................................... 16
7 System Requirements ..................................................... 17
   7.1 Supported Host Platforms .......................................... 17
   7.2 Eclipse® Integration Prerequisites ................................. 17
   7.3 Host Prerequisites and Resource Requirements .............. 17
      7.3.1 Host Space Requirements by Component ................ 17
      7.3.2 Intel® Integrated Performance Primitives (Intel® IPP) Details ......................................................... 18
      7.3.3 Intel® C++ Compiler ........................................... 18
   7.4 Target Software Requirements .................................... 18
   7.5 Target Prerequisites and Resource Requirements ............ 18
      7.5.1 Target Space Requirement by Component ................ 18
      7.5.2 Intel® VTune™ Amplifier target OS kernel configuration ......................................................... 19
      7.5.3 Intel® VTune™ Amplifier Feature vs. Resource Matrix ......................................................... 20
   7.6 Hardware Requirements ............................................ 20
   7.7 Intel® Graphics Technology development specific requirements ......................................................... 20
8 Installation Notes ........................................................................................................... 21
8.1 Installing the Tool Suite .............................................................................................. 21
  8.1.1 Running the Installer ............................................................................................. 21
  8.1.2 Activating the Product .......................................................................................... 21
  8.1.3 Default / Customized Installation ......................................................................... 22
8.2 Uninstalling / Modifying / Repairing the Tool Suite .................................................... 22
8.3 Installation Directory Structure .................................................................................. 22
8.4 Development Target Package Installation .................................................................... 24
  8.4.1 Intel® Inspector Command line interface installation ........................................... 25
  8.4.2 Intel® VTune™ Amplifier Collectors Installation on Remote Systems .................. 25
  8.4.3 Preparing a Target Android* System for Remote Analysis ................................. 26
  8.4.4 Intel® VTune™ Amplifier Sampling Enabling Product Installation ....................... 26
  8.4.5 Intel® Integrated Performance Primitives redistributable installation .................. 26
  8.4.6 Intel® Math Kernel Library redistributable shared object installation ................ 26
  8.4.7 Intel® C++ Compiler dynamic runtime library installation ................................... 27
8.5 Eclipse* IDE Integration .............................................................................................. 27
  8.5.1 Installation ............................................................................................................. 27
  8.5.2 Launching Eclipse for Development with the Intel C++ Compiler ....................... 27
  8.5.3 Editing Compiler Cross-Build Environment Files ................................................ 27
  8.5.4 Cheat Sheets ......................................................................................................... 28
  8.5.5 Integrating the provided GDB into Eclipse* for remote debug ......................... 28
  8.5.6 Integrating the Intel® System Debugger into Eclipse* .......................................... 28
8.6 Wind River* Workbench* IDE Integration ................................................................ 28
  8.6.1 Documentation ..................................................................................................... 28
  8.6.2 Installation ............................................................................................................ 29
  8.6.3 Manual installation ............................................................................................... 29
  8.6.4 Uninstall ................................................................................................................. 29
8.7 Installing Intel® XDP3 JTAG Probe ............................................................................ 29
8.8 Ordering JTAG Probe for the Intel® System Debugger ............................................... 30
9 Known Issues and Limitations ....................................................................................... 30
  9.1.1 Documentation Links and FAT32 file system ....................................................... 30
  9.1.2 Running online-installer behind proxy server fails ............................................. 30
  9.1.3 Some hyperlinks in HTML documents may not work when you use Internet Explorer. ........................................................... 30
  9.2 Graphics Analysis Tools ........................................................................................... 31
9.2.1 Intel® Graphics Performance Analyzers installation problem ..................31
9.2.2 Installation failure on Windows* host with script custom actions ............31
9.2.3 Support for Intel® Atom™ Processor Z3560 and Z3580 code-named “Moorefield” missing 31

9.3 Intel® Energy Profiler ........................................................................32
9.3.1 \boot\config-‘uname –r’ file must be present on platform .........................32
9.3.2 Power and Frequency Analysis support for Intel® Atom™ Processor covers Android* OS only. .................................................................32

9.4 Intel® VTune™ Amplifier Usage with Yocto Project* ..............................32
9.4.1 Building Sampling Collector (SEP) for Intel® VTune™ Amplifier driver on host Linux* system .................................................................32
9.4.2 Remote Intel® VTune™ Amplifier Sampling on Intel® 64 Yocto Project* Builds ..32
9.4.3 Building 64bit Sampling Collector against Yocto Project* targeting Intel® Atom™ Processor E38xx requires additional build flags ..................................32

9.5 Intel® System Debugger .........................................................................32
9.5.1 Intel® Puma™ 6 Media Gateway Firmware Recovery Tool not available ........32
9.5.2 Using the symbol browser on large data sets and large symbol info files not recommended .................................................................33
9.5.3 Limited support for Dwarf Version 4 symbol information ........................33

9.6 GDB* - GNU* Project Debugger .............................................................33
9.6.1 Eclipse* integration of GDB* requires Intel® C++ Compiler install ..........33

9.7 Intel® Integrated Performance Primitives ..................................................33
9.7.1 Some Intel® IPP domains are not installed by default ..........................33

9.8 Intel® C++ Compiler ..............................................................................33
9.8.1 “libgcc_s.so.1” should be installed on the target system .......................33

10 Attributions ..............................................................................................34

11 Disclaimer and Legal Information ..............................................................35
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

The actual product release contains the following new features and changes. More detailed information in the respective product release notes (s. also section ‘6.1 Release Notes and User Guide Location’) or on the release notes weblinks.

Intel® System Studio 2016 Update 4

1. Intel® C++ Compiler
   - Update to version 16.0.4
   - Support for Microsoft® Visual Studio® 2015 Update 3
   - Several fixes for reported problems
   - Documentation updates
   More detailed information in the full product release notes (s. section ‘6.1 Release Notes and User Guide Location’)

2. Intel® Math Kernel Library (Intel® MKL)
   - Update to version 11.3 Update 4
   New features:
   - BLAS:
     o Introduced new packed matrix multiplication interfaces (?gemm_alloc, ?gemm_pack, ?gemm_compute, ?gemm_free) for single and double precisions.
     o Improved performance over standard S/DGEMM on Intel Xeon processor E5-xxxx v3 and later processors.
   - LAPACK:
     o Improved LU factorization, solve, and inverse (?GETR?) performance for very small sizes (<16).
     o Improved General Eigensolver (?GEEV and ?GEEVD) performance for the case when eigenvectors are needed.
     o Added TBB parallelism for ?ORGQR/?UNGQR.
   More detailed information on the Intel® MKL release notes webpage.

3. Intel® Integrated Performance Primitives (Intel® IPP)
   - Update to version 9.0 Update 4
   - Fixed a number of reported defects
   More detailed information on the Intel® IPP release notes webpage

4. Intel® Threading Building Blocks (Intel® TBB)
   - Update to version 4.4 Update 6
   Changes:
     - For 64-bit platforms, quadrupled the worst-case limit on the amount of memory the Intel TBB allocator can handle.
   Bugs fixed:
     - Fixed a memory corruption in the memory allocator when it meets internal limits.
• Fixed the memory allocator on 64-bit platforms to align memory to 16 bytes by default for all allocations bigger than 8 bytes.
• Fixed parallel_scan to provide correct result if the initial value of an accumulator is not the operation identity value.
• As a workaround for crashes in the Intel TBB library compiled with GCC 6, added -flifetime-dse=1 to compilation options on Linux* OS

More detailed information on the Intel® TBB release notes webpage

5. Intel® System Debugger
• Update to version 2016 Update 4 (internal version U1629)
• Fix for AET decoder crash after multiple start/stop cycles
• Several fixes for reported problems

New features for System Trace:
• Architectural Event Traces (AET) support added
• CSME verbosity can be set to ‘Verbose’ or ‘Normal’ in the configuration editor.
• Eclipse* Neon (4.6) supported
• New buttons for de-/selecting all traces sources in the Event Distribution View (EDV).

More detailed information in the full product release notes (s. section ‘6.1 Release Notes and User Guide Location’)

6. Intel® Graphics Performance Analyzers (Intel® GPA)
• Update to version 2016 R2

More detailed information on the Intel® GPA release notes webpage

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

Intel® System Studio 2016 Update 3

7. Intel® C++ Compiler
• Annotated source listing
  o 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.

8. Intel® Math Kernel Library (Intel® MKL)
• BLAS:
  o Improved small matrix [S,D]GEMM performance on Intel AVX2
  o Improved [C,Z]GEMV, [C,Z]TRMV, and [C,Z]TRSV performance on Intel AVX2
• LAPACK:
  o Updated Intel MKL LAPACK to the latest LAPACK version 3.6 specification. New features introduced in this version are:
    ▪ SVD by Jacobi ([CZ]GESVJ) and preconditioned Jacobi ([CZ]GEJSV) algorithms
    ▪ SVD via EVD allowing computation of a subset of singular values and vectors (?GESVDX)
    ▪ Level 3 BLAS versions of generalized Schur (?GGES3), generalized EVD (?GGEV3), generalized SVD (?GGSVD3) and reduction to generalized upper Hessenberg form (?GGH3)
    ▪ Multiplication of general matrix by a unitary/orthogonal matrix possessing 2x2 structure ( [DS]ORM22/[CZ]UNM22)
    o Improved check of parameters for correctness in all LAPACK routines to enhance security

• SCALAPACK:
  o Improved hybrid (MPI + OpenMP) performance of ScaLAPACK/PBLAS by increasing default block size returned by pilaenv

• SparseBlas:
  o Added examples that cover spmm and spmmd functionality
  o Improved performance of parallel mkl_sparse_d_mv for general BSR matrices on Intel AVX2
  o Parallel Direct Sparse Solver for Clusters:
    o Improved performance of solving step for small matrices (less than 10000 elements)
    o Added mkl_progress support in Parallel Direct sparse solver for Clusters and fixed mkl_progress in Intel MKL PARDISO

• Vector Mathematical Functions:
  o Improved implementation of Thread Local Storage (TLS) allocation/de-allocation, which helps with thread safety for DLLs in Windows when they are custom-made from static libraries

9. Intel® Integrated Performance Primitives (Intel® IPP)
• Improved zlib decompression performance for small data for Intel® 64 architectures.
• Fixed a number of defects, including the memory corruption problem on ippiSet_16u_C1R functions.

10. Intel® Threading Building Blocks (Intel® TBB)
• Removed a few cases of excessive user data copying in the flow graph.
• Improved robustness of concurrent_bounded_queue::abort() in case of simultaneous push and pop operations.
• 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.
• Added tbb::flow::async_msg, a special message type to support communications between the flow graph and external asynchronous activities.
• async_node modified to support use with C++03 compilers

Bugs fixed:
• Fixed a bug in dynamic memory allocation replacement for Windows* OS.
• Fixed excessive memory consumption on Linux* OS caused by enabling zero-copy realloc.

11. Intel® System Debugger
• Support for Eclipse* 4.5 (Mars.2) 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 processors debugging.
• A new EFI script and three buttons are added for loading PEI/DXE modules easily in System Debug

12. Intel® VTune™ Amplifier for Systems
• Support for the next generation Intel® Xeon® Processor E5 v4 Family (formerly codenamed “Broadwell-EP”)
• 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 the Memory Access analysis.
• Support for the Microsoft* Visual Studio* 2015 Update 2

Preview features:
• Disk Input and Output analysis that monitors utilization of the disk subsystem, CPU and processor buses, helps 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.

Intel® Energy Profiler for Windows:
• Update to version v1.14.1
• Extended collection start time information to include microseconds to better enable correlation with event trace logs.
• Corrected reporting of Gfx P-states on Intel® 6th Generation Core™ (formerly code-named “Skylake”) platform.
13. Intel® Graphics Performance Analyzers (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.

**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® Threading Building Blocks threading support for all and OpenMP* for some BLAS level-1 functions.

3. Intel® Integrated 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 theippiFilterBorderSetMode function to support high accuracy rounding mode inippiFilterBorder.
  - Added theippiCopyMirrorBorder 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 theippiFilterBorder image filtering functions are used in direct order, which is different from theippiFilter functions in the previous releases.
- Computer Vision:
  - Added 32-bit floating point input data support to theippiSegmentWatershed function.
  - Added mirror border support to the following filtering functions:ippiFilterGaussianBorder,ippiFilterLaplacianBorder,ippiMinEigenVal,ippiHarrisCorner,ippiPyramidLayerDown, andippiPyramidLayerUp.
- Signal Processing:
  - Added theippsThreshold_LTAbsVal function, which uses the vector absolute value.
  - Added theippsIIRIIR64f functions to perform zero-phase digital IIR filtering.
- The multi-threaded libraries only depend on the 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.
- 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).
- Intel Graphics Performance Analyzers are now in a single package for Windows users.
- 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 Microsoft Windows* 10 November update
- 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
   - Support for Windows* 10
   - Support for Microsoft* Visual Studio* 2015

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 Windows System Tray application to provide a simplified delivery mechanism for product updates, 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](http://intel.ly/SoftwareImprovementProgram).
4 Product Contents

The product contains the following components:

1. Intel® C++ Compiler 16.0 Update 4
2. Intel® Integrated Performance Primitives 9.0 Update 4
3. Intel® Math Kernel Library 11.3 Update 4
4. Intel® Threading Building Blocks 4.4 Update 6
5. Intel® System Debugger 2016
   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 (Provided under GNU General Public License v3) (64-bit host only)
   7.1. Source of GNU* GDB 7.8 (Provided under GNU General Public License v3)
8. Intel® VTune™ Amplifier 2016 Update 3 for Systems with Intel® Energy Profiler
9. Intel® Inspector 2016 for Systems
10. Intel® Graphics Performance Analyzers 2016 R2

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 <install-dir>. The default installation directory is C:\Program Files (x86)\IntelSWTools unless indicated differently.

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_win.htm
6 Technical Support and Documentation

6.1 Release / Installation Notes and User Guides Location

The release notes and getting started guides for the tools components making up the Intel® System Studio product can be found at the following locations after installation is complete.

Please refer also to ch. ‘8.1 Development target package installation’ for more information on how to set up target systems.

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
- \<install-dir\>\documentation_2016\en\compiler_c\iss2016\l_a_compiler_get_started.htm

GNU* GDB / Intel® Debugger for Heterogeneous Compute

- \<install-dir\>\documentation_2016\en\debugger\iss2016\gdb\get_started.htm
- \<install-dir\>\documentation_2016\en\debugger\iss2016\gdb\GDB_Release_notes.pdf
- \<install-dir\>\documentation_2016\en\debugger\gdb-ia\gdb.pdf

Intel® Integrated Performance Primitives

- \<install-dir\>\documentation_2016\en\ipp\iss2016\ReleaseNotes.htm
- \<install-dir\>\documentation_2016\en\ipp\common\get_started.htm

Intel® Math Kernel Library

- \<install-dir\>\documentation_2016\en\mkl\common\Release_Notes.htm
- \<install-dir\>\documentation_2016\en\mkl\iss2016\get_started.htm

Intel® Threading Building Blocks

- \<install-dir\>\documentation_2016\en\tbb\common\Release_Notes.txt
- \<install-dir\>\documentation_2016\en\tbb\common\get_started.htm

Intel® System Debugger

- \<install-dir\>\documentation_2016\en\debugger\iss2016\system_debugger\system_debug\sysdebug-release-install.pdf
• <install-dir>\documentation_2016\en\debugger\iss2016\system_debugger\system_trace\systrace-release-install.pdf
• <install-dir>\documentation_2016\en\debugger\iss2016\system_debugger\get_started.htm
• <install-dir>\documentation_2016\en\debugger\iss2016\system_debugger\system_trace\system_trace-user-guide.pdf

Intel® VTune™ Amplifier for Systems

• <install-dir>\Vtune Amplifier for Systems\documentation\en\amplsys_install_guide_windows.pdf
• <install-dir>\Vtune Amplifier for Systems\documentation\en\release_notes_amplifier_for_systems_windows.pdf
• <install-dir>\Vtune Amplifier for Systems\documentation\en\socwatch_<OS>_release_notes.pdf
• <install-dir>\Vtune Amplifier for Systems\documentation\en\socwatch_<OS>_users_guide.pdf
• <install-dir>\Vtune Amplifier for Systems\documentation\en\SEP_Users_Guide.pdf
• <install-dir>\Vtune Amplifier for Systems\documentation\en\emon_user_guide.pdf

Intel® Inspector for Systems

• <install-dir>\Inspector for Systems\documentation\en\Release_Notes_Inspector_Windows.pdf

Intel® Graphics Performance Analyzers (Intel® GPA)

Release Notes of the latest Intel® GPA 2016 R1 release can be found at:

Documentation of the Intel® GPA is available at:
https://software.intel.com/en-us/articles/intel-gpa-online-help

Intel® System Studio - Target User Documentation

After unpacking the <install-dir>\system_studio_2016.3.xxx\Targets\system_studio_target.tgz package you can find several documentation to setup target systems for operation:

• A user’s guide for WuWatch for Android* targets:
  ../system_studio_target/wwwatch_android_v3.x.x/WakeUpWatchForAndroid.pdf

• User guides for SocWatch for Android* and Linux* targets:
  ../system_studio_target/socwatch_linux_v2.x_x86[64]/SoCWatchForLinux.pdf
  ../system_studio_target/socwatch_linux_v2.x_x86[64]/ReleaseNotes.pdf
  ../system_studio_target/socwatch_android_v1.x.x/SoCWatchForAndroid_vX_x.x.pdf
6.2 Articles and Whitepaper Locations

Intel® System Studio Tutorials and Samples

Intel® System Studio Articles and Whitepapers
- 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.

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 the option `/Qgpu-arch:<arch>`. The option is described in detail in the User’s Guide.

7 System Requirements

7.1 Supported Host Platforms
One of the following operation 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).

- Windows* 7, 8.x, 10

Individual Intel® System Studio 2016 components may support additional distributions. Please refer to the release notes of the respective components (section ‘6.1 Release Notes and User Guides Locations’).

7.2 Eclipse* Integration Prerequisites

During System Studio installation you have the option to integrate product components into an existing Eclipse* installation, install Eclipse* that is included in the System Studio package or skip Eclipse integration at all.

If you decide to integrate System Studio tools into your existing Eclipse* installation (usually this would be C:\Program Files (x86)\eclipse\), make sure the prerequisites meet the following:

- Eclipse* IDE for C/C++ Developers, supported versions 3.8/4.2 (Juno) – Eclipse* 4.5 (Mars); additionally Eclipse* 4.6 (Neon) support for the Intel® System Debugger for System Trace
- 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

<table>
<thead>
<tr>
<th>Component</th>
<th>Minimum RAM</th>
<th>Recommended RAM</th>
<th>Disk Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® System Studio</td>
<td>2Gb</td>
<td>4Gb</td>
<td>7Gb</td>
</tr>
<tr>
<td>Intel® C++ Compiler</td>
<td>1Gb</td>
<td>2Gb</td>
<td>2.5Gb</td>
</tr>
<tr>
<td>Intel® Integrated Performance Primitives</td>
<td>1Gb</td>
<td>4Gb</td>
<td>1.2Gb</td>
</tr>
<tr>
<td>Intel® Math Kernel Library</td>
<td>1Gb</td>
<td>4Gb</td>
<td>2.3Gb</td>
</tr>
</tbody>
</table>
### 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 or 6.x installation that the compiler can integrate into.

### 7.4 Target Software Requirements

The target platform should be based on one of the following environments:

- Yocto Project* 1.4, 1.5, 1.6, 1.7 based environment
- CE Linux* PR35 based environment
- Tizen* IVI 3.x
- Wind River* Linux* 4, 5, 6 based environment
- Android* 4.0.x through 4.4.x

**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
<table>
<thead>
<tr>
<th>Application</th>
<th>Minimum RAM</th>
<th>Dependencies</th>
<th>Disk Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® C++ Compiler</td>
<td>application dependent</td>
<td>Linux kernel 1.26.18 or newer glibc-2.5 or compatible libgcc-4.1.2 or compatible libstdc++-3.4.7 or compatible</td>
<td>13Mb (IA-32) 15Mb (Intel® 64)</td>
</tr>
<tr>
<td>Intel® VTune™ Amplifier CLI</td>
<td>4Gb</td>
<td>Specific kernel configuration reqs. Details below.</td>
<td>200Mb</td>
</tr>
<tr>
<td>Intel® VTune™ Amplifier SEP</td>
<td>(# logical cores+2) Mb</td>
<td>Specific kernel configuration reqs. Details below.</td>
<td>8Mb</td>
</tr>
<tr>
<td>SoC Watch</td>
<td>(# logical cores+2) Mb</td>
<td>Specific kernel configuration reqs. See SoCWatch documentation</td>
<td>8Mb</td>
</tr>
<tr>
<td>WakeUp Watch</td>
<td>(# logical cores+2) Mb</td>
<td>Specific kernel configuration reqs. See WuWatch documentation</td>
<td>8Mb</td>
</tr>
<tr>
<td>Intel® Inspector for Systems CLI</td>
<td>2Gb</td>
<td></td>
<td>350Mb</td>
</tr>
<tr>
<td>gdbserver</td>
<td>negligible</td>
<td></td>
<td>none</td>
</tr>
<tr>
<td>xdbntf.ko</td>
<td>&lt;1Mb</td>
<td>kernel build environment</td>
<td>&lt;1Mb</td>
</tr>
</tbody>
</table>

### 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
  RING_BUFFER=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

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

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, Z2xxx, Z3xxx, E3xxx, 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.
- Alternatively development platform based on 5th generation Intel® Core™ M processor

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:

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

C:\Program Files (x86)\IntelSWTools

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:


8.1.1 Running the Installer
You have the choice to use the online installer which is a small agent that downloads installation packages according to the products you will choose for installation. Alternatively you can use the full package offline installer which doesn’t require an Internet connection for installation.

To start installation, run one of the following (you must have administrator rights):

- Double-click the downloaded online installer agent
  system_studio_2016.3.xxx_online.exe
  or
- Double-click the downloaded full package offline installer
  system_studio_2016.3.xxx.exe

8.1.2 Activating the Product
During installation of the Intel® System Studio 2016 an activation dialog pops up providing 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)
- Evaluation activation (no serial number required; installs a 30-days license on the system with full functionality)
- Use a license manager (license manager must be running and accessible from the install machine)
- Use license file (license file .lic must be available on the install machine, no internet connection required)
The Intel® Software Manager (see section. 3) can be used to manage your activations after product installation. It can for example convert an evaluation activation into a full product activation (after product license purchase) without re-installing the product.

8.1.3  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.

8.2  Uninstalling / Modifying / Repairing the Tool Suite
You can uninstall the complete product, modify (if you want to uninstall specific component or install new components) or repair an installation (if you think something has broken). You can choose one of the following:

- Start the Windows* system’s Control Panel, choose ‘Uninstall a program’ / Intel System Studio 2016 [Update #]
  or
- Run the c:\Program Files (x86)\Intel\Download\system_studio_2016.3.xxx\setup.exe

and choose the desired option, ‘Modify’, ‘Repair’ or ‘Remove’

8.3  Installation Directory Structure
Intel® System Studio 2016 installs components which are unique to System Studio into <install-dir>\system_studio_2016.3.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 for Intel® C++ Compiler, Intel® Integrated Performance Primitives, Intel® Math Kernel Library, Intel® Threading Building Blocks, Intel® System Debugger, Intel® VTune™ Amplifier for Systems, Intel® Inspector for Systems respectively and Wind River Linux* development environment integration as follows:

- <install-dir>\system_studio_2016.3.xxx\compilers_and_libraries_2016
- <install-dir>\system_studio_2016.3.xxx\debugger
- <install-dir>\system_studio_2016.3.xxx\documentation_2016
- <install-dir>\system_studio_2016.3.xxx\GPA_2016
- <install-dir>\system_studio_2016.3.xxx\GraphicsFrameAnalyzerOGL
- <install-dir>\system_studio_2016.3.xxx\ide_support_2016
- <install-dir>\system_studio_2016.3.xxx\Inspector for Systems
- <install-dir>\system_studio_2016.3.xxx\licensing
- <install-dir>\system_studio_2016.3.xxx\samples_2016
The Intel® System Studio contains components under GNU* 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, SoC Watch, WakeUp Watch and Intel® Inspector on the development target, please follow the sub-chapter on the command line interface (CLI) installations below.

Furthermore a target package system_studio_target.tgz in <install-dir>\system_studio_2016.3.xxx\Targets contains Intel® C++ Compiler runtime libraries, the Intel® VTune™ Amplifier Sampling Enabling Product (SEP), target components for the Intel® VTune™ Amplifier Data Collector and the kernel module used by the Intel® System Debugger to export Linux* dynamically kernel module memory load information to host.

The Intel® System Debugger at this point is intended for Intel® Atom™ processor targeted system software cross-debug.
8.4 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 `<install-dir>\system_studio_2016.3.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 compiler runtime libraries that you find in `.../system_studio_target/compilers_and_libraries_2016.3.175/linux/compiler/lib/<OS>` to your target environment search path.

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_2016/gdb/targets/<arch>/<OS>`, where `arch` and `OS` can be the following:
   o arch: ia32
     target: Android, CELinuxPR35, ChromiumOS, KendrickCanyon, TizenIVI, WindRiverLinux4, WindRiverLinux5, WindRiverLinux6, WindRiverLinux7, Yocto1.4, Yocto1.5, Yocto1.6, Yocto1.7, Yocto2.0
   o arch: intel64
     target: Android
   o arch: intel64_igfx
     target: N/A (graphic offload, debugger for heterogeneous computing)
   o arch: Quark
     target: Galileo (eglibc, 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_2016_update3_for_systems_target/linux/vtune_amplifier_target_sep_x86[64].tgz`
arch: 32, 64

5. For the Intel® VTune Amplifier for Systems target package pick
   
   
   .../system_studio_target/vtune_amplifier_2016_update3_for_systems_target/linux/vtune_amplifier_target_x86[ _64].tgz
   
   arch: 32, 64

6. For WakeUp Watch for Android* follow the instructions at
   
   
   .../system_studio_target/wuwatch_android_v3.1.6b
   
   /WakeUpWatchForAndroid.pdf

7. For SoC Watch for Android* follow the instructions at
   
   
   .../system_studio_target/socwatch_android_vx.x.x/SocWatchForAndroid_vx_x_x.pdf

8. For SoC Watch for Linux* follow the instructions at
   
   
   .../system_studio_target/socwatch_linux_v2.1/SoCWatchForLinux.pdf

9. For the Intel® Inspector for Systems follow the instructions in
   
   

8.4.1 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. The fully functional command-line Intel® Inspector installation can be found in the bin32 and bin64 subdirectories for IA32 and Intel® 64 targets respectively.

8.4.2 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_2016_update3_for_systems_target/linux/vtune_amplifier_target_x86[ _64].tgz
   
   arch: 32, 64
   
   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.
8.4.3 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_vx.x.x/
   
   on the target.

   o ..../system_studio_target/socwatch_android_vx.x.x/
     SocWatchForAndroid_vx_x_x.pdf


2. You will find WakeUp Watch at
   
   ..../system_studio_target/wuwatch_android_v3.1.6b/

   on the target.

   Please follow the instructions for installation and usage in

   o ..../system_studio_target/socwatch_android/
     WakeUpWatchForAndroid.pdf


8.4.4 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_2016_update3_for_systems_target/linux/vtune_amplifier_target_sep_x86[_64].tgz

2. After unpacking this zip file follow the instructions in
   
   ../vtune_amplifier_2016_for_systems.2.x.xxxxxx\sepdk\src\README.txt

8.4.5 Intel® Integrated Performance Primitives redistributable 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.

The redistributable shared objects can be found at

<install-dir>\system_studio_2016.3.xxx\compilers_and_libraries_2016\linux\ipp\lib

8.4.6 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.

The redistributable shared objects can be found at
<install-dir>\system_studio_2016.3.xxx\compilers_and_libraries_2016\linux\mkl\lib

### 8.4.7 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

```bash
.../system_studio_target/
compilers_and_libraries_2016.3.xxx/linux/compiler/lib/<arch>,
```

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

### 8.5 Eclipse* IDE Integration

#### 8.5.1 Installation

During System Studio installation you have the option to integrate product components into an existing Eclipse* installation, install Eclipse* (actually v4.5 Mars) that is included in the System Studio package or skip Eclipse integration at all. If you decide to integrate System Studio tools into your existing Eclipse* installation (usually this could be `C:\Program Files (x86)\eclipse`) then make sure the prerequisites meet the following:

- Eclipse* IDE for C/C++ Developers, supported versions 3.8/4.2 (Juno) – Eclipse* 4.5 (Mars); additionally Eclipse* 4.6 (Neon) support for the Intel® System Debugger for System Trace
- 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.

#### 8.5.2 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.:

```bash
eclipse --vm \JRE folder\bin\java
```

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

```bash
<eclipse-install-dir>\eclipse\eclipse
```

#### 8.5.3 Editing Compiler Cross-Build Environment Files


For details on the Environment File Editor, please check the Intel® C++ Compiler documentation at [Link](#)
8.5.4 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

8.5.5 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
   (<install-dir>\debugger_2016\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
      <install-dir>\debugger_2016\gdb\intel64\bin,.

8.5.6 Integrating the Intel® System Debugger into Eclipse*
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 “Archive” in the following menus ...
3. Browse to <install-dir>\System Debugger 2016\system_debug\ide_plugins\.
4. Click on the com.intel.debugger.update-site-x.x.x-SNAPSHOT.zip file.

8.6 Wind River* Workbench* IDE Integration

8.6.1 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.

8.6.2 Installation

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. 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.bat. 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.

8.6.3 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.bat <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.

Note for Windows* host: As the Wind River* Linux* target platform is defined on Linux* host and only imported into the Wind River* Workbench* on Windows* host, Intel® System Studio platform recipes may not be applicable for Windows* host users.

8.6.4 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.bat

8.7 Installing Intel® XDP3 JTAG Probe

If it is not already pre-installed, the Intel® ITP-XDP3 driver is automatically installed as part of the Intel® System Debugger installation process.
The Intel® ITP-XDP3 driver installer will issue a warning that the publisher could not be verified. Please acknowledge the warning and proceed with the installation.

8.8 Ordering JTAG Probe for the Intel® System Debugger

1. To order the Intel XDP3 JTAG probe, please go to:
   http://designintools.intel.com/product_p/itpxdp3brext.htm (ITP-XDP 3BRKit)
2. To order the closed-chassis adapter, please go to:

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.

9 Known 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 Documentation Links and FAT32 file system
If Intel® System Studio is installed on a FAT32 file-system, the symbolic links for Intel® VTune™ Amplifier for Systems and Intel® Inspector for Systems documentation from <install-dir>\system_studio_2016.3.xxx\documentation_2016\ may not work. Please refer directly to the <install-dir>\system_studio_2016.3.xxx\VTune Amplifier for Systems\documentation\en\ and <install-dir>\system_studio_2016.3.xxx\Inspector for Systems\documentation\en\

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 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 Graphics Analysis Tools

9.2.1 Intel® Graphics Performance Analyzers installation problem
Installation of Intel® System Studio 2016 Ultimate or Professional Edition for Windows* may fail with an error on the Intel® Graphics Performance Analyzers (Intel® GPA) component. The following instruction may help to get the GPA component installed:

- Close the product installer (if it’s still running).
- Open a command prompt on Windows* with administrator privileges (hold down Ctrl and Shift keys while you open the cmd tool)
- Type in the following commands:

  regsvr32 /u jscript.dll
  regsvr32 /i jscript.dll

Alternatively, if you don’t need the GPA tools, you can de-select the Intel® Graphics Performance Analyzers component from the customer installation dialog to complete the installation of Intel® System Studio without the PGA tools being installed.

9.2.2 Installation failure on Windows* host with script custom actions
The installation of Intel® System Studio System Analyzer, Frame Analyzer and Platform Analyzer may fail on rare occasions with the following Windows* error message:

1. 2738, Could not access VBScript run time for custom action [2].
2. 2739, Could not access JScript run time for custom action [2].

If this error message occurs, the installation can be completed after applying the following steps:

- Check that vbscript.dll and jscript.dll aren’t registered in HKEY_CURRENT_USER (HKCU), checking for the registry keys below.
- VBScript, HKCU\SOFTWARE\Classes\CLSID\{ B54F3741-5B07-11CF-A4B0-00AA004A55E8}
- JScript, HKCU\SOFTWARE\Classes\CLSID\{ F414C260-6AC0-11CF-B6D1-00AA00BBB58}
- JScript, HKCU\SOFTWARE\Classes\CLSID\{ F414C261-6AC0-11CF-B6D1-00AA00BBB58}
- JScript, HKCU\SOFTWARE\Classes\CLSID\{ F414C262-6AC0-11CF-B6D1-00AA00BBB58}
- Remove these keys if they exist in HKEY_CURRENT_USER..

9.2.3 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
9.3 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 Debugger

9.5.1 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.
9.5.2 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.

9.5.3 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.

9.6 GDB* - GNU* Project Debugger

9.6.1 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.

9.7 Intel® Integrated Performance Primitives

9.7.1 Some Intel® IPP domains are not installed by default

Several Intel® IPP domains (ippRR, ippGEN, ippJP, ippAC, ippVC, and ippSC) are not installed by default. Please use the Intel® System Studio installation customization option to add them.

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:
- 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 updcre 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

© 2016 Intel Corporation.