Intel® Advisor 2021.3 Release

Release Notes

19 July 2021

Customer Support

For technical support, including answers to questions not addressed in this product, visit the technical support forum, FAQs, and other support information at:

- Intel Advisor Forum
- General oneAPI Support
- Intel Advisor Home Page

Please remember to register your product at https://registrationcenter.intel.com/ by providing your email address. Registration entitles you to free technical support, product updates and upgrades for the duration of the support term. It also helps Intel recognize you as a valued customer in the support forum.

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

Contents

1 Introduction

Intel® Advisor is a design and analysis tool to achieve high application performance through efficient threading, vectorization and memory use, and GPU offload on current and future Intel hardware.

- **Offload Advisor** helps to get your code ready for efficient GPU offload even before you have the hardware. Identify offload opportunities, quantify potential speedup, locate bottlenecks, estimate data transfer costs, and get guidance on how to optimize.
- **Vectorization Advisor** is a vectorization optimization tool that lets you identify high-impact, under-optimized loops, what is blocking vectorization, and where it is safe to force vectorization. It also provides code-specific how-can-I-fix-this-issue recommendations.
- **Roofline Analysis** visualizes actual performance of CPU and GPU kernels against hardware-imposed performance ceilings (rooflines). It provides insights into where the bottlenecks are, which loops are worth optimizing for performance, what are the likely causes of bottlenecks and what should be the next optimization steps.
- **Threading Advisor** is a fast-track threading design and prototyping tool that lets you analyze, design, tune, and check threading design options without disrupting your normal development.

- **Flow Graph Analyzer** is a visual prototyping tool that lets you represent and analyze performance for applications that use the Intel® Threading Building Blocks (Intel® TBB) flow graph interfaces.

This document provides system requirements, issues and limitations, and legal information.

To learn more about this product, see:

- New features listed in the What's New section below.
- The help document at https://software.intel.com/en-us/advisor-user-guide

2 What's New

Intel® Advisor 2021.3 Release

- **Offload Modeling:**
  - **Introduced GPU-to-GPU performance modeling (feature preview)**
    The Offload Modeling perspective introduces a new GPU-to-GPU performance model*. With this model, you can* analyze your Data Parallel C++ (DPC++), OpenMP* target, or OpenCL™ application running on a graphics processing unit (GPU) and model its performance on a different GPU platform. Use this workflow to understand how you can improve your application performance and check if you can get a higher speedup if you offload the application to a different GPU platform. This feature is currently available via command line only.

    Currently, you can model performance for the following target devices: **Intel® Iris® X^e**, **Intel® Iris® X^e MAX**, **Intel® Iris® Plus** Graphics, **Intel® Iris®** Graphics **550**, **Intel® Iris® Pro** Graphics **580**, **Intel® HD** Graphics **530**.

    The GPU-to-GPU performance modeling is based on the following:

    - **Compute throughput model** estimates time by compute throughput based on GPU kernel instruction mix with respect to GPU compute throughput capabilities and workload decomposition. The model consists of the following:
      - Compute pipeline model: Each pipeline sequentially processes compute instructions of specific types; all pipelines work in parallel. Pipeline configuration reflects hardware architecture and is stored in a configuration file. Pipeline is implemented for all supported GPU devices.
      - Workload decomposition model: global work size, local work size, SIMD width. Performance model estimates the number of parallel threads, thread and execution unit (EU) occupancy and searches for an optimal workload decomposition to minimize kernel execution time. This is a special operation mode to model the same workload decomposition as on a baseline.
    - **Memory throughput model** estimates cache and memory traffic based on a target GPU configuration. It implements a simple SLM model, considering that SLM traffic does not change from a baseline to a target, and a simple eDRAM model if the EDRAM is configured as an additional cache level (**Intel® Iris® Graphics 550**, **Intel® Iris® Pro Graphics 580**). Memory subsystem configuration specified for all the supported devices. Based on cache/memory traffic, the model also estimates time by cache/memory bandwidth.
    - **Memory latency model** estimates latency of read memory instructions based on the number of the instructions in the kernel. The model considers a memory latency is overlapped between

threads, but memory latency and compute throughput are not overlapped. Memory latency values are gathered from b-spec and stored into all device configuration files.

  - **Atomic throughput model** estimates time by atomic throughput based on hardware counter of atomic accesses on the baseline device. The model assumes ratio of atomic access conflicts is fixed, but you can change it.
  - **Data transfer model** estimated offload overhead for transferring data between host and GPU devices. The model tracks memory objects on the host device using instrumentation of runtime calls and uses the memory object sizes and transfer directions of memory objects to estimate the data transfers.
    - Estimates separately host-to-device and device-to-host traffic and taxes
    - Supports different PCIe link bandwidth
    - Models memory object transfer overhead: transferring many small objects usually requires more time than transferring a single object of the same size

- **Added recommendations for effectively offloading your code from CPU to GPU:**
  The Offload Modeling perspective introduces recommendations for offloading code regions to GPU, performance bottleneck analytics and actionable recommendations to resolve them when you offload your code from a CPU to a GPU.

  The recommendations are reported in a new **Recommendations** pane in the Accelerated Regions report. For code regions *recommended for offloading*, the pane reports the modeled performance summary, including the speedup on the target GPU device and programming models that you can use to offload the region (Data Parallel C++ and OpenMP*).

  Each recommendation shows examples of offloaded code:

    - When collapsed, the code snippet highlights the required DPC++ and OpenMP constructs.
    - When expanded, the code snippet provides the complete algorithm implementation

  For code regions *not recommended for offloading*, the pane reports reasons why the region is not expected to have high speedup, suggesting you refactoring the code.

- **Added a recommendation for DPC++/OpenMP reduction pattern optimization:**
  The Offload Modeling perspective recognizes the *reduction patterns* used in the code regions recommended for offloading.

  Ineffective implementation of the reduction patterns might decrease performance, so Intel Advisor can warn you about it. It suggests you how to optimize the pattern and rearrange the code to better use GPU resources or minimize the bottleneck impact.

  Each recommendation shows examples of more effective implementations of the reduction using a division of array (global size) or work groups (local size):

    - When collapsed, the code snippet highlights the required DPC++ and OpenMP constructs.
    - When expanded, the code snippet provides the complete algorithm implementation.

- **Added a recommendation for algorithmic constraints optimization:**
  The Offload Modeling perspective recognizes the *algorithmic constraints* in the code regions recommended for offloading. In some cases, the number of iterations in the main loop of the kernel is not enough to fully use the GPU device capabilities.

In this recommendations, Intel Advisors shows the measured number of iterations, percentage of the thread slot utilization and suggests adding more parallel work, for example, to use a larger dataset or collapse nested loop.

- **GPU Roofline**

  - **Extended the category of potential integer operations (INTOP) with logical operations:**

    When measuring the number of integer operations for the GPU Roofline, Intel Advisor counts logical operations, such as AND, OR, XOR, as potential integer operations.

    Some applications might contain a lot of logical operations, which take time to execute on a CPU, but do not affect the number of integer operations. This might decrease your application performance. As Intel Advisor counts logical operations as potential INTOP, it reflects the actual performance of the profiled application on the GPU Roofline chart showing all the hotspots with logical operations closer to a performance boundary.

    For more information about operations counted for the GPU Roofline, see Examine Bottlenecks on GPU Roofline Chart.

  - **Added GPU Roofline interpretation hints to the kernel details:**

    Intel Advisor provides hints for memory-bound codes to increase application performance and remove memory subsystems bottlenecks:

    - If a kernel is bounded by L3 cache, Intel Advisor analyses L3 cache line utilization. It is calculated as a ratio of data transferred to an execution units and data transferred to L3 cache. Intel Advisor gives a new hint to improve memory access pattern if L3 cache line utilization is bad.
    - If a kernel has high external memory utilization, Intel Advisor suggests implementing a cache-blocking technique to improve cache locality and increase code performance.

    See Examine Bottlenecks on GPU Roofline Chart for details.

  - **Improved memory metric grid for the GPU Roofline:**

    In the GPU Roofline report, memory columns of the GPU grid now provide better and clear view of memory metrics:

    - Memory metrics are grouped by memory subsystem. By default, you can see total data transferred and total bandwidth metrics. If you want to see more details for one of the memory levels, you can expand the columns and see additional metrics, such as data read and data written to/from host and target, read and write bandwidth.
    - A new CARM column group includes CARM traffic and L3 cache line utilization metrics. Review these metrics to analyze memory access pattern of GPU kernels.

      See Accelerator Metrics for details.

  - **Introduced per-kernel instances for the GPU Roofline:**

    The GPU Roofline Insights perspective introduces a new kernel visualization feature that breaks down a kernel into instances grouped by workload parameters (global and local sizes).

    If the kernel was executed with different workloads or work-groups, you can compare performance characteristics for different executions.

    The feature is shown in the following panes of the GPU Roofline report:

- In the GPU grid, you can expand a source kernel to see its sub-rows. Each sub-row is group of kernel instances executed with the same kernel properties.
    - In the GPU Roofline chart, if the kernel was executed with different properties, it has a plus icon near the kernel dot. The parent dot corresponds to the source compute task. Click the plus icon to expand it and see kernel instances that visualize performance dependency on workload parameters.
- Select the source compute task or the instance task from the grid or from the chart to see the details metrics in the Details pane.

See Examine Bottlenecks on GPU Roofline Chart for details.

Intel® Advisor 2021.2 Release

- **Expanded hardware support:**
    - Vectorization and Roofline analysis is supported for **11th Gen Intel Core processors (code-named "Tiger Lake"), 3rd Gen Intel® Xeon® Scalable Processors (code-named "Ice Lake") and 2nd Generation Intel® Xeon® Scalable Processors (code-named "Cooper Lake")** providing the same functionality as on other supported platforms. For the Roofline analysis, floating point and Integer operations metrics are adjusted for the instruction set architecture extensions, including the bfloat16 data type, avx512_bf16 and vbmi2 instructions.

- **Introduced GPU Data Transfer estimation with data reuse:**
    - To get more accurate estimations of data transfer costs for GPU performance modeling, use a new data transfer reuse analysis introduced for the Offload Modeling perspective.
    - Data transfer reuse analysis detects groups of regions that can reuse the same memory objects on GPU, improving the data transfer tax estimation accuracy. It also shows which kernels can benefit from data reuse and how it impacts application performance.
    - You can enable the data reuse analysis for Performance Modeling from Intel Advisor GUI or from command line interface. With the analysis enabled, the estimated data transfer metrics are reported with and without data reuse. See Accelerator Metrics for details.

- **Improved user interface:**
    - *New Source view for the Offload Modeling and GPU Roofline Insights perspectives:*

      Use the full-screen Source view with syntax highlighting located in a separate tab to explore application source code and related metrics, such as elapsed time. For the GPU Roofline Insights perspective, the Source view also includes the Assembler view, which you can view side-by-side with the source.

      To switch to the Source view, double-click a kernel from the main report.

    - *New Details pane with in-depth GPU kernel analytics for the GPU Roofline Insights perspective:*

      Use the Details pane in the GPU Roofline Regions report to review in-depth kernel execution metrics for a single kernel, such as execution time on GPU, work size and SIMD width, a single-kernel Roofline highlighting the distance to the nearest roof (performance limit), floating-point and integer operation summary, memory and cache bandwidth, EU occupancy, and instruction mix summary.

Intel® Advisor 2021.1 Initial Release

- **Data Parallel C++ (DPC++):**

  - Implemented support for Data Parallel C++ (DPC++) code performance profiling on CPU and GPU targets.
  - Implemented support for oneAPI Level Zero specification for DPC++ applications.
- **Usability:**
  - Introduced a new and improved Intel Advisor user interface (UI) that includes:
    - New look-and-feel for multiple tabs and panes, for example, **Workflow** pane and **Toolbars**
    - **Offload Modeling** and **GPU Roofline** workflows integrated in GUI
    - New notion of *perspective*, which is a complete analysis workflow that you can customize to manage accuracy and overhead trade-off. Each perspective collects performance data but processes and presents it differently so that you could look at it from different points of view depending on your goal. Intel Advisor includes Offload Modeling, GPU Roofline Insights, Vectorization and Code Insights, CPU / Memory Roofline Insights, and Threading perspectives.

    > **NOTE**
    > To switch back to the old UI, set the `ADVISOR_EXPERIMANTAL=advixe_gui`.

  - Renamed executables and environment scripts:
    - `advixe-cl` is renamed to `advisor`.
    - `advixe-gui` is renamed to `advisor-gui`.
    - `advixe-python` is renamed to `advisor-python`.
    - `advixe-vars.[c]sh` and `advixe-vars.bat` are renamed to `advisor-vars.[c]sh` and `advisor-vars.bat` respectively.

    > **NOTE**
    > The previous command line interface and executables are supported for backward compatibility.

- **Offload Modeling:**
  - Introduced the Offload Modeling perspective (previously known as Offload Advisor) that you can use to prepare your code for efficient GPU offload even before you have a hardware. Identify parts of code can be efficiently offloaded to a target device, estimate potential speedup, and locate bottlenecks.
  - Introduced data transfer analysis as an addition to the Offload Modeling perspective. The analysis reports *data transfer costs* estimated for offloading to a target device, estimated *amount of memory* your application uses per memory level, and *guidance* for data transfer optimizations.
  - Introduced strategies to manage kernel invocation taxes (or kernel launch taxes) when modeling performance: do not hide invocation taxes, hide all invocation taxes except the first one, hide a part of invocation taxes. For more information, see Manage Invocation Taxes.
  - Added support for modeling application performance for the Intel® Iris® Xe MAX graphics.
- **GPU Profiling and Roofline:**
  - Introduced a GPU Roofline Insights perspective. GPU Roofline visualizes actual performance of GPU kernels against hardware-imposed performance limitations. Use it to identify the main limiting factor of your application performance and get recommendations for effective memory vs. compute optimization. GPU Roofline report supports float and integer data types and reports metrics for all memory levels.
  - Added support for profiling GPU workloads that run on the Intel® Iris® Xe MAX graphics and building GPU Roofline for them.

- Introduced Memory-Level Roofline feature (previously known as Integrated Roofline, tech preview feature). Memory-Level Roofline collects metrics for all memory levels and allows you to identify memory bottlenecks at different cache levels (L1, L2, L3 or DRAM).
- Added a limiting memory level roof to the Roofline guidance and recommendations, which improves recommendation accuracy.
- Added a single-kernel Roofline guidance for all memory levels with dots for multiple levels of a memory subsystem and limiting roof highlighting to the **Code Analytics** pane.

- **Flow Graph Analyzer:**
  - Implemented DPC++ support: You can profile DPC+ code on CPU on Linux* OS. The collector is only available on Linux OS, but you can view the data on any platform.
  - Added support for visualizing DPC++/SYCL asynchronous task-graphs and connecting the executions traces with graph nodes.
  - Added analytics for determining inefficiencies in thread start-up and join for DPC++ algorithms running on the CPU using cpu_selector.
  - Added rules to the Static Rule-check engine to determine issues with unnecessary copies during the creation of buffers, host pointer accessor usage in a loop, multiple build/compilations for the same kernel when invoked multiple times.

3 System Requirements

Supported Architectures and Terminology

Intel® Advisor supports the following architectures:

- **IA-32 Architecture** refers to systems based on 32-bit processors generally compatible with the Intel Pentium® processors (for example, Intel® Pentium® 4 processor or Intel® Xeon® processor), or processors from other manufacturers supporting the same instruction set, running a 32-bit operating system.
- **Intel® 64 Architecture** refers to systems based on IA-32 architecture processors which have 64-bit architectural extensions, for example, Intel® Core™ architecture processors, running a 64-bit operating system such as Microsoft Windows* 10 x64 or a Linux* "x86_64" variant. If the system is running a 32-bit operating system, then IA-32 architecture applies instead.

Hardware Requirements

- At least 2 GB of RAM
- 650 MB free disk space required for all product features and all architectures.
- A system based on an IA-32 or Intel® 64 architecture processor supporting the Intel® Streaming SIMD Extensions 2 (Intel® SSE2) instructions (Intel® Pentium 4 processor or later, or Intel® Xeon® Scalable Processors)
  - Since the Intel Advisor requires specific knowledge of assembly-level instructions, its analysis may not operate correctly if code contains non-Intel instructions. In this case, run the analysis with a target executable that contains only Intel® instructions. After you finish using the Intel Advisor, you can use the assembler or optimizing compiler options that provide the non-Intel instructions.
  - For the best experience, a multi-core or multi-processor system is recommended

- GPU Profiling and Roofline features are supported on Intel® processors with integrated GPU with Gen9, Gen11 architecture, and discrete Intel® Iris® Xe MAX GPU.

Software Requirements

NOTE: On Linux, the X.Org X11 libXss runtime library must be installed for the Intel Advisor GUI.

- Supported operating systems (embedded editions are not supported):
    - Offload Advisor performance projection is supported on:
        - Microsoft Windows* 10 operating system
        - Microsoft Windows Server* 2016 and 2019
        - Red Hat* Enterprise Linux* 7 and 8
        - CentOS* versions equivalent to Red Hat* Enterprise Linux* versions listed above
        - Fedora* 33 and 34 (pangox-compat package should be installed)
        - SUSE* Linux* Enterprise Server* 12 and 15
        - Debian* 9,10
        - Ubuntu* 18.04, 20.04 and 21.04
    - GPU Roofline analysis is supported on:
        - Microsoft Windows* 10 operating system
        - Red Hat* Enterprise Linux* 8
        - SUSE* Linux* Enterprise Server* 15
        - Ubuntu* 18.04, 20.04 and 21.04

      GPU profiling and Roofline on Microsoft Windows* systems is supported **only for 64-bit applications**.

      GPU profiling and Roofline is currently supported on **Linux* with kernel version 4.11 or higher**.
    - Viewer only: macOS* 10.15,11
- Supported Microsoft Visual Studio* (and Microsoft Visual C++* compiler) versions:
    - Microsoft Visual Studio* 2017
    - Microsoft Visual Studio* 2019
- Supported compilers:
    - Intel® C++ Compiler 15 or higher
    - GNU* C/C++ Compiler 3.4.6 and higher (Linux)
    - Intel® Fortran Compiler 15 or higher (Linux)
    - Intel® Visual Fortran Compiler 15 or higher (Windows)
- Application coding requirements
    - Supported programming languages:
        - Fortran
        - C
        - C++
    - Supported threading methodologies:
        - Intel® Threading Building Blocks (Intel® TBB)
        - Intel® Cilk™ Plus
        - OpenMP*
        - Windows and POSIX Threads (low-level)
- To view PDF documents, use a PDF reader, such as Adobe Reader*.

NOTE: C# and .NET support is deprecated in Advisor since 2021 version.

NOTE: Intel Advisor samples work with the Intel® Threading Building Blocks (Intel® TBB) by using the environment variable `TBBROOT`. Intel® TBB is included with the Intel® compiler and is available from http://software.intel.com/en-us/intel-tbb/. To build the Intel® TBB projects, you will need to define the `TBBROOT` environment variable. To set the `TBBROOT` environment variable, see the Intel Advisor Help or Getting Started Tutorial.

4 Where to Find the Release

If you have a qualifying license for this product (i.e. an unexpired free license or a paid license with active support at the time of the build date for this product release), you can download the installer by logging in to https://registrationcenter.intel.com/ and selecting the appropriate product.

You may need to create an account and/or register your product. For additional information please see:

- https://software.intel.com/content/www/us/en/develop/articles/oneapi-commercial-faq.html

This product can be also downloaded as part of one of the Intel® oneAPI Toolkits with forum support.

5 Installation Notes

For instructions on installing, updating or removing Intel® Advisor, see the appropriate installation guide for your operating system.

- For Linux*
- For Windows*
- For macOS*

Default Installation Folders

The default top-level installation folders for this product are:

- Windows: `C:\Program Files (x86)\Intel\oneAPI\Advisor\2021.3.0`
    - If you are installing on a system with a non-English language version of Windows OS, the name of the `Program Files` folder may be different. On Intel® 64 systems, the directory name is `Program Files (x86)` or the equivalent.
- Linux: `/opt/intel/oneapi/advisor/2021.3.0`
- macOS*: `/opt/intel/oneapi/advisor/2021.3.0`

In both cases, there is also a generic soft link to the most recently installed version.

Known Installation and Configuration Issues

On Windows, please close Microsoft Visual Studio* sessions before installing, modifying, or removing the Intel Advisor. Not doing so will affect Visual Studio integration and may result in obscure error messages.

Setting Up the Command Line

Before using Intel® Advisor on the command line, you will need to set up the environment:

- Windows: `<installation_dir>\advisor-vars.bat`
- Linux: `source <installation_dir>/advisor-vars.sh`
- macOS*: `source <installation_dir>/advisor-vars.sh`

> **NOTE:** To build the Intel Advisor sample applications, you may need to first define the `ADVISOR_2021_DIR` environment variable. Running the `advisor-vars.sh` script will do this automatically.

6 Known Issues

This section is quite long. To skip to operating system specific limitations, use these links:

- Windows* Specific Limitations
- Linux* Specific Limitations

Common Limitations

GPU Profiling: Not supported for applications using Intel oneAPI Math Kernel Library with OpenCL backend

FGA does not run on Mac OS 11

Apple* System Integrity Protection technology prevents Flow Graph Analyzer from working on Mac OS 11.2.3. Possible workaround is to disable System Integrity Protection service temporarily.

Offload Advisor Python scripts are sensitive to Python version

analyze.py, collect.py and run_oa.py scripts only work with Python* 3.6 and 3.7. If you don't have Python* 3.6 or 3.7 installed you can use internal python command available in Advisor directory.

For example: `advisor-python run_oa.py<project_directory>`

Offload Modelling may not work on profile collected with previous Advisor versions

If you run performance modeling on the results collected with a previous release of the Intel Advisor (including Beta versions), analyze.py script may report `"Error: Memory model cache hierarchy incompatible”` error. To work around the issue, delete the “perf_models” directory from the results directory and re-run analyze.py.

Option to select loops for markup by criteria select all loops despite criteria

For `--select` option markup criteria select all loops instead of intended by corresponding markup, it may cause very high overhead. To work around this problem, select loops by ID or source location.

WRONG VERSION OF LIBMD.SO LIBRARY MAY LEAD TO MISSING ELAPSED TIME, MEMORY BANDWIDTH AND EXECUTION UNITS METRICS IN GPU PROFILING RESULTS

Wrong version of libmd.so library might cause missing elapsed time, memory bandwidth and execution units metrics in GPU profiling results. To work around this problem, remove libmd* and libmd-devel packages from the system. Advisor will make use of the version provided with the product.

SUITABILITY TOOL MAY NOT ACCURATELY MEASURE VERY SHORT LOOPS

The Suitability tool cannot accurately measure and analyze the behavior of loops with very small task times (on the order of sub-microseconds) if those tasks are annotated with the `TASK_BEGIN`/`TASK_END` annotations. If possible, use the `ITERATION_TASK` annotation instead.

SUITABILITY REPORT COMBINES LOCK DATA

The Suitability Report combines all lock data into a single lock and may not have source line information for `LOCK_ACQUIRE` and `LOCK_RELEASE` annotations.

HUGE SOURCE FILES MAY NOT BE COMPLETELY PARSED FOR ANNOTATIONS

If your sources include huge source files that contain annotations, be aware that only the first 8 MB of each file will be parsed for annotations. If not all of your annotations are being parsed in such huge source files, consider breaking that source file into several source files each less than 8MB.

NON-RETURNING FUNCTIONS MAY CAUSE INCORRECT LOOP REPORTS

The Survey Report may incorrectly report loops if a function contains calls to a non-returning function such as the `exit()` function.

LOOPS IN FUNCTIONS WITHOUT DEBUG INFORMATION MAY NOT BE RECOGNIZED AS LOOPS

Loops within a function that does not contain debug information may not be reported as a loop in the Survey Report. For the suggested build settings, see the help topics under **Choosing and Building a Target > Build Settings for C/C++ Applications** or **Build Settings for Fortran Applications**. After you fix the build settings to specify debug information, rebuild the target and run Survey analysis again.

SURVEY MAY REPORT NO SIGNIFICANT HOTSPOTS

After analyzing the specified target executable, the Survey tool may report that no significant hotspot areas were detected. In this case, view the collected result using the Survey Report window. The suggested build settings are described in the help topics under **Choosing and Building a Target > Build Settings for C/C++ Applications** or **Build Settings for Fortran Applications**.

INTEL® ADVISOR MAY FAIL TO CORRELATE DATA WITH SOURCE CODE

If debug information correlating an instruction to source code is not available to an Intel® Advisor tool, the tool cannot correlate collected data with source code. In this case it will show an empty field, question mark, or broken icon. Source correlation may not be available for many different reasons, including:

- Omitting debug information options when building the target executable - see the help topics under Choosing and **Building a Target > Build Settings for C/C++ Applications** or **Build Settings for Fortran Applications**.
- A compiler error.
- The compiler not generating debug information for this specific source line or source file.

- The linker not copying the debug information into the debug information database.
- The debug information database not being found by the finalization step in the Intel Advisor, because it was not moved along with the executable to the location where the Intel Advisor is being run.
- When using the Intel Advisor GUI, insufficient project properties were specified for the Binary/Symbol Search tab. See https://software.intel.com/en-us/advisor-user-guide-binary-symbol-search-tab.

Windows* Specific Limitations

USER MAY NOTICE SLOWDOWN IN ONEAPI CODES EXECUTED ON CPU HOST ON WINDOWS* OS COMPILED WITH DEBUG INFO. SET THE BELOW ENVIRONMENT VARIABLES TO AVOID SLOWNESS OF PURE APPLICATION RUNTIME WHEN BUILT WITH DEBUG INFO.

CL_CONFIG_USE_NATIVE_DEBUGGER=0

CL_CONFIG_CPU_NO_DBG_CBK=1

ADVISOR SETS THESE AUTOMATICALLY AND HENCE NO ACTION REQUIRED WHILE RUNNING ADVISOR ANALYSIS.

ADVISOR MAY NOT COLLECT DATA PROPERLY FOR BINARIES LINKED WITH /DEBUG OR /DEBUG:FASTLINK OPTION ON WINDOWS.

Use /DEBUG:FULL option to avoid this issue

INTEL ADVISOR EXTENSION COULD BE DISABLED BY DEFAULT IN TOOLBAR OF MICROSOFT VISUAL STUDIO* 2019

To enable Intel Advisor in Microsoft Visual Studio* 2019 right click on toolbar and enable Advisor extension from drop-down list.

DEFAULT PROFILING FLOW FOR CPU DPC++ AND OPENMP CPU OFFLOAD DOES NOT WORK ON WINDOWS

Default and modern profiling flow for CPU DPC++ and OpenMP CPU Offload does not work on Windows yet. To enabled legacy mode set environment variable `INTEL_JIT_BACKWARD_COMPATIBILITY=1`.

INTEL ADVISOR MAY NOT MANUALLY INTEGRATE INTO MICROSOFT VISUAL STUDIO* 2017

Intel Advisor integrates into Microsoft Visual Studio 2017 only using the official Intel Advisor installation. Any other ways like `advixe-vsreg.exe` tool or manual launching `.vsix` package will not work.

INTEL ADVISOR MAY HANG IF GUI OR MICROSOFT VISUAL STUDIO* IS CLOSED WHILE EXPANDING A SURVEY TREE

Do not close the Intel Advisor GUI or Microsoft Visual Studio* software while expanding the Survey tree. Doing so may cause the product to crash or hang.

INTEL ADVISOR MAY REPORT INCORRECT DATA IF LOCAL CHARACTER SETS ARE USED

Intel Advisor may present incorrect or incomplete information if local character sets are used. For general character set support, use Unicode.

MICROSOFT VISUAL STUDIO MAY DISPLAY A WARNING MESSAGE WHEN OPENING INTEL ADVISOR SAMPLES

When opening the Intel Advisor samples, Microsoft Visual Studio software will display a warning message related to Intel® Cilk Plus projects or list them as unsupported when the Intel® C++ Compiler is not installed. This message may be ignored. However, Intel® Cilk™ Plus projects will not compile without an Intel® C++ Compiler with Intel® Cilk Plus support, such as the Intel® Parallel Studio XE (or similar Intel® software suite).

ANTIVIRUS SOFTWARE MAY PREVENT INTEL ADVISOR FROM INSTRUMENTING CODE

Virus checking can sometimes interfere with our ability to instrument code. Currently there is no workaround, except turning off the virus checker.

Linux* Specific Limitations

LIBNSS3.SO AND LIBNSSUTIL3.SO LIBRARIES ARE REQUIRED TO BE PRE-INSTALLED ON LINUX SYSTEM FOR ADVISOR GUI TO WORK PROPERLY.

TARGET APPLICATION OUTPUT MAY NOT DISPLAY WHEN BEING ANALYZED FROM THE INTEL ADVISOR GUI

When the Intel Advisor GUI is launched from a Linux OS desktop GUI (such as the KDE* or GNOME* GUI) the output of the non-GUI target application being analyzed will not be displayed. To avoid this, launch the Intel Advisor GUI from a command shell with the `advisor-gui` command instead of the desktop GUI, or redirect the non-GUI application output to the Intel Advisor output window by choosing **File > Options > Application output destination > Application output window**.

INTEL ADVISOR HELP MAY NOT WORK IN SOME BROWSERS

If you encounter problems searching the Intel Advisor Help using a certain web browser, consider using a newer version of that browser or use a different web browser as your default browser.

INTEL ADVISOR MAY DETECT TIMER SOURCE INCORRECTLY ON SOME PROCESSORS

Intel Advisor may detect which timer source to use incorrectly on extreme desktop SNB-E (e.g. Intel® Core i7-3960X) processors. In Survey and Suitability collections the following message may be displayed in the command line or Debug Window:

Warning: Cannot load data file `/tmp/linux/openmp_triad/r007hs/data.0/25448-25455.0.trace' (Data file is corrupted).

To work around the problem, collect results from the command line using "`-run-pass-thru`" option:

```
advisor -collect survey -run-pass-thru=-timestamp=sys -- <application>
```

RUNNING INTEL ADVISOR GUI IN BACKGROUND MODE AND ANALYZING AN APPLICATION THAT TAKES CONSOLE INPUT MAY APPEAR TO CAUSE A HANG

Running Intel Advisor in the Linux console in background mode (`./advisor-gui &`) and profiling an application that uses input from the console (stdin) cause a product stop by the operating system that looks like a product hang. To avoid this problem, either run your application with input from a file specified in application arguments or do not use background mode.

HELP OR RECOMMENDATIONS HTML FILES MAY NOT WORK WITHOUT XDM-UTILS PACKAGE

Opening links to HTML files, such as help or recommendations, is not guaranteed if the `xdm-utils` package is not installed. Possible solutions are:

- Install the `xdm-utils` package.
- Use the default terminal program for KDE or Gnome desktop environments.

Intel Advisor may not work under Security-enhanced Linux* (SELinux) settings

Security-enhanced Linux* settings (SELinux) are currently not supported by the Intel Advisor and need to be either disabled or set to permissive for a successful tool suite installation. This product supports only "Permissive" mode. To run the product successfully you may either disable SELinux by

- setting the line "`SELINUX=disabled`" in your `/etc/sysconfig/selinux` file
- adding "`selinux=0`" kernel argument in `lilo.conf` or `grub.conf` files

or make a SELinux mode adjustment by

- setting the line "`SELINUX=permissive`" in your `/etc/sysconfig/selinux` file
- ask your system administrator to make a SELinux mode adjustment

You may need to reboot your system after changing the system parameters. More information about SELinux can be found at https://www.nsa.gov/what-we-do/research/selinux/documentation/

Contents tab of Intel Advisor documentation may appear truncated in certain browsers

When using certain HTML browsers to view the Intel Advisor documentation, if the Contents tab width appears truncated and hides some of the topic titles:

- Click the **Index** (or **Search**) button
- Click the **Contents** button

7 Attributions

The license information for third party software that was used to develop Intel® Advisor can be found in: `<install_dir>/licensing/third-party-programs.txt`

For the avoidance of doubt, the Intel Advisor is solely governed by the terms and conditions of the End User License Agreement for Intel® Software Development Product that accompanies the Intel Advisor.

8 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 which may cause deviations from published specifications. Current characterized errata are available on request. No product or component can be absolutely secure.

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.html.

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

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

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

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