Intel® System Studio 2017 Update 3 Release Notes and Installation Guide

Release Notes and Installation Guide for Windows* Host

18 May 2017

Contents

1 Introduction

This document provides an overview of the **Intel® System Studio 2017 Professional and Ultimate Editions for Windows* host**, installation notes, new features and changes, the release history, additional product information and references to articles and whitepapers.

The Intel® System Studio has three editions, covered by three different licenses: Composer Edition, Professional Edition, and Ultimate Edition. Each edition consists of a separate download packages for Linux* and Windows* hosts. The target audience is the performance-orientated C/C++ embedded/mobile/wearable/IoT developer who is developing on Linux* and /or Windows* host environments for Embedded Linux*, Wind River Linux*, Android* and/or Tizen* targets. For more details please refer to ch. 3. *Product Contents* to identify which sections of this document are relevant for the edition of System Studio you are using.

On completing the Intel® System Studio installation process, locate the files in the `documentation_2017\en\` folder under the target installation path which provide an Intel® System Studio User Guide as well as several Getting-started Guides for the respective System Studio components.

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

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

For full product information as well as links to evaluation licenses (30-days), please refer to Intel® System Studio 2017 product webpage https://software.intel.com/en-us/intel-system-studio.

2 What's New

This section highlights important changes in the Intel® System Studio 2017 product release. For more information on what is new in each component, please read the individual component release notes (see ‘*4.2 Documentation Locations*’).

Intel® System Studio 2017 Update 3

1. **Intel® C++ Compiler**
   - Fixes to reported problems

2. **Intel® Integrated Performance Primitives (Intel® IPP)**
   - Fixed some known problems in Intel® IPP Cryptography functions.
   - Added support for Microsoft Visual Studio* 2017 on Windows*.
   - Added Support for Conda* repositories installation
   - Visit the Intel® IPP 2017 bug fixes for the bugfix list.

   View the full release notes for more details.

3. **Intel® Math Kernel Library (Intel® MKL)**
   - BLAS:
     - Improved performance for ?GEMM for medium problem sizes on Intel® Xeon® Processor supporting Intel® Advanced Vector Extensions 512 (Intel® AVX-512) (codename Skylake Server)
     - Improved performance for SGEMM/DGEMM for small problem sizes on Intel® Xeon® Processor supporting Intel® Advanced Vector Extensions 512 (Intel® AVX-512) (codename Skylake Server)
     - Improved performance for ?GEMM_BATCH on all architectures
     - Improved performance for SSYMV/DSYMV on Intel® Advanced Vector Extensions 2 (Intel® AVX2) and later architectures
     - Improved performance for general BLAS functions on the 32-bit Intel® Advanced Vector Extensions 512 (Intel® AVX512) architecture
     - Fixed ?AXPBY to propagate NaNs in the y vector when beta = 0 on 64-bit Intel® Advanced Vector Extensions 2 (Intel® AVX2) and later architectures
   - High Performance Conjugate Gradients (HPCG):
     - Add support of Intel® Xeon® Processor supporting Intel® Advanced Vector Extensions 512 (Intel® AVX-512) (codename Skylake Server)
   - Deep Neural Networks:
     - Average pooling has an option to include padding into mean values computation
   - LAPACK:

    - Optimized ?GELQ and ?GEMLQ performance for short-and-wide matrices
    - Optimized performance of ?ORCSD2BY1 and ? DORCSD routines
    - Fixed LU performance degradation for medium sizes on 6 threads
- Vector Statistics:
    - Improved performance of Outlier Detection (BACON) algorithm for single and double precisions for processor supporting Intel® AVX2 and intel® AVX512 Instruction sets

View the full release notes for more details.

4. Intel® Threading Building Blocks (Intel® TBB)

- **Added functionality:**
    - Added support for Android* NDK r14.
- **Preview Features:**
    - Added a blocking terminate extension to the task_scheduler_init class that allows an object to wait for termination of worker threads.
- **Bugs fixed:**
    - Fixed compilation and testing issues with MinGW (GCC 6).
    - Fixed compilation with /std:c++latest option of VS 2017 (https://github.com/01org/tbb/issues/13).

In addition, you can download the latest Intel TBB open source version from https://github.com/01org/tbb/releases.

View the full release notes for more details.

5. GDB* - GNU* Project Debugger

No update for this System Studio release

6. Intel® Graphics Performance Analyzers (Intel® GPA)

No update for this System Studio release

View the full release notes for the actual Intel GPA release for more details.

7. Intel® VTune™ Amplifier for Systems

- Application Performance Snapshot (Preview) provides a quick look at your application performance and helps you understand where your application will benefit from tuning. The revised tool shows metrics on MPI parallelism (Linux* only), OpenMP* parallelism, memory access, FPU utilization, and I/O efficiency with recommendations on further in-depth analysis.
- Improved insight into parallelism inefficiencies for applications using Intel® Threading Building Blocks (Intel® TBB) with extended classification of high Overhead and Spin time.

- Automated installation of the VTune Amplifier collectors on a remote Linux* target system. This feature is helpful if you profile a target on a shared resource without VTune Amplifier installed or on an embedded platform for Microsoft Visual Studio* 2017.

  SocWatch New Features:

- Initial support for Intel Atom® E39xx platforms and 7th generation Intel® Core™ processors.
- *pkg-temp* on Intel Atom® E39xx platforms: Measure the thermal status of multiple domains in the package.
- *dram-bw* on Intel Atom® E39xx platforms: Reports total DDR bandwidth with memory slice breakdown.

8. Intel® Inspector

- Correctness improvements of uninitialized memory reads detection algorithm.
- Fixes for reported bugs.

9. Intel® System Debugger

- **System Trace:**

  New Features

  - Add support for the new OpenIPC API.
  - Timeline View: Add Context Menu for better accessibility. The menu contains items available in the editor toolbar.
  - Timeline View: The timeline view height axis is reduced to save vertical space. Keyboard shortcuts are added to all the commands available in the menu. A new command – "Suspended Details" – is added.
  - Message Router: Watchdog for Intel Trace Hub Server has been added. Heartbeat mechanism has been added between UI and the native side. This was done following RFC6455 when the client sends PING message (0x09) and the server responds with PONG message (0x0A).

  Bug Fixes

  - Fixed missing mnemonics on XML serialization, the missing mnemonics causing a delay in decode certain formats.
  - Fixed an issue related to ignoring the user extension destination. The collateral installer was ignoring the destination folder setting from the manifest.
- **System Debug:**

  New Features

  - Providing information to the users about breakpoint re-application after target reset
  - Updated Intel® DFx Abstraction layer (Intel® DAL)
  - I/O port based reset shows now an information which port writes were executed.
  - Improved System Table discovery for LoadDXEModules function.

- The Target Access Module hang detection timeout can now be configured in the options dialog. Especially for server platforms with a large number of threads increasing the timeout helps to identify real target access module hangs.
- Intel® Processor Trace Plugin added support for multiple configuration tables and regions.

Bug Fixes

- Providing an appropriate message if a target access call is taking a long time instead of trying to abort the call.
- Fixed crash during usage of the LoadPEIM function if PEI modules were built in release mode.
- I/O port based reset does not use to default reset method anymore but do a port based reset

2.1 Version History

This section highlights important changes in the previous Intel® System Studio 2017 and 2016 releases.

Intel® System Studio 2017 Update 2

1. **Intel® C++ Compiler**
   - Support for the Intel® Joule™ Development Platform
   - Support for Android NDK r14
   - Support for Eclipse* CDT 8.4 - 8.8, and 9.0
   - Fixes to reported problems

2. **Intel® Integrated Performance Primitives (Intel® IPP)**
   - Added the new functions in ZLIB to support the user-defined Huffman tables, which allows to increase ZLIB compression ratio at the fastest compression level.
   - Increased LZO compression performance by 20% to 50%. Added level 999 support in LZO decompression
   - Introduced support for Intel® Xeon Phi™ processor x200 (formerly Knights Landing) leverage boot mode in the Intel IPP examples
   - Added an example code on building custom dispatcher for the processor-specific optimization codes
   - Visit the Intel® IPP 2017 bug fixes for the bugfix list.

   View the full release notes for more details.

3. **Intel® Math Kernel Library (Intel® MKL)**
   - Library Engineering:
     - Intel® Advanced Vector Extensions 512 (Intel® AVX-512) code is dispatched by default on Intel® Xeon® processors
   - BLAS:
     - Improved performance of dgemv non transpose when number of threads are large (typically on Intel® Xeon Phi™ processor x200 (formerly Knights Landing)). For example: factor 2 speedup when M=K=10000 with 68 threads on Intel® Xeon Phi™ processor x200
     - Improved performance for dgemm, TN and NN cases, with very small N on Intel® Xeon Phi™ processor x200 and 6th Generation Intel® Core™ processor (as known as Skylake)
     - Introduced MKL_NUM_STRIPES environment variable and accompanying Intel MKL support functions to control the 2D partitioning of multithreaded *GEMM on all Intel architectures except from Intel® Xeon Phi™ Coprocessor x100 Product Family. Please see the related section in Intel MKL Developer Guide for details.
     - Improved the {s,d}gemm_compute performance on Intel64 architectures supporting Intel® AVX2 instruction set.
     - Improved ?gemm_batch performance when N==1.
   - Sparse BLAS
     - Improved performance of BCSMV functionality with 3-10, 14 and 18 problem sizes for processor supporting Intel® AVX2 and intel® AVX-512 Instruction sets
     - Improved performance of CSRMV functionality for processor supporting Intel® AVX2 and intel® AVX-512 Instruction sets
     - Added Intel® Threading Building Blocks (Intel® TBB) threading support for CSRMV functionality with symmetric matrices
   - Intel MKL Pardiso
     - Added support of Intel TBB threading support for Intel MKL Pardiso at the solving step
   - Deep Neural Networks:
     - Improved performance on Intel Xeon processors with Intel® AVX2 and Intel® AVX-512 instruction set support
     - Improved performance on the second generation of Intel® Xeon Phi™ processor x200
     - Introduced support for rectangular convolution kernels
     - Significantly improved reference convolution code performance
     - Added unsymmetric padding support in convolution and pooling
     - Introduced extended Batch Normalization API that allows access to mean, variance, scale and shift parameters
   - LAPACK:

  - Added ?GEQR, ?GEMQR and ?GETSLS functions with performance optimized for tall-and-skinny matrices.
  - Improved LAPACK performance for very small sizes (N<16) in LP64 layer by reducing internal LP64/ILP64 conversion overhead.
  - Improved ?[SY|HE]EVD scalability up to 32 and beyond threads on Intel® Xeon® and Intel® Xeon Phi™ processor x200
  - Significantly improved ?LANGE (‘Frobenius’ norm) performance
- ScaLAPACK:
  - Added MKL_RPOGRESS() support in P?GETRF
  - Improved P?TRSM/P?SYRK performance
  - Optimized ?GE(SD|RV|BS|BR)2D routines in BLACS
  - Fixed failure in P?GEMM (‘N’, ‘N’ case)
- Vector Mathematics:
  - Added Intel TBB threading support for all mathematical functions.
- Vector Statistics:
  - Improved C interfaces of vsl*SSEdit*() functions

View the full release notes for more details.

4. **Intel® Threading Building Blocks (Intel® TBB)**

- Added support for C++11 move semantics in parallel_do.
- Added support for Android* 7.0 and Android* NDK r13, r13b.

Changes affecting backward compatibility:

- Minimal compiler versions required for support of C++11 move semantics raised to GCC 4.5, VS 2012, and Intel(R) C++ Compiler 14.0.

Preview Features:

- Added template class gfx_factory to the flow graph API. It implements the Factory concept for streaming_node to offload computations to Intel(R) processor graphics.

Bugs fixed:

- The workaround for crashes in the library compiled with GCC 6 (-flifetime-dse=1) was extended to Windows*.
- Fixed a possible deadlock caused by missed wakeup signals in task_arena::execute().

Open-source contributions integrated:

- A build fix for Linux* s390x platform by Jerry J.

View the full release notes for more details.

5. **GDB* - GNU* Project Debugger**

- Support for the Intel® Joule™ Development Platform.

6. Intel® Graphics Performance Analyzers (Intel® GPA)

New Features for Analyzing All Graphics APIs

- Graphics Frame Analyzer:
  - Bar Chart now supports grouping draw calls by regions (debug regions, shader usage, etc.) for Microsoft DirectX*** 12 and OpenGL* APIs.**
- Trace Analyzer [Beta]:
  - Trace Analyzer is a new timeline analysis tool supplemental to Platform Analyzer. At this moment, Trace Analyzer provides the same functionality as Platform Analyzer and adds support of platform analysis of desktop OpenGL* applications on Ubuntu* systems. For detailed information on Trace Analyzer, see the Platform Analysis topic in online help (https://software.intel.com/en-us/articles/intel-gpa-online-help).

    Note: You cannot analyze traces on a remote system using Trace Analyzer.

    Here is a summary of the supported features per target application type:
- Microsoft DirectX* applications:
  - DirectX API calls
  - CPU and GPU queues and relations between CPU and GPU tasks
  - VSync
  - CPU and GPU frame boundaries
  - GPU and API metrics
  - ETW Events
  - Context switches
- Android OpenGL ES* applications:
  - OpenGL ES API calls
  - Context switches
  - GPU and API metrics
  - atrace events

Note:

You can analyze Android* apps on both rooted and non-rooted systems.

x86, x64 and ARM*-based 32-bit systems are supported.

- Ubuntu* OpenGL* applications:
  - OpenGL API calls
  - API metrics
  - Context switches

New Features for Analyzing Microsoft DirectX* 12 Applications

- Graphics Frame Analyzer:
  - Resource viewer for ResourceBarrier calls
  - Resource usage history. You can watch a video illustrating this feature: https://www.youtube.com/watch?v=CBCzMszOvsY
  - Support for MSAA textures

    - Support for bundles
    - Support for DirectX* 12.1
    - Arguments for DirectX* 12 API call visualization

New Features for Analyzing DirectX 9/10/11 Applications

- Graphics Frame Analyzer:
- Added support for VR games (tested on HTC Vive* and Oculus Rift*).
- Added support for high DPI Monitors.

New Features for Analyzing OpenGL/OpenGL ES* Applications

- Graphics Frame Analyzer:
    - Enabled support for ARB shaders
    - Disabled pipeline view

      Note: On Ubuntu* and macOS* host systems, Trace Analyzer is the only platform analysis tool installed by default. If you would like to analyze Android* OpenGL ES* applications with the Platform Analyzer, you need to install it separately. For desktop OpenGL* applications, you can only use Trace Analyzer to capture and analyze trace files.

Documentation Updates

- Documented all newly added product features.

View the full release notes for more details.

7. Intel® VTune™ Amplifier for Systems

- Support for cross-OS analysis to all license types. The installation packages for additional operating systems can be downloaded from registrationcenter.intel.com.
- Support for the Intel® Atom™ processors series E9xxx (formerly codenamed ‘Apollo Lake’) and Intel® Atom™ Processor C3338 (formerly codenamed ‘Denverton’), and support for 7th Gen Intel® Core™ Intel processor family (formally codenamed ‘Kabylake’).
- Support for the mixed **Python*** and native code in the **Locks and Waits** analysis including call stack collection
- **HPC Performance Characterization** analysis improvements:
    - Increased detail and structure for vector efficiency metrics based on FLOP counters in the FPU Utilization section
    - New section presenting the data on the hottest loops and functions with arithmetic operations, which enables you to identify which loops/functions with FPU Usage took the most CPU Time
- **DRAM Bandwidth Bound metric** based on uncore events used in the Memory Usage viewpoint for the **Memory Access** and **HPC Performance Characterization** analyses.
- **GPU Hotspots** Summary view extended to provide the **Packet Queue Depth** and **Packet Duration** histograms for the analysis of DMA packet execution.

New version 1.17.1 for SoCWatch for Windows*

- CPU & GPU P-state features include performance limiting reasons in their report.
- Collection across hibernation is now supported.
- Improved support for Intel platform code named Kaby Lake - H

8. **Intel® Inspector**
   - Support for C++ 17 **std::shared_mutex** and Windows **SRW Locks**, that enable threading error analysis for applications with read/write synchronization primitives.
   - Support for Window* Server 2016.
   - Support for cross-OS analysis to all license types. The installation packages for additional operating systems can be downloaded from registrationcenter.intel.com.

9. **Intel® System Debugger**
   - **System Trace:**

     New Features
     - Support for 7th Gen Intel® Core™ Processor Family (formally codenamed 'Kabylake')
     - Export of TRace Analysis and Mining (TRAM) detection results and timing table to CSV or tab delimited text files

     Bug Fixes
     - Fix copyright header in user documentation
     - Fix multiple TRAM, search and filter feature issues
     - Fix and improve TRAM detection results display
     - Fix loading performance of detection details table
     - Fix to support special characters for search/filter
     - Fix dialog exception when no trace session open
   - **System Debug:**

     New Features
     - Support for 7th Gen Intel® Core™ Processor Family (formally codenamed 'Kabylake')
     - Added the platform breakpoint types: Shutdown Break and Machine Check Break
     - For code lines that are mapping to multiple assembly blocks the context menu provides now also the options "Go Here", "Set Current Location" and "Jump to assembler".
     - Improved the messaging for cases when no additional hardware breakpoints are available.
     - Updated Simics
     - The context menu in the assembler and source window allows now to set a software or hardware breakpoint explicitly.
     - The Script file: <installdir>scripts/startup.xdb was added to allow execution of commands automatically during startup of Intel® System Debugger

- Variables are now marked as <optimized out> if not available because of compiler optimizations and as <currently unavailable> if they are out of scope.
- UEFI Buttons are loaded automatically. No need to manually execute efi.xdb anymore
- Improved DWARF Version 4 debug information support

Bug Fixes

- Corrected masking of base addresses in paging structures.
- Added correct handling of Memory Protection Key IDs in paging structures.
- For the BATCH command the option /NOOUT was fixed.
- Complex debug information decoding improved especially for Memory Reference Code (MRC) debugging.
- Setting breakpoints on source lines that map to multiple assembly locations is working now also if the source file is not part of the first symbol file loaded

Intel® System Studio 2017 Update 1

1. **Intel® C++ Compiler**
   - OpenMP* enabled now in the compiler license.
   - Supporting Clang environment in Android NDK r13 and further with a new compiler option and an environment variable:
     - New option "-[no]use-clang-env"
     - New environment variable "INTEL_USE_ENVIRONMENT" (recognized values: CLANG, GCC)
   - Fixes for reported problems.

2. **Intel® Integrated Performance Primitives (Intel® IPP)**
   - Cryptography Domain:
     - Added functions for the finite field GF(p) arithmetic, and the elliptic curves over the finite field GF(p)
     - Added ippsECCPBindGxyTblStd functions that allow to control memory size for the elliptic curves over GF(p).

   View the full release notes for more details.

3. **Intel® Math Kernel Library (Intel® MKL)**
   - BLAS :
     - The Intel Optimized MP LINPACK Benchmark supports various MPI implementations in addition to Intel MPI, and the contents of the mp_linpack directory have changed.
     - Improved single thread SGEMM/DGEMM performance on Intel® Advanced Vector Extensions 2 (Intel® AVX2), Intel® Advanced Vector Extensions 512 (Intel® AVX-512), and Intel® Xeon® for Intel® Many Integrated Core Architecture.

- Deep Neural Networks (DNN) primitives :
  - Introduced additional optimizations for Intel® Xeon® processor E3-xxxx V5.
  - Added support of non-square cores of convolution
- Sparse BLAS :
  - Improved Sparse BLAS matrix vector functionality in block compressed sparse row (BSR) format for block size equal to 6,10,14, or 18 on Intel AVX2.
  - Improved Inspector-executor Sparse BLAS matrix-vector and matrix-matrix functionality for symmetric matrices.
- LAPACK :
  - Improved performance of ?GETRF, ?GETRS and ?GETRI for very small matrices via MKL_DIRECT_CALL.
  - Improved performance of ?ORGQR and SVD functionality for tall-and-skinny matrices.
  - Parallelized ?ORGQR in Intel® Threading Building Blocks (Intel® TBB) threading layer.
- Vector Math :
  - Introduced the exponential integral function E1 with three accuracy levels HA, LA, and EP, for single precision and double precision real data types.
- ScaLAPACK :
  - Improved performance of PZGETRF.
- Known Limitations for MKL 2017 Update 1 :
  - 1D complex-to-complex FFT may return incorrect results on systems with Intel-AVX 512 support if the number of threads is different at DFT descriptor commit time and DFT execution.
  - Issue Description: If the user tries to use MSVS 2015 with our provided build.bat script to build their own xhpl.exe executable, they will see a number of unresolved external symbol errors like:
  - libhpl_intel64.lib(HPL_pdmatgen.obj) : error LNK2001: unresolved external symbol __iob_func
- Bugs fixed:
  - Parallel Direct Sparse Solver for Cluster crahes or returns wrong solving comlex linear systems with Hermit Matrixes in the case if number of MPI processes > 2
  - Fixed the problem when Intel MKL produces the error message "mismatch detected for '_MSC_VER' .... " when staticaly links with mkl_tbb_thread.lib
  - Added description of LapackE_xerbla routine to the MKL's documentation
  - Fixed the Intel MKL Pardiso failure when multiple simultaneous (single-threaded) instances are used in a OpenMP* loop
  - Fixed the Intel MKL Pardiso failures when METIS reodering and huge block on the diagonal
  - df?InterpolateEx1D function returns incorrect index of the cell that contains the right boundary of the interpolation interval
  - Extended version of the Integrate function, df?IntegrateEx1D does not pass the type of integration limit, left or right, into the callback function
  - Fixed CP2K code crash in MKLs routines after moving since MKL 11.3.1 to 11.3.2
  - Fixed Intel MKL ERROR: Parameter 3 was incorrect on entry to DSYTRF
  - Optimized performance of ?getrf, ?getrs and ?getri for very small cases.

    - Fixed mkl_lapack.h : const modificator have to be removed (?lacrm,?larcm)
    - Added support of partial SVD functionality in MKL
    - Fixed low Automatic Offload performance of mkl_?getrfnpi
    - Fixed issue with calling LAPACK functions with a workspace query so that unused parameters could be passed in as nullptr

View the full release notes for more details.

4. Intel® Threading Building Blocks (Intel® TBB)

Change:

- Removed the long-outdated support for Xbox* consoles

Bugs fixed:

- Fixed the issue with task_arena::execute() not being processed when the calling thread cannot join the arena.
- Fixed dynamic memory allocation replacement failures on Windows* 10 Anniversary Update.
- Fixed emplace() method of concurrent unordered containers not to require a copy constructor.

View the full release notes for more details.

5. Intel® Graphics Performance Analyzers (Intel® GPA)

New Features for Analyzing All Graphics APIs

- Graphics Frame Analyzer:
    - Enabled API log export into a text file.
    - Added Pixel Overdraw view for render targets.
    - Updated Metrics Viewer in accordance with the latest performance optimization guide.

New Features for Analyzing Microsoft DirectX* 12 Applications

- All Tools:
    - Added support for Universal Windows Platform (UWP) apps.
    - Added support for Windows 10 Anniversary Update. Make sure to capture frames with Intel® GPA 2016 R3 as previously captured frames are not compatible with this version.
    - Added support for the 7th Generation Intel® Core™ i7 Processors.
- Graphics Frame Analyzer:
    - Implemented extended grouping in the Bar Chart. You can now group by Render targets, Pipeline States and Shader Sets.
    - Added Wireframe and Metrics map views into the Render Target viewer.
    - Added Command Queue debug region visualization into the API log.
    - Improved geometry support.
    - Implemented API call parameters visualization.

    - Added resource labels to the API log and Resource viewer.

New Features for Analyzing DirectX 9/10/11 Applications

- Graphics Frame Analyzer:
    - Unified the opening mechanism for all frames. You can now open all frames using the Graphics Frame Analyzer, regardless of the DirectX* version. Intel(R) GPA automatically opens DirectX 9/10/11 frames in the legacy version of Graphics Frame Analyzer.
    - Fixed an issue with handling frame files with spaces in the name.

New Features for Analyzing OpenGL/OpenGL ES* Applications

- Graphics Frame Analyzer:
    - Added support for glBegin/glEnd.
    - Implemented visualization of assembly ARB shaders. Shaders are available on read-only mode.
    - Added support for GL_Texture_1D.
    - Added support for Multiple RT in highlighting and Simple FS experiments.

Documentation Updates

- Documented all newly added product features

View the full release notes for more details.

6. Intel® VTune™ Amplifier for Systems

- Support for locator hardware event metrics for the General Exploration analysis results in the Source/Assembly view that enable you to filter the data by a metric of interest and identify performance-critical code lines/instructions.
- Support for hotspot navigation and filtering of stack sampling analysis data by the *Total* type of values in the Source/Assembly view.
- **Summary** view of the General Exploration analysis extended to explicitly display measure for the hardware metrics: **Clockticks vs. Pipeline Slots**.
- Command line `summary` report for the HPC Performance Characterization analysis extended to show metrics for CPU, Memory and FPU performance aspects including performance issue descriptions for metrics that exceed the predefined threshold. To hide issue descriptions in the `summary` report, use a new `report-knob show-issues` option.
- Support for the **Average Latency** metric in the Memory Access analysis based on the driverless collection.
- PREVIEW: New **Full Compute** event group added to the list of predefined GPU hardware event groups collected for Intel® HD Graphics and Intel Iris™ Graphics. This group combines metrics from the **Overview** and **Compute Basic** presets and allows to see all detected GPU stalled/idle issues in the same view.
- **GPU Hotspots** analysis extended to detect hottest computing tasks bound by GPU L3 bandwidth.

- New **Render and GPGPU Packet Stage** grouping levels that help analyze the CPU activity while the GPU Execution Units were either idle or executing some code.
- **QPI Bandwidth** data in the Memory Usage viewpoint displayed as *data* and *non-data* to better differentiate the QPI activity for applications that are not transmitting any data between sockets.
- Analysis support for Java* applications executed with **OpenJDK.**

SoCWatch for Android*/Chrome*/Linux*:

- Update to v2.3.0
- New output formats for reports in the summary and timeline trace output files.
- Numerous fixes for reported bugs.

7. Intel® Inspector

Fixes for reported problems

Intel® System Studio 2017 Initial Release

1. General New Changes and Features

- **Host OS Support**
  - Support for IA32 based HOST system has been removed from Intel® System Studio 2017. Target OS support for IA32 however continues.
- **Online Sample Projects**
  - A new Intel® Software Product Samples and Tutorials webpage has been created to learn specific features of the various product components. For Intel® System Studio 2017 you will find also sample bundles which you can download from this webpage.
  - The online samples replace most of the samples which were provided in previous Intel® System Studio packages.
- **New Online Installer Features**
  - 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.
- **Start-up Documentation now on Eclipse* IDE Welcome page**
  - 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™ Amplifier for Systems and Intel® Inspector (which have their own standalone Eclipse framework) can be started directly from within the Eclipse* Welcome window.
- **Eclipse* package and JRE now provided with Intel® System Studio**
  - The Intel® System Studio provided Eclipse* 4.5 (Mars) package with JRE 1.8 can be optionally installed under: `<INSTALLDIR>\eclipse_<eclipse_mars>`, for example `c:\Program Files (x86)\IntelSWTools\eclipse_mars\`.

2. Intel® C++ Compiler

- Support for Eclipse* Neon 4.6 and CDT 9.0
- Android* NDK r10 is not supported. Details in Intel® C++ Compiler Release Notes.
- Compiler options starting with –o are deprecated
    - All compiler options starting with –o are deprecated. These will be replaced by new options preceded with –q. For example, -opt-report should now be –*qopt-report*. This is to improve compatibility with third-party tools that expect –o<text> to always refer to output filenames.

Annotated source listing

- This feature annotates source files with compiler optimization reports. The listing format may be specified as either text or html. The location where the listing appears can be specified as the caller site, the callee site, or both sites.

New attribute, pragma, and compiler options for code alignment

- New attribute *__attribute__((code_align(n)))* is provided to align functions to a power-of-two byte boundary *n*.
- New pragma *#pragma code_align[(n)]* is provided to align the subsequent loop heard to a power-of-two byte boundary *n*.
- New compiler option */Qalign-loops[:n]* is provided to align all loops to a power-of-two byte boundary *n*, or to provide no special alignment for loops */Qalign-loops-* (the default).

C++14 features supported under the */Qstd:c++14* options:

- C++14 variable templates (N3651)
- C++14 relaxed (aka extended) constexpr (N3652)
- C++14 sized deallocation (N3663)
- Please see C++14 Features Supported by Intel® C++ Compiler for an up-to-date listing of all supported features, including comparisons to previous major versions of the compiler.

C11 features supported under the */Qstd:c11* options:

- Support for all C11 features except C11 keyword _Atomic and __attribute((atomic))
- Please see C11 Features Supported by Intel® C++ Compiler for an up-to-date listing of all supported features, including comparisons to previous major versions of the compiler.

New and Changed Compiler Options

- */fp:consistent* Enables consistent, reproducible results for different optimization levels or between different processors of the same architecture.
- */guard:keyword* Enables the control flow protection mechanism.
- */MP-force* Disables the default heuristics used when compiler option /MP is specified. This lets you control the number of processes spawned.
- */Qalign-loops[-]* Aligns loops to a power-of-two byte boundary.
- */Qopt-report-annotate* Enables the annotated source listing feature and specifies its format.
- */Qopt-report-annotate-position* Enables the annotated source listing feature and specifies the site where optimization messages appear in the annotated source in inlined cases of loop optimizations.
- */Zo[-]* Enables generation of enhanced debugging information for optimized code.

For a list of deprecated compiler options, see the Compiler Options section of the Intel® C++ Compiler 17.0 User's Guide. Refer also to the full compiler release notes for more details.

3. Intel® Math Kernel Library (Intel® MKL)

- Introduced Deep Neural Networks (DNN) primitives including convolution, normalization, activation, and pooling functions intended to accelerate convolutional neural networks (CNNs) and deep neural networks on Intel® Architecture.
    - Optimized for Intel® Xeon® processor E5-xxxx v3 (formerly Haswell), Intel Xeon processor E5-xxxx v4 (formerly Broadwell).
    - Introduced inner product primitive to support fully connected layers.
    - Introduced batch normalization, sum, split, and concat primitives to provide full support for GoogLeNet and ResidualNet topologies.
- BLAS:
    - Introduced new packed matrix multiplication interfaces (?gemm_alloc, ?gemm_pack ,?gemm_compute, ?gemm_free) for single and double precisions.
    - Improved performance over standard S/DGEMM on Intel Xeon processor E5-xxxx v3 and later processors.
- Sparse BLAS:
    - Improved performance of parallel BSRMV functionality for processor supporting Intel® Advanced Vector Extensions 2 (Intel® AVX2) instruction set.
- Intel MKL PARDISO:
    - Improved performance of parallel solving step for matrices with fewer than 300000 elements.
    - Added support for mkl_progress in Parallel Direct Sparse Solver for Clusters.
    - Added fully distributed reordering step to Parallel Direct Sparse Solver for Clusters.
- Fourier Transforms:
    - 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® AVX-512) and IntelAVX512_MIC instruction sets
    - Improved performance for small size batched 2D FFT on the Intel Xeon processor E5-xxxx v3, and Intel Xeon processor E5-xxxx v4.
- LAPACK
    - Included the latest LAPACK v3.6 enhancements. New features introduced are:
        - SVD by Jacobi ([CZ]GESVJ) and preconditioned Jacobi ([CZ]GEJSV)
        - SVD via EVD allowing computation of a subset of singular values and vectors (?GESVDX)
        - In BLAS level 3, generalized Schur (?GGES3), generalized EVD (?GGEV3), generalized SVD (?GGSVD3), and reduction to generalized upper Hessenberg form (?GGHD3)

        - Multiplication of a general matrix by a unitary or orthogonal matrix that possesses a 2x2 block structure ([DS]ORM22/[CZ]UNM22)
    - Improved performance for large size QR(?GEQRF) on processors supporting theIntel AVX2 instruction set.
    - Improved LU factorization, solve, and inverse (?GETR?) performance for very small sizes (<16).
    - Improved General Eigensolver (?GEEV and ?GEEVD) performance for the case when eigenvectors are needed.
- ScaLAPACK
    - Improved performance for hybrid (MPI + OpenMP*) mode of ScaLAPACK and PBLAS.
    - Improved performance of P?GEMM and P?TRSM resulted in better scalability of Qbox First-Principles Molecular Dynamics code.
- Data Fitting:
    - Introduced two new storage formats for interpolation results (DF_MATRIX_STORAGE_SITES_FUNCS_DERS and DF_MATRIX_STORAGE_SITES_DERS_FUNCS).
    - Added Hyman monotonic cubic spline.
    - Modified callback APIs to allow users to pass information about integration limits.
- Vector Mathematics:
    - Improved performance for Intel Xeon processor E5-xxxx v3 and Intel Xeon processor E5-xxxx v4.
- Vector Statistics:
    - Introduced additional optimization of SkipAhead method for MT19937 and SFMT19937.

Deprecation Notices:

- Removed pre-compiled BLACS library for MPICH v1; MPICH users can still build the BLACS library with MPICH support via Intel MKL MPI wrappers.
- The SP2DP interface library is removed.
- The PGI* compiler on IA32 is no longer supported.

Known Limitations:

- cblas_?gemm_alloc is not supported on Windows* OS for the IA-32 architectures with single dynamic library linking.
- Intel MKL Integration with Microsoft* Visual Studio* in IA-32 environment is limited. This issue does not affect the Intel® 64 target environment. Intel MKL (in Intel® System Studio) integration with Microsoft* Visual Studio* is limited in both IA-32 and Intel 64 environments.
  Workaround: How to Build an Intel® MKL Application with Intel® Visual Fortran Compiler

4. Intel® Integrated Performance Primitives (Intel® IPP)

- Added Intel® IPP Platform-Aware APIs to support 64-bit parameters for image dimensions and vector length on 64-bit platforms and 64-bit operating systems:

    - This release provides 64-bit data length support in the memory allocation, data sorting, image resizing, and image arithmetic functions.
    - Intel® IPP Platform-Aware APIs support external tiling and threading by processing tiled images, which enables you to create effective parallel pipelines at the application level.
- Introduced new Integration Wrappers APIs for some image processing and computer vision functions as a technical preview. 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.
- Data Compression:
    - Added the patch files for the zlib source to provide drop-in optimization with Intel IPP functions. The patches now supports zlib version 1.2.5.3, 1.2.6.1, 1.2.7.3 and 1.2.8.
    - Significantly improved performance of zlib compression functions on the standard compression modes.
    - Introduced a new fastest zlib data compression mode, which can significantly improve compression performance with only a small sacrifice in compression ratio.
- Signal Processing:
    - Added the ippsIIRIIR functions that perform zero-phase digital IIR filtering.
    - Added 64-bit data length support to the ippsSortRadixAscend and ippsSortRadixDescend functions.
    - Added unsigned integer data support to the ippsSortRadixAscend, ippsSortRadixDescend, ippsSortRadixIndexAscend and ippsSortRadixIndexDescend functions.
- Image Processing:
    - Added the ippiScaleC functions to support image data scaling and shifting for different data types by using 64-bit floating multiplier and offset.
    - Added the ippiMulC64f functions to support image data multiplication by a 64-bit floating point value.
- Removed the tutorial from the installation package, and its sample code and documentation are now provided online.
- 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.

5. **Intel® Threading Building Blocks (Intel® TBB)**
   - static_partitioner class is now a fully supported feature.
   - async_node class is now a fully supported feature.
   - Improved dynamic memory allocation replacement on Windows* OS to skip DLLs for which replacement cannot be done, instead of aborting.
   - For 64-bit platforms, quadrupled the worst-case limit on the amount of memory the Intel TBB allocator can handle.
   - Added TBB_USE_GLIBCXX_VERSION macro to specify the version of GNU libstdc++ when it cannot be properly recognized, e.g. when used with Clang on Linux* OS. Inspired by a contribution from David A.
   - Added graph/stereo example to demostrate tbb::flow::async_msg.
   - Removed a few cases of excessive user data copying in the flow graph.
   - Reworked split_node to eliminate unnecessary overheads.
   - Added support for C++11 move semantics to the argument of tbb::parallel_do_feeder::add() method.
   - Added C++11 move constructor and assignment operator to tbb::combinable template class.
   - Added tbb::this_task_arena::max_concurrency() function and max_concurrency() method of class task_arena returning the maximal number of threads that can work inside an arena.
   - Deprecated tbb::task_arena::current_thread_index() static method; use tbb::this_task_arena::current_thread_index() function instead.
   - All examples for commercial version of library moved online: https://software.intel.com/en-us/product-code-samples. Examples are available as a standalone package or as a part of Intel® System Studio Online Samples packages.

   Changes affecting backward compatibility

   - Renamed following methods and types in async_node class:
     Old → New
     async_gateway_type → gateway_type
     async_gateway() → gateway()
     async_try_put() → try_put()
     async_reserve() → reserve_wait()
     async_commit() → release_wait()
   - Internal layout of some flow graph nodes has changed; recompilation is recommended for all binaries that use the flow graph.

   Preview Features:

   - Added template class streaming_node to the flow graph API. It allows a flow graph to offload computations to other devices through streaming or offloading APIs.
   - Template class opencl_node reimplemented as a specialization of streaming_node that works with OpenCL*.
   - Added tbb::this_task_arena::isolate() function to isolate execution of a group of tasks or an algorithm from other tasks submitted to the scheduler.

Bugs fixed:

- Added a workaround for GCC bug #62258 in std::rethrow_exception() to prevent possible problems in case of exception propagation.
- Fixed parallel_scan to provide correct result if the initial value of an accumulator is not the operation identity value.
- Fixed a memory corruption in the memory allocator when it meets internal limits.
- Fixed a race in the flow graph implementation.

Open-source contributions integrated:

- Enabling use of C++11 'override' keyword by Raf Schietekat.

6. **GNU* GDB**
   - Update to GNU* GDB 7.10.1

7. **Intel® System Debugger**
   - Support for Intel® Pentium® Processor N4200, Intel® Celeron® Processor N3350, Intel® Atom™ Processors x7-E3950, x5-3940, x3-3930 (Broxton Apollo Lake) via USB3-only (DbC) connection.

System Debug features/changes:

- Support for debug format Dwarf4
- SMM support for Intel® Core™ based pocessors debugging.
- New EFI script "EFI.xdb" adds UEFI-specific helper buttons (LoadThis, LoadPEIMs, LoadDXEModules) to the user interface to discover PEI/DXE phase debug symbols."
- Improved support for SMM debugging:
  - Special stepping handling for RSM (Return from System Management Mode) instruction: Stepping over the RSM instruction will now use SMM Exit Break.
  - SMM Entry Break and SMM Exit Break available now available from the GUI
- Single debugger startup scripts, "xdb.bat" and "xdb.sh", added. This provides a single interface to connect to different targets and platforms.

System Trace features/changes:

- Support for Architectural Event Traces (AET) on Intel® 100 Series Chipset (formerly known as Sunrisepoint H) / 6th Generation Intel® Core™ Platform I/O (formerly known as Sunrisepoint LP)
- Added new Intel® TRAM, Search and Filter features
  - Improved search performance
  - Replaced TMV configurator and rules
  - Quick search and search dialog for large datasets
  - Create and save custom search/filter scenariosr
- CSME traces verbosity configuration
  - In addition to enable/disable CSME traces in the configuration editor, CSME tracing verbosity can be selected. CSME tracing can be set to "Verbose" or "Normal".
- Support for integration of the trace viewer into Eclipse* Neon (4.6)
- Pre-configured Eclipse* Mars 64bit IDE for C/C++ developers now included in the installation package for integration of the trace viewer.

- New Buttons for de-/selecting all trace sources in the Event Distribution View (EDV).
- New column picker location: New icon and dialog where message view column can be selected.
- Column presets: The Message View toolbar contains a new functionality to use and create column presets.
- Bug fixes:
  - AET decoder crash after multiple start/stop capture cycles.
  - Suppressed SVEN messages, e.g. due to non-matching catalog, will now be reported in the Message View
  - Improve error message of File Reader decoder on empty trace
  - Enabled cancel button during memory extraction in the trace to memory use-case
  - Improve target access stability
  - Fix parameter --root-path in Trace Decode Engine (TDE) frontend
  - Fix trace hub server crash when running BIOS self-test
  - Fix for unintentional change of trace profile with "CTRL + s" shortcut
  - Fix for wrong blue background color in several system trace views
  - Fix crash on start capture without open configuration editor
  - Event Distribution View: Fix default zoom range not loaded in some scenarios
  - Event Distribution View: Fix flickering histogram depending on screen resolution.

8. **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.
   - Support for the next generation **Intel® Xeon® Processor E5 v4** Family (formerly codenamed "Broadwell-EP")
   - UI improvements for the grid views and identification of performance issues (HTML-enabled grid)
   - Navigation from the **Hottest GPU computing tasks** summary to the details provided in the **Graphics** tab
   - Support for the **Attach to Process** target analysis for **Intel Media SDK** and **OpenCL™** programs

- Easier hardware event list selection in custom EBS analysis configuration via **Filter** field.
- Support for the **Microsoft* Visual Studio* 2015 Update 2**.

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.

9. **Intel® Inspector**
   - Fix for suppression file usage when run in command line mode.
   - Added support for C++11 synchronization primitives during threading analysis.
   - Fixes for analyzing MPI applications
   - Variable name detection for threading analysis (global, static and stack variables)

10. **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.

11. New Usability Features

- **Online Installer**
  Download for later installation on the same or another computer is now available.
- **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™ Amplifier 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
  `c:\Program Files (x86)\IntelSWTools\eclipse_mars\`

Intel® System Studio 2016 Update 3

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)

- BLAS:
  - Improved small matrix [S,D]GEMM performance on Intel AVX2
  - Improved [C,Z]GEMV, [C,Z]TRMV, and [C,Z]TRSV performance on Intel AVX2
- LAPACK:
  - 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 (?GGHD3)
        - Multiplication of general matrix by a unitary/orthogonal matrix possessing 2x2 structure ([DS]ORM22/[CZ]UNM22)
    - Improved check of parameters for correctness in all LAPACK routines to enhance security
- SCALAPACK:
    - Improved hybrid (MPI + OpenMP*) performance of ScaLAPACK/PBLAS by increasing default block size returned by pilaenv
- SparseBlas:
    - Added examples that cover spmm and spmmd functionality
    - Improved performance of parallel mkl_sparse_d_mv for general BSR matrices on Intel AVX2
    - Parallel Direct Sparse Solver for Clusters:
    - Improved performance of solving step for small matrices (less than 10000 elements)
    - Added mkl_progress support in Parallel Direct sparse solver for Clusters and fixed mkl_progress in Intel MKL PARDISO
- Vector Mathematical Functions:
    - 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

3. 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.

4. 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.

5. **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

6. **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™ platforms.

7. **Intel® Inspector**
    - No update vs. Update 2

8. **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® TBB threading support for all and OpenMP* for some BLAS level-1 functions.

3. Intel® Integrated PP (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)
    - Support for Eclipse* 4.4 (Luna) integration with the 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 Analyzers (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

7. Intel® Inspector
    - Support for the Microsoft Windows* 10 OS support
    - Support for Microsoft* Visual Studio* 2015 IDE integration

Intel® System Studio 2016 Update 1

1. Intel® C++ Compiler:
    - Support for Windows* 10
    - Support for Microsoft* Visual Studio* 2015

2. Intel® Energy Profiler (SoC Watch):

- Intel® System Studio 2016 Update 2 – Installation Guide and Release Notes – Windows* Host 8
- 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 Product Contents and Cross Reference

The following table provides a reference of actual versions of the Intel® Software Development Tools present in each edition of Intel® System Studio 2017 for Windows*.

This Release Notes and Installation Guide document is included in the Intel® System Studio 2017 **Professional and Ultimate Editions**. Please note that the documentation related to the Intel® System Debugger is not related to the Professional Edition as this doesn't include the system debugger.

Component	Version	Composer Edition	Professional Edition	Ultimate Edition
Intel® C++ Compiler	17.0 Update 4	X	X	X
Intel® Integrated Performance Primitives (Intel® IPP)	2017 Update 3	X	X	X
Intel® Math Kernel Library (Intel® MKL)	2017 Update 3	X	X	X
Intel® Threading Building Blocks (Intel® TBB)	2017 Update 6	X	X	X
GNU* GDB and source	7.10	X	X	X
Integration into Eclipse* software (with optional installations of a pre-configured Eclipse* 4.5 Mars package and a JAVA Runtime Environment (JRE) 1.8 64-bit package)	2017	X	X	X
Integration into Wind River* Linux and Wind River* Workbench (workbench must be pre-installed)	2017	X	X	X
Integration into Android* NDK (NDK must be pre-installed)	2017	X	X	X
Intel® Inspector	2017 Update 3		X	X
Intel® VTune™ Amplifier for Systems with Intel® Energy Profiler	2017 Update 3		X	X
Intel® Graphics Performance Analyzers	2017 R1		X	X
GNU* GDB and source	7.10		X	X
Intel® System Debugger (system debug & trace)	2017			X
OpenOCD* library and source	0.8.0			X

3.1 Intel® Software Manager

The Intel® Software Manager, automatically installed with the Intel® System Studio product, is a graphical tool to provide a simplified delivery mechanism for product updates, current license status and news on all installed Intel Software Development Products.

The software manager is automatically started after product installation. It can also be manually started as

- *`Intel® Software Manager`* application (Windows* 8.x/10)

or from

- *`Start / Intel System Studio 2017 / Intel Software Manager`* (Windows* 7).

The software manager resides in the Windows System Tray and can be stopped and closed from there again (*Open the System Tray, right-click on the Intel® Software Manager Icon and click 'Exit')*.

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 Technical Support and Documentation

4.1 Technical 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 Online Service Center webpage, search for the product **Intel System Studio** and submit your support request.

Additionally you may submit questions and browse issues in the Intel® System Studio User Forum.

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

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

4.2 Documentation Locations

From the documentation directory

```
<INSTALLDIR>\documentation_2017\en
```

where the default installation directory of `<INSTALLDIR>` is:

```
c:\Program Files (x86)\IntelSWTools\documentation_2017
```

you can find the overall System Studio product release notes and installation guide, an Intel System Studio User Guide as well as several Getting Started Guides for the installed components (according to the System Studio edition you have installed on your system).

The Getting Started Guides link to several user documentation and references and other useful information.

Note: Release Notes and documentation of the latest Intel® GPA 2017 R1 release are available online only, *see '4.3 Online Documentation '*

4.3 Online Documentation

The complete product documentation of Intel® System Studio as well as articles, training material and other useful links can be found online at:

- https://software.intel.com/en-us/intel-system-studio-support/documentation

- https://software.intel.com/en-us/articles/intel-system-studio-tutorials

Documentation of the latest Intel® GPA 2017 R1 release is available at:

- https://software.intel.com/en-us/articles/intel-gpa-release-notes
- https://software.intel.com/en-us/articles/intel-gpa-online-help
- https://software.intel.com/en-us/gpa/documentation

5 System Requirements

5.1 Supported Host Platforms

One of the following 64-bit Windows* 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 (all 64-bit only)

Individual Intel® System Studio 2017 components may support additional distributions. Please refer to the individual components release notes as listed under chapter *4.2 Documentation Locations* for details.

5.2 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 `C:\Program Files (x86)\eclipse\`.

The prerequisites for successful Eclipse* integration are:

1. Eclipse* 4.4 (Luna) – Eclipse* 4.6 (Neon)
2. Java Runtime Environment (JRE) version 7.0 (also called 1.7) or later – recommended version is JRE 8 (1.8).

5.3 Wind River* Linux Build Environment Integration

Cross-build for Wind River Linux* targets currently requires an existing Wind River* Linux 6, 7 or 8 (Native support) installation on the host system that the compiler can integrate into. The System Studio installer provides an option for an easy integration of the compiler into the host Wind River Linux* build environment.

5.4 Android* NDK Integration

Cross-build for Android* targets requires an existing Android* NDK installed on the host system. The System Studio installer provides an option for an easy integration of the compiler into the NDK.

5.5 Support for Intel® C++ Compiler for Android* in Microsoft Visual Studio* 2015

The Intel® System Studio 2017 supports native Android development using the Intel® C++ Compiler with Microsoft Visual Studio* 2015 (VS2015) IDE. In order to use the Intel® C++ Compiler for Android within VS2015 IDE, it is required to customize the VS2015 installation. If the required modifications are not done before System Studio installation, the compiler can be used from command line and within Eclipse*.

To enable support for Intel® C++ Compiler for Android* in Microsoft Visual Studio* 2015 please follow the steps described in this document: https://software.intel.com/en-us/articles/intel-c-compiler-for-android-integration-into-microsoft-visual-studio-2015.

5.6 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, 1.8, 2.0 based environment
- CE Linux* PR35 based environment
- Tizen* IVI 3.x
- Wind River* Linux* 6, 7, 8 based environment
- Android* 5.0, 5.1, 6.0

Note:

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

6 Installation Notes

6.1 Installing the Tool Suite

Please refer to section *'5 System Requirements'* to check the prerequisites for installing the Intel® System Studio 2017.

6.1.1 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 chose 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:

- Double-click the downloaded online installer agent `system_studio_2017.x.xxx_online.exe`

or

- Double-click the downloaded full package offline installer `system_studio_2017.x.xxx.exe`

6.1.2 Activating the Product

During installation of the Intel® System Studio 2017 an activation dialog pops up. The following options are available:

- **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; see also How to find serial number)
- **Alternative activations**
  - **Offline activation** by using a license file .lic which must be available on the install machine (no internet connection required; see also Offline activation of Intel Software Development Products)
  - **Use a license manager** (Intel® Software License Manager must be running on the license server and connection to the server from the client machine must be established, no internet connection required; see also Intel Software License Manager Users Guide)

6.1.3 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.

6.2 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 got damaged with the product). You can choose one of the following

- Start the Windows* system's Control Panel, choose *'Uninstall a program' / Intel System Studio 2017*

or

- Run the `c:\Users\<UUID>\Downloads\Intel\system_studio_2017.x.xxx\setup.exe`

and choose the desired option, **'Modify'**, **'Repair'** or **'Remove'**

6.3 Installation Directory Structure

The default base installation, in the following referred to as `<INSTALLDIR>` directory is:

```
C:\Program Files (x86)\IntelSWTools
```

Intel® System Studio 2017 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>.`

IMPORTANT NOTES:

- **The destination folder <INSTALLDIR> MUST NOT exceed the length of 35 characters. If you decided to specify a customized destination folder, please take care to not exceed this 35-characters limitation.**
- **The destination folder <INSTALLDIR> is fixed to what you specified it the first time when installing an Intel® Software Development Product. Any later update installation or installation of a new major version is bound to the existing <INSTALLDIR>.**

The Intel® System Studio 2017 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 respectively, Intel® Graphics Performance Analyzers and Wind River Linux* development environment integration as follows:

- `<INSTALLDIR>\system_studio_2017.x.xxx\compilers_and_libraries_2017`
- `<INSTALLDIR>\system_studio_2017.x.xxx\debugger`

- `<INSTALLDIR>\system_studio_2017.x.xxx\documentation_2017`
- `<INSTALLDIR>\system_studio_2017.x.xxx\ide_support_2017`
- `<INSTALLDIR>\system_studio_2017.x.xxx\licensing`
- `<INSTALLDIR>\system_studio_2017.x.xxx\samples_2017`
- `<INSTALLDIR>\system_studio_2017.x.xxx\Targets`
- `<INSTALLDIR>\system_studio_2017.x.xxx\VTune Amplifier for Systems`
- `<INSTALLDIR>\system_studio_2017.x.xxx\wr-iss-2017`

Intel® Software Development Products Common Components Directory with Links from System Studio:

- `<INSTALLDIR>\compilers_and_libraries`
- `<INSTALLDIR>\compilers_and_libraries_2017`
- `<INSTALLDIR>\compilers_and_libraries_2017.x.xxx`
- `<INSTALLDIR>\debugger_2017`
- `<INSTALLDIR>\documentation_2017`
- `<INSTALLDIR>\eclipse_mars`
- `<INSTALLDIR>\GPA_2017`
- `<INSTALLDIR>\ide_support_2017`
- `<INSTALLDIR>\Inspector`
- `<INSTALLDIR>\Inspector 2017`
- `<INSTALLDIR>\samples_2017`
- `<INSTALLDIR>\System Debugger 2017`
- `<INSTALLDIR>\system_studio_2017.x.xxx`
- `<INSTALLDIR>\VTune Amplifier 2017 for Systems`
- `<INSTALLDIR>\VTune Amplifier for Systems`

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 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 `<INSTALLDIR>\system_studio_2017.x.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.

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.

6.4 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

- 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
- 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`.

- 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.

- For the Intel® VTune Amplifier Sampling Enabling Product (SEP) pick `../system_studio_target/vtune_amplifier_2017_for_systems_target/linux/vtune_amplifier_target_sep_x86[_64].tgz`

- For the Intel® VTune Amplifier for Systems target package pick `../system_studio_target/vtune_amplifier_2017_for_systems_target/linux/vtune_amplifier_target_x86[_64].tgz`

- 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.

- For the Intel® Inspector follow the instructions in `../system_studio_target/inspector_2017_target_zip/documentation/en/Release_Notes_Inspector_Linux.pdf`

6.4.1 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 `../system_studio_target/inspector_2017_target_zip/` 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.

6.4.2 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_2017_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.

6.4.3 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`

6.4.4 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.

6.4.5 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_for_systems_target/linux/vtune_amplifier_target_sep_x86[_64].tgz`

2. After unpacking this zip file follow the instructions in
   `../vtune_amplifier_2017_for_systems.x.x.xxxxxx/sepdk/src/README.txt`

6.4.6 Intel® Integrated Performance Primitives runtime shared object installation

If you are using dynamic linking when using the Intel® Integrated Performance Primitives (Intel® IPP), 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.

6.4.7 Intel® Math Kernel Library runtime shared object installation

If you are using dynamic linking when using the Intel® Math Kernel Library (Intel® MKL), 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.

6.4.8 Intel® Threading Building Blocks runtime shared object installation

If you are using dynamic linking when using the Intel® Threading Building Blocks (Intel® TBB), 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.

6.4.9 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.

6.5 Eclipse* IDE Integration

6.5.1 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

```
C:\Program Files (x86)\IntelSWTools\eclipse_mars
```

or skip this installation and point the installer to an existing Elipse* directory (usually

```
C:\Program Files (x86)\eclipse)
```

or you can skip the integration of Intel® System Studio components into Eclipse* completely.

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

1. Eclipse* 4.4 (Luna), Eclipse* 4.5 (Mars) or Eclipse* 4.6 (Neon)
2. Java Runtime Environment (JRE) version 7.0 (also called 1.7) or later – recommended version is JRE 8 (1.8).

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.

6.5.2 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.:

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

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

```
c:\Program Files (x86)\IntelSWTools\eclipse_mars\eclipse
```

or

```
c:\Program Files (x86)\eclipse\eclipse
```

6.5.3 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® C++ Compiler documentation at
`<INSTALLDIR>\documentation_2017\en\compiler_c\iss2017`

6.5.4 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

6.5.5 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/mars1) 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>/<target_os>/bin/gdbserver`
   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`.

6.5.6 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 `<INSTALLDIR>\intel_system_debugger_plugins`, where the default for `ISS_INSTALL_PATH` is `<INSTALLDIR>\system_studio_2017.x.xx\ide_support_2017\eclipse\`
4. Click on the `com.intel.iss.ide.integration.site.all-1.0.0-SNAPSHOT.zip` file.

6.6 Wind River* Workbench* IDE Integration

6.6.1 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.

6.6.2 Installation

Intel® System Studio provides Wind River* Linux* build environment integration and platform recipes for Intel® C++ Compiler, Intel® Integrated 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. 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.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.

6.6.3 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.bat <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® 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.

6.6.4 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.bat`

6.7 Installing the Intel® ITP-XDP3 In-Target 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.

6.8 Ordering JTAG Probe / USB-Cable for the Intel® System Debugger

1. To order the Intel ITP-XDP3 In-target 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: http://designintools.intel.com/product_p/itpxdpsvt.htm (Intel SVT Closed Chassis Adapter)
3. To order an USB3 cable for probe-less debugging on system which support DCI/DbC, please go to one of the following:
   - https://designintools.intel.com/product_p/itpdciamam1m.htm (Intel SVT DCI DbC2/3 Debug Cable 1 Meter)
     - This 1.0 meter USB debug cable enables debug of platforms supporting Silicon View Technology, using DCI-USB2 or DCI-USB3. This is similar to a standard USB A male to A male cable, except that the VBus signal is removed. USB 3.0 compliant cable.
   - http://www.datapro.net/products/usb-3-0-super-speed-a-a-debugging-cable.html (USB 3.0 Super-Speed A/A Debugging Cable)
     - Make sure to order a cable that is as short as possible!

7 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

7.1 General Known Issues and Limitations

7.1.1 <INSTALLDIR> must be Limited to 35 Characters

The length of the destination installation folder (in this document also referred to as `<INSTALLDIR>`) MUST NOT exceed the length of 35 characters.

The default destination folder is

```
c:\Program Files (x86)\IntelSWTools
```

which is exactly 35 characters. If you decided to specify a customized destination folder, please take care to not exceed this 35-characters limitation.

7.1.2 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 documentation from `<INSTALLDIR>\system_studio_2017.x.xxx\documentation_2017\en\` may not work.

Please refer directly to the

```
<INSTALLDIR>\VTune Amplifier 2017 for Systems\documentation\en\
```

or

```
<INSTALLDIR>\Inspector 2017\documentation\en\
```

7.1.3 Running online-installer behind proxy server may fail

Running online-installer behind proxy server may produce the error: "Connection to the IRC site cannot be established". If the proxy settings issue cannot be resolved, you need to download the full package (from a different computer) and run the installer from the downloaded .exe file.

7.1.4 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.

7.2 Intel® Energy Profiler

7.2.1 \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`
```

7.3 Intel® VTune™ Amplifier Usage with Yocto Project*

7.3.1 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.

7.3.2 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.

7.3.3 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
```

7.4 Intel® System Debugger

7.4.1 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.

7.4.2 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`.

7.5 GDB* - GNU* Project Debugger

7.5.1 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.

7.6 Intel® Integrated Performance Primitives

7.6.1 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.

7.7 Intel® C++ Compiler

7.7.1 “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.

7.7.2 Add “/lib64->/lib” symlink on Intel® Joule™ Platforms

To avoid link errors you need to add a symlink `/lib64->/lib` as `ld-linux-x86-64.so.2` is located in `/lib` on the target OS.

8 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.

9 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