Intel® System Debugger 2020

Release Notes for Linux host*

20 March 2020

Contents

Version History

Date	Version	Major Change Summary
March, 2020	2020 Update 1	Added support for instruction tracing based on the Last Branch Record (LBR) hardware feature, documented a Python* script to dump the Interrupt descriptor table (IDT) from the System Debugger console.
December, 2020	2020 Initial Release	Initial release of the Intel® System Debugger 2020.

Intended Audience

Original Equipment Manufacturers (OEM), Original Design Manufacturers (ODM), IBVs, OS vendors, device manufacturers, System Integrators.

Customer Support

To submit an issue, ask a question, or participate in discussions, visit the technical support forum. For additional support resources, go to the Intel® System Studio Support or the main Support page.

Introduction

This document provides system requirements, issues, limitations, and legal information for all components of the Intel® System Debugger 2020 for Linux* OS host, which includes following tools:

- Intel® System Debugger
  - System Debug (new fully Eclipse* integrated debugger)
  - System Debug Legacy (legacy debugger that is provided in previous releases)
- Intel® System Debugger - System Trace

To learn more about this product, see the Intel® System Debugger product page.

Additional Tools Version – 2020 Update 1

Tool	**Version**
OpenIPC	OpenIPC 1.2008.4464.100

New in This Release – 2020 Update 1

Intel® System Debugger – System Debug

- Added a new breakpoint dialog, which enables setting execution breakpoints by address as well as platform breaks (for example, SMM Entry or Reset break).
- Provided support for instruction tracing based on the Last Branch Record (LBR) hardware feature.
- Added clipboard support for the content of the Platform Register Dictionary and Platform Register Watch views.
- Restricted the Reset button in the toolbar to "Warm Reset" and made it enabled independently of the thread selection. Now you can warm-reset the target from root CPU node or from a stack frame as well.
- Added a column for PE/ELF modules in the Instruction Trace view.
- Documented a Python* script to dump the Interrupt descriptor table (IDT) from the System Debugger console. Check the System Debug User and Reference Guide.

Unsupported or Discontinued Features

- The Android Trace feature of Intel® System Debugger - System Trace will be permanently removed from Intel® System Studio, beginning with the next release.

Supported Operating Systems

Intel® System Debugger 2020 for Linux* host supports the following operating systems:

- Ubuntu* 18.04 LTS
- Fedora* 30.

Supported Platforms

Each Intel® System Debugger tool supports its own set of platforms and probes. The following probes are commonly used for connection:

- Intel® In-Target Probe (Intel® ITP) XDP3
- Intel® Silicon View Technology (Intel® SVT) Closed Chassis Adapter (CCA)
- Intel® Direct Connect Interface (Intel® DCI) Debug Class (DbC) cable.

See the table below for the combinations of platforms and probes supported by Intel® System Debugger 2020 (Update 1) for Linux* OS host.

	System Debug			System Trace		
	XDP3	**CCA**	**DbC**	**XDP3**	**CCA**	**DbC**
6th Gen Intel® Core™ Processor (Skylake) / 6th Gen Intel® Core™ Platform I/O (SunrisePoint PCH-LP)	XDP	CCA			CCA	
7th Gen Intel® Core™ Processor (Kaby Lake) / Intel® 200 Series Chipset (Kaby Lake PCH-H)	XDP	CCA	USB		CCA	USB
8th Gen Intel® Core™ Processor (Amber Lake-Y 2+2)	XDP	CCA			CCA	
8th Gen Intel® Core™ Processors (Coffee Lake-S) / Intel® H370 Chipset, Intel® H310 Chipset, Intel® B360 Chipset for Consumer (Cannon Lake PCH)	XDP	CCA	USB		CCA	USB
8th Gen Intel® Core™ Processor (Coffee Lake-S) / Intel® Z370 Series Chipset (Kaby Lake PCH-H)	XDP	CCA	USB		CCA	USB
8th Gen Intel® Core™ Processor (Kaby Lake R) / 6th Gen Intel® Core™ Platform I/O (SunrisePoint PCH-LP)	XDP	CCA			CCA	

	System Debug			System Trace		
	XDP3	CCA	DbC	XDP3	CCA	DbC
8th Gen Intel® Core™ (Whiskey Lake U)	XDP	CCA	USB		CCA	USB
9th Gen Intel® Core™ Processor (Coffee Lake H) / Cannon Lake PCH-H	XDP	CCA	USB		CCA	USB
9th Gen Intel® Core™ Processor (Coffee Lake S Refresh) / Cannon Lake PCH-H	XDP	CCA	USB		CCA	USB
10th Gen Intel® Core™ Processor (Amber Lake Y 4+2) / Sunrise Point PCH-LP	XDP	CCA	USB		CCA	USB
10th Gen Intel® Core™ Processor (Comet Lake) / Comet Lake PCH-LP	XDP	CCA	USB		CCA	USB
10th Gen Intel® Core™ Processor (Ice Lake) / Ice Lake PCH-LP	XDP	CCA, CCA_2wire	USB		CCA, CCA_2wire	USB
Intel Atom® Processor (Apollo Lake)			USB			USB
Intel Atom® Processor (Bay Trail / MinnowBoard MAX)	XDP					
Intel Atom® Processor (Denverton)			USB			USB
Intel Atom® Processor (Tunnel Creek)	XDP					
Intel® Celeron® Processor (Whiskey Lake U)	XDP	CCA	USB		CCA	USB
Intel® Core™ X-series Processor (Basin Falls Refresh)	XDP	CCA	USB		CCA	USB
Intel® Pentium® and Intel® Celeron® Processor (Coffee Lake S) / Cannon Lake PCH-H	XDP	CCA	USB		CCA	USB
Intel® Pentium® and Intel® Celeron® Processor (KBL-R platform based)	XDP	CCA			CCA	

	System Debug			System Trace		
	XDP3	CCA	DbC	XDP3	CCA	DbC
Intel® Pentium® Processor (Whiskey Lake U)	XDP	CCA	USB		CCA	USB
Intel® Pentium® Silver Processor or Intel® Celeron® Processor (Gemini Lake)			USB			USB
Intel® Xeon® Processor (Cascade Lake) / Lewisburg PCH	XDP	CCA	USB		CCA	USB
Intel® Xeon® Processor (Coffee Lake-S) / Cannon Lake PCH-H	XDP	CCA	USB		CCA	USB
Intel® Xeon® Scalable Processor (Skylake-SP) / Intel® C620 Series Chipset (Lewisburg)	XDP	CCA, CCA_USB, USB			CCA, CCA_USB, USB	

Known Issues – 2020 Update 1

- **Warm reset does not work properly on Apollo Lake and Denverton platforms with OpenRC configuration.**
    - **Issue:** Warm reset on Apollo Lake and Denverton with OpenRC run control puts the target in an undefined state (cores cannot be released).
    - **Workaround:** Manually reset the target to regain control.

- **Target Indicator is not visible when being started on Linux* hosts with GNOME 3.26 and above.**
    - **Issue:** On Linux* hosts with GNOME 3.26 and above, the Target Indicator is not visible when being started because GNOME 3.26 removes the system tray that the Target Indicator heavily relies upon.
    - **Workaround:** You can start the Target Indicator executable with the `--window-only` (or short `-w`) flag, which makes the Target Indicator avoid using the system tray.

- **Cannot start the Target Indicator from a file explorer with older package versions.**
    - **Issue:** Starting the Target Indicator executable from a file explorer (e.g. Nautilus) fails, since it is not recognized as an executable file.
    - **Workaround:** Upgrade the *"file"* package to a version higher than 5.36. Start the Target Indicator from the command line or via the provided application link (the package upgrade does not affect it). See the original bug report at https://bugs.launchpad.net/ubuntu/+source/file/+bug/1747711.

- **Platform security policy might inhibit debugger operation.**
    - **Issue:** In some platforms, the security policy might disable JTAG access to the CPU. This is intended to prevent reverse-engineering. In this case, the Intel® System Debugger will successfully connect to the target but will not be able to discover any CPUs on the JTAG bus and will report that no CPU is available.
    - **Workaround:** Ensure that that platform firmware has enabled access to the CPUs via JTAG. You can do it by flashing a special "debug" firmware into the target. In some cases, the CPU or the CPU module might physically disable JTAG access, especially in production or near-production versions. In this case, please work with the platform business unit to obtain a JTAG-enabled hardware.

Intel® System Debugger – System Debug

- **Breakpoint hits are not displayed in the System Debug Legacy console.**
    - **Issue:** Console messages in the System Debug Legacy do not show notifications after hitting software and hardware breakpoints, although you can see the breakpoint hits in the source code window.
    - **Workaround:** N/A

- **Reset break is not hit in System Debug Legacy with Comet Lake H (CML-H) and Coffee Lake S (CFL-S) platforms.**
    - **Issue:** After setting a reset break and resetting a target, reset break is not hit and the target and cores keep running.
    - **Workaround:** Use Intel® System Debugger – System Debug.

- **Cannot add map files compiled in a Linux* host.**
    - **Issue:** When adding map files that have been compiled in a Linux* host using Intel® System Debugger – System Debug, the debugger disconnects.
    - **Workaround:** Add the modules manually:
        1. Right-click the thread instance in the Debug view and select "Symbol Files". Alternatively, open the "Debug Configurations" dialog and select the required configuration or right-click the current debug configuration in the Debug view and select "Edit <debug_configuration>".
        2. In the "Symbol Files" tab, click Add and provide paths to the required modules. You can find paths to the modules and their offsets in the map file, which is a plain text file.

- **Target becomes unresponsive after the reset breakpoint is hit in System Debug Legacy on Apollo Lake and Gemini Lake platforms.**
    - **Issue:** On Apollo Lake and Gemini Lake platforms, a target becomes unresponsive in Intel® System Debugger - System Debug Legacy after reset breakpoint is hit.
    - **Workaround:** Use Intel® System Debugger – System Debug (new System Debug).

Change History

2020 Initial Release

- Started transition phase from Python* 2.7 to Python* 3. The transition to Python* 3 will be finalized by end of year 2019. Intel® System Debugger ships both Python* 2.7 and Python* 3.6 versions during the transition phase until 2020.
- Added support for the following platforms:
  - 10th Gen Intel® Core™ Processor (Ice Lake) / Ice Lake PCH-LP
  - 10th Gen Intel® Core™ Processor (Comet Lake) / Comet Lake PCH-LP
  - 10th Gen Intel® Core™ Processor (Amber Lake Y 4+2) / Sunrise Point PCH-LP
  - Intel® Xeon® Processor (Cascade Lake) / Lewisburg PCH.
- Introduced Target Indicator - a cross-platform tool that indicates the status of an Intel® DCI debug connection to a target platform.

Intel® System Debugger – System Debug

- Added a new Eclipse*-integrated source-level debugger, providing, among others, reworked support for platform registers, improved support for the PCI configuration space, better debug information support, and a Python*-based scripting console.

Related Documentation

Offline Documentation

You can find offline documentation for all Intel® System Debugger components at `<install_dir>/system_studio_2020/documentation_2020/en/debugger/system_studio_2020`.

For additional tools, refer to the following locations:

- Target Indicator: `<install_dir>/system_studio_2020/documentation_2020/en/debugger/system_studio_2020/system_debugger/target_indicator`
- OpenIPC: `<install_dir>/system_studio_2020/tools/openipc_<version>`.

Online Documentation

You can find documentation for all Intel® System Debugger components online at:

- System Debug
  - Get Started: https://software.intel.com/en-us/get-started-with-system-debug
  - User Guide: https://software.intel.com/en-us/system-debug-user-guide
- System Trace
  - Get Started: https://software.intel.com/en-us/get-started-with-system-trace-linux
  - System Trace User Guide: https://software.intel.com/en-us/system-trace-user-guide
  - TraceCLI User Guide: https://software.intel.com/en-us/tracecli-user-guide
- System Debug (Legacy)
  - Get Started: https://software.intel.com/en-us/get-started-with-system-debug-legacy-linux
  - User Guide: https://software.intel.com/en-us/system-debug-legacy-user-guide
  - Basic Debugging Tutorial: https://software.intel.com/en-us/debugging-with-intel-system-debugger-legacy

Notices and Disclaimers

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No product or component can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com].

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.

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 nonIntel 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