



Intel® System Debugger NDA Release Notes

Rev. 2418

Intel Confidential

Contents

- 1. Intel(R) System Debugger NDA Release Notes**
- 2. Get Help 4**
- 3. Version History 4**
- 4. Supported Systems and Tools 14**
 - 4.1 Host Operating Systems 14
 - 4.2 Supported Platforms and Probes 15
 - 4.3 Additional Tools Versions 24
- 5. Installation 24**
- 6. New Features 26**
- 7. Known Issues 27**
- 8. Change History 34**
 - 8.1 Intel® System Debugger 2024 NDA Update 2414 34
 - 8.2 Intel® System Debugger 2024 NDA Update 2410 36
 - 8.3 Intel® System Debugger 2024 NDA Update 2406 37
- 9. Related Documentation 38**
- 10. Notices and Disclaimers 39**

Intel® System Debugger NDA Release Notes



Intel® System Debugger is a modern, GUI-based architecture system software debugger. It provides a toolset to explore the state of the system, capture and analyze traces from hardware, firmware, and software sources, analyze system crash logs, and much more.

This document provides information on new features, system requirements, limitations, and legal information for all components of the Intel® System Debugger 2024 NDA for Windows* and Linux* OS hosts, which includes following tools:

- Intel® System Debugger - System Debug
- Intel® System Debugger - System Trace
- Intel® System Debugger - Crash Log
- Intel® Debug Extensions for WinDbg (Windows OS host only)
- Intel® Intelligent Debug & Validation Tool (IDV Tool) (Windows OS host only)
- Additional tools for target connection.

To obtain and install Lauterbach Trace32* software, refer to [additional instructions](#) in the User Guide.

- [Get Help](#)
- [Version History](#)
- [Supported Systems and Tools](#)
 - [Host Operating Systems](#)
 - [Supported Platforms and Probes](#)
 - [Debug and Trace](#)
 - [Crash Log](#)
 - [Additional Tools Versions](#)
- [Installation](#)
 - [Installing on Windows* OS Host](#)
 - [Selecting UI Theme for Intel® Debug Extensions for WinDbg](#)
 - [Installing on Windows* OS Host for On-target usage](#)

- Installing on Linux* OS Host
- New Features
- Fixed Issues
- Known Issues
 - Target Connection Assistant
 - Intel® System Debugger - System Debug
 - Intel® System Debugger - System Trace
 - Intel® Debug Extensions for WinDbg
 - Intelligent Debug & Validation Tool (IDV Tool)
- Change History
 - Intel® System Debugger 2024 NDA Update 2414
 - New Features
 - Fixed Issues
 - Intel® System Debugger 2024 NDA Update 2410
 - New Features
 - Fixed Issues
 - Intel® System Debugger 2024 NDA Update 2406
 - New Features
 - Fixed Issues
- Related Documentation
- Notices and Disclaimers

Get Help

To submit an issue, go to the [Online Service Center](#) and select Intel® System Bring-up Toolkit NDA.

Alternatively, you can report issues at [Intel® Premier Support](#). For additional support resources, go to <https://software.intel.com/content/www/us/en/develop/tools/support.html>.

Version History

Note:

Intel® System Debugger U2418 does not include the latest functional and security updates. Intel® System Debugger U2422 is targeted to be released in June 2024 and will include additional functional and security updates. Customers should update to the latest version as it becomes available.

Date	Version	Major Change Summary
May, 2024	U2418	IDV Tool CLI enhancements for report generation. Instruction Trace view and Source Lookup path improvements. Transition from sventx SDK to MIPI Sys-T protocol. Auto detection and token handling improvements in Target Connection Assistant. Simplified Virtual Console startup process.
April, 2024	U2414	Upgraded the Intel® System Debugger IDE to Eclipse 2024-03. IDV Tool added delay option for rule analysis and reporting on error conditions of buffer. System Debug implemented offline PT decoding and loading KASLR enabled Linux Kernels. Added PoC for on target debug token creation and injection. Switched to new service based Trace Message Viewer.
February, 2024	U2410	
February, 2024	U2406	
December, 2023	U2350	

Date	Version	Major Change Summary
		<p>Release of IDV Tool CLI API Python package. Enhancements in remote hardware-based debug scenario and Modules view. New options for BSP thread filtering and EFI firmware volume module loading. Initialization of Intel® System Debugger perspective at IDE startup.</p>
		<p>Improved FTDI driver delivery in IDV Tool. Target state UI improvements. Fetch PMC firmware information. Support for checking Virtualization status and creating kernel memory dump in WinDbg. Improvements to TraceCLI and improved warnings about unreliable captures.</p>
		<p>Platform Register context for Simics® simulator connections. Multiple improvements to the MSR and PCI views. New On-Target connection support in TCA and Trace.</p>
November, 2023	U2346	<p>Various improvements to the I/O Port and MSR views. Support for decomposing debug tokens. Reworked the command line interface for the OnTarget Agent.</p>

Date	Version	Major Change Summary
October, 2023	U2342	Support reading of multiple part IDs. Support NPU driver in windbg. Custom number of columns in the I/O Port view.
September, 2023	U2338	Improved Interrupt Descriptor Table view. DnX support in the UI. New command to dump information about pci devices like MSFT.
August, 2023	U2334	Full compatibility with RISC-V for Interrupt Descriptor Table. Re-enable telemetry. BMC support.
August, 2023	U2330	Added intrusiveness control to new ITH Windows* Driver. Intrusive flag allows to prevent power managing of Intel® Trace Hub device by OS Windows. When intrusive flag is set, trace captures will be in lossy mode, but preventing low power states. Default mode is to allow low power states, but loss messages when memory is not available for use by Intel® Trace Hub. Improved overall tool and platform stability, see release notes section for further information.
July, 2023	U2326	

Date	Version	Major Change Summary
		<p>IDV Tool added Drag & Drop functionality for Signal Order in Setup Tabs and in the Waveform viewer. Preview of Debug Token injection and Firmware update using Intel® Download and Execute (Intel® DnX). Improved Trace Target Agent Eclipse integration, support usage of local SSH config files and auto re-connect. Added System Debug I/O Port, Raw MSR and RISC-V PageTables Views. Intel® Debug Extensions for WinDbg added support for remote hardware debugging. Updated IDE to Eclipse 2023-03 using JRE 17.</p>
June, 2023	U2322	<p>Implemented customization of LoadThis command by using the extension "Load Module (Custom Search)". Added raw PCI config space tab to the PCI devices view. Added Simics® simulation support for Panther Lake.</p>
May, 2023	U2318	
April, 2023	U2314	
March, 2023	U2310	



Date	Version	Major Change Summary
		<p>Improved Intel® System Debugger User Guide to include IPCCLI documentation and Intel® System Debugger Python Modules API documentation. Added new debug token configurations to support all platforms previously supported with Platform Flash Tool (PFT), added auto selection of decode catalog for Trace CLI. Implemented auto complete of remote host addresses to new Connection dialog.</p>
		<p>Implemented auto detection for Closed Chassis Adapter + 2-Wire Adapter / I3C Adapter. Enabled Trusted Domain Execution (TDX) debugging. Add processor trace events decoding. Added Simics® simulator guest physical memory access and support for operating modes. Implemented Register Viewer based on RegDB for WinDbg*. IDV Tool: Added Platform Connected support, Rule Copy across projects and Rule Creation Date column under Setup Rules. Finally removed Intel® Dfx Abstraction Layer support.</p>

Date	Version	Major Change Summary
March, 2023	U2310	Improved Support for MTL-P 682 B1 Stepping. System Debugger include TDX registers on MSRs list. Added offline debug token creation flow using Flash Programming Tool (FPT).
February, 2023	U2306	System Debug support automatic hardware topology updates. Show Simics® simulator specific connection indicator in IDE. Added support for periodic Intel® Trace Hub flush to TraceCLI EmergencyBoot. Extend debug token support for: DG1, Tiger Lake platforms and Elkhart Lake-based platforms. Implemented support for offline debug token generation.
December, 2022	U2250	Added support for Hypervisor-Managed Linear Address Translation (HLAT), added Simics® simulator reverse debugging support for Intel® Debug Extensions for WinDbg, improved source level debugging support for Intel® Debug Extensions for WinDbg, implemented Eclipse* Help / Browser navigation actions (added home, backwards and forward)
November, 2022	U2246	

Date	Version	Major Change Summary
		<p>Introduced System Trace On-target trace capability, add support to subset of Structured Binary Data specification from MIPI SyS-T Specification v1.1, Intel® Debug Extensions for WinDbg: Add debug support for Simics® simulator instances, Implemented generate status report functionality to store all available information.</p>
September, 2022	U2238	<p>Platform Provisioning to manage debug tokens, preview of new Trace Message Viewer, new live target information view, support for Architectural LBRs in WinDbg over Intel® DCI Extensions; Support for Flexible Return and Event Delivery (FRED).</p>
August, 2022	U2234	<p>Introduced new Desktop Shortcuts for Target Indicator and Intel® Debug Extensions for WinDbg, Update Eclipse base to 2022-06, Added Simics disassembly and source-level debugging support for VPU firmware (including SHAVE cores)</p>
July, 2022	U2230	
June, 2022	U2226	

Date	Version	Major Change Summary
		<p>Support for breaking on BIOS POST codes (conditional IO breakpoints), support for LBR-based instruction trace on Lakemont / MinuteIA cores on Simics® simulation, enabled Simics® simulation support for Granite Rapids-D / Kaseyville target.</p>
		<p>Added early Meteor Lake Support, new more intuitive Connection and Reset Toolbar controls, System Trace integrated a memory management library that allows users to limit the amount of memory decoding process uses, Intel® Debug Extensions for WinDbg: Added debug support for pre-NT environment, IDV Tool: Enhancements and fixes to rule editing: undo and redo, copy and paste, drag and drop.</p>
<p>May, 2022</p>	<p>U2222</p>	<p>Updated the debugger to use the latest Eclipse* 2022-03 base, improved Eclipse* context help (use HTML user guide), added support for Zephyr 2.7 based Intel® Internet of Things Group (IOTG) Intel® Programmable Services Engine (Intel® PSE) SDK, implemented new Intel® System Debugger - System Debug Symbol Browser View.</p>

Date	Version	Major Change Summary
April, 2022	U2218	Skipped.
March, 2022	U2214	Added remote debug support for hardware platforms, introduced advanced rule creation functionality for Intelligent Debug & Validation Tool (IDV Tool), fixed issues.
March, 2022	U2210	Enabled tracing of <code>ptwrite</code> instruction and power events by System Debug, integrated Target Connection Assistant into <code>ExdiServer</code> console of WinDbg, fixed issues. For details, see Change History .
January, 2022	U2206	Added new environment and launcher scripts, updated the debugger to use the latest Eclipse* 2021-12 base, upgraded the BSOD bugcheck database, enabled Intelligent Debug & Validation Tool (IDV Tool) to toggle rule-checking for historical captures, extended target support, fixed issues. For details, see Change History .
December, 2021	U2152	Introduced Log4j version update.

Date	Version	Major Change Summary
December, 2021	U2150	Created a single Eclipse* perspective for all Intel® System Debugger components, enabled support for Python* 3.8 by TraceCLI and Target Agent, enabled TraceCLI to install and remove trace extensions, added support for hardware breakpoints by WinDbg over Intel® Direct Connect Interface (Intel® DCI), added support for Windows* 11 and Rocky Linux* host OS, fixed issues. For details, see Change History

To check Release Notes for previous product versions:

- Online: open the [online book](#) and expand the Version drop-down menu on the top-left.
- Offline: go to the [Intel® System Debugger download page](#) at Intel® Products and select a required version from the drop-down menu.

Supported Systems and Tools

See the child pages of this section for detailed information about host operating systems, target platform configurations, connection methods, and additional tools supported in this release.

- [Host Operating Systems](#)
- [Supported Platforms and Probes](#)
 - [Debug and Trace](#)
 - [Crash Log](#)
- [Additional Tools Versions](#)

Host Operating Systems

Intel® System Debugger 2024 NDA supports the following host operating systems:

- Microsoft Windows* 11
- Microsoft Windows* 10 21H1/21H2
- Microsoft Windows* Server 2019
- Microsoft Windows* Server 2022
- Ubuntu* 22.04 LTS
- Rocky Linux* 9
- Fedora Linux* 38, 39
- RHEL 9
- SUSE Linux Enterprise 15 SP
- Debian* 11, 12

Support for following Operating Systems is deprecated and will be removed in upcoming releases:

- Ubuntu* 20.04 LTS (Until Intel® System Debugger 2426)
- Fedora Linux* 37 (Until Intel® System Debugger 2426)
- Rocky Linux* 8 (Until Intel® System Debugger 2426)

Due to compatibility issues, Intel® System Debugger 2024 NDA does not support older Linux versions than those mentioned above.

For questions and assistance, contact [Customer Support](#).

Supported Platforms and Probes

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
ITP-XDP3 in the table.
- Intel® Direct Connect Interface (Intel® DCI) Out of Band (OOB) via Intel® Silicon View Technology (Intel® SVT) Closed Chassis Adapter (CCA).
CCA (DCI OOB) in the table
- Intel® DCI 2-wire via Intel® SVT Closed Chassis Adapter (CCA).
CCA (DCI 2-wire) in the table.
- Intel® SVT Debug Class (DbC) USB Debug Cable.
DbC in the table.
- Intel® DCI OOB via Intel® SVT Closed Chassis Adapter (CCA) + Intel® DCI USB Debug Class (DbC). This method suggests simultaneous use of a DbC cable and a CCA adapter.
CCA + DbC in the table.

- Simics® simulator – virtual target simulation.

See also: Learn More about Debug Probes and Technologies

Debug and Trace

The table below provides an overview of target/probe combinations supported by System Debug and System Trace components of the Intel® System Debugger NDA in the present release.

Note:

The core count of the target platform component does not affect connection and debugging experience. If you face any issues, refer to the [Known Issues section](#) of Release Notes or contact Customer Support.

Target Components	Connection Methods: System Debug	Connection Methods: System Trace
Ash Creek Falls CPU / Lewisburg PCH	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Alder Lake CPU / Alder Lake PCH-N	ITP-XDP3, CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI 2-wire), DbC
Alder Lake CPU / Alder Lake PCH-P	ITP-XDP3, CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI OOB), CCA (DCI 2-wire), DbC
Alder Lake CPU / Alder Lake PCH-S	ITP-XDP3, CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI 2-wire), DbC, Simics® simulator
Arctic Sound GPU	Simics® simulator	Lauterbach* MIPI60 probe

Target Components	Connection Methods: System Debug	Connection Methods: System Trace
Arrow Lake CPU / Meteor Lake PCH-S	ITP-XDP3, CCA (DCI 2-wire), DbC, Simics® simulator	ITP-XDP3, CCA (DCI 2-wire), DbC, Simics® simulator
Broxton I SOC (Apollo Lake)	DbC, Simics® simulator	DbC
Broxton P SOC (Apollo Lake)	DbC, Simics® simulator	DbC
Broxton P SOC: Gordon Peak Modular Reference Board (GP MRB)	DbC, Simics® simulator	DbC
Cannon Lake CPU / Cannon Point PCH-LP	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Cascade Lake CPU / Kaby Lake PCH-H (Glacier Falls)	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Cascade Lake CPU / Lewisburg PCH (Purley Refresh)	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Coffee Lake CPU / Cannon Point PCH-H	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Coffee Lake CPU / Kaby Lake PCH-H	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC

Target Components	Connection Methods: System Debug	Connection Methods: System Trace
Coffee Lake CPU / Cannon Point PCH-LP	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Comet Lake CPU / Comet Lake PCH-H	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Comet Lake CPU / Cannon Point PCH-H	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Comet Lake CPU / Comet Lake PCH-V	CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Comet Lake CPU / Tiger Lake PCH-H (Rocket Lake S)	ITP-XDP3, CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI OOB), CCA (DCI 2-wire), DbC
Comet Lake CPU / Comet Lake PCH-LP	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Cooper Lake CPU / Lewisburg PCH (Cedar Island)	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Denverton SOC (Car Branch 1, Car Branch 2, Denverton)	DbC, Simics® simulator	DbC
DG1 PCI Express Card GPU	Simics® simulator	Lauterbach* MIPI60 probe
DG2 PCI Express Card GPU	Simics® simulator	Lauterbach* MIPI60 probe



Target Components	Connection Methods: System Debug	Connection Methods: System Trace
Diamond Rapids CPU	Simics® simulator	Simics® simulator
Elkhart Lake CPU / Mule Creek Canyon PCH	CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI OOB), CCA (DCI 2-wire), DbC
Emerald Rapids CPU / Emmitsburg PCH (Eagle Stream Refresh)	ITP-XDP3, CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator
Gemini Lake SOC	DbC, Simics® simulator	DbC
Gemini Lake Refresh SOC (Gemini Lake +)	DbC, Simics® simulator	
Grand Ridge CPU / Emmitsburg PCH	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator
Granite Rapids CPU	ITP-XDP3, Simics® simulator	ITP-XDP3, Simics® simulator
Granite Rapids D (Kaseyville) CPU	ITP-XDP3, Simics® simulator	ITP-XDP3, Simics® simulator
Ice Lake CPU / Cedar Fork PCH	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Ice Lake CPU / Lewisburg PCH (Whitley)	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC

Target Components	Connection Methods: System Debug	Connection Methods: System Trace
Ice Lake CPU / Ice Lake PCH-LP	CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI OOB), CCA (DCI 2-wire), DbC
Ice Lake CPU / Ice Lake PCH-N	CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI OOB), CCA (DCI 2-wire), DbC
Jasper Lake CPU / Jasper Lake PCH-N (Jasper Lake +)	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Kaby Lake CPU / Kaby Lake PCH-H	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Kaby Lake CPU / Sunrise Point PCH-H	ITP-XDP3, CCA (DCI OOB), Simics® simulator	CCA (DCI OOB)
Kaby Lake CPU / Sunrise Point PCH-LP	CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Lakefield SOC	CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI OOB), CCA (DCI 2-wire), DbC
Lunar Lake CPU / Lunar Lake SoC M	ITP-XDP3, CCA (DCI 2-wire), DbC, Simics® simulator	ITP-XDP3, CCA (DCI 2-wire), DbC, Simics® simulator
Meteor Lake CPU / Meteor Lake PCH-S	ITP-XDP3, CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI 2-wire), DbC, Simics® simulator

Target Components	Connection Methods: System Debug	Connection Methods: System Trace
Raptor Lake CPU / Alder Lake PCH-S	ITP-XDP3, CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator
Rocket Lake CPU / Comet Lake PCH-H	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Rocket Lake CPU / Tiger Lake PCH-H (Tatlow)	ITP-XDP3, CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI OOB), CCA (DCI 2-wire), DbC
Sapphire Rapids CPU / Alder Lake PCH-S (Fishhawk Falls)	ITP-XDP3, CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator
Sapphire Rapids CPU / Emmitsburg PCH (Eagle Stream)	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC, Simics® simulator
Sierra Forest CPU	ITP-XDP3, Simics® simulator	ITP-XDP3, Simics® simulator
Sierra Forest CPU / Emmitsburg PCH	ITP-XDP3, Simics® simulator	ITP-XDP3, Simics® simulator
Skylake CPU / Sunrise Point PCH-H	ITP-XDP3, CCA (DCI OOB), Simics® simulator	CCA (DCI OOB)
Skylake Server CPU / Kaby Lake PCH-H (Basin Falls)	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Skylake Server CPU / Lewisburg PCH (Anderson Creek, Purley)	ITP-XDP3, CCA (DCI OOB), CCA+DbC, DbC, Simics® simulator	CCA (DCI OOB), CCA +DbC, DbC

Target Components	Connection Methods: System Debug	Connection Methods: System Trace
Skylake CPU / Comet Lake PCH-LP	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	
Skylake CPU / Sunrise Point PCH-LP	ITP-XDP3, CCA (DCI OOB), Simics® simulator	CCA (DCI OOB)
Snow Ridge CPU / Cedar Fork PCH (Jacobsville)	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Spring Hill CPU / Ice Lake PCH-LP	CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI OOB), CCA (DCI 2-wire), DbC
Tiger Lake CPU / Tiger Lake PCH-H	CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator
Tiger Lake CPU / Tiger Lake PCH-LP	CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator	CCA (DCI OOB), CCA (DCI 2-wire), DbC, Simics® simulator
Whiskey Lake CPU / Comet Lake PCH-LP	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Whiskey Lake CPU / Cannon Point PCH-LP	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC
Whiskey Lake CPU / Sunrise Point PCH-LP (Amber Lake Y 4+2)	ITP-XDP3, CCA (DCI OOB), DbC, Simics® simulator	CCA (DCI OOB), DbC



Target Components	Connection Methods: System Debug	Connection Methods: System Trace
Ponte Vecchio (PVC) PCI Express Card GPU		Lauterbach* MIPI60 probe
Battlemage		Lauterbach* MIPI60 probe, Simics® simulator
Panther Lake	Simics® simulator	Simics® simulator
Clearwater Forest	Simics® simulator	Simics® simulator
Granite Rapids Workstation (Meteor Lake PCH)	Simics® simulator	Simics® simulator

Crash Log

Intel® System Debugger - Crash Log supports the following target components in the present release:

- Alder Lake CPU (variants: H, K, L, M, N, P, R, S)
- Alder Lake PCH (variants: N, P, S)
- Arrow Lake (variants: P, S)
- Broxton I, Broxton P
- Comet Lake PCH (variants: H, LP)
- Cannon Lake (variants: H, LP)
- Emmitsburg PCH
- Emerald Rapids (variants: SP)
- Granite Rapids (variants: AP, SP)
- Granite Rapids-D
- Ice Lake CPU (variants: H, I, S3, S4, UN/YN)
- Ice Lake PCH (variants: H, LP, N)
- Jasper Lake CPU
- Jasper Lake PCH
- Lakefield CPU (base die and compute die)

- Lunar Lake (variants: M)
- Meteor Lake CPU
- Meteor Lake PCH
- Mule Creek Canyon PCH
- Raptor Lake (variants: P, S, U)
- Rocket Lake (variants: S, U)
- Sapphire Rapids SP
- Sapphire Rapids HBM
- Sierra Forest (variants: SP, AP)
- Tiger Lake CPU (variants: H, R, S, UN/YN)
- Tiger Lake PCH (variants, H, K, LP)

Additional Tools Versions

Tool	Used on OS host	Version
OpenIPC	Windows* OS and Linux* OS	24.15.13565.200
IPCCLI	Windows* OS and Linux* OS	4.24.21511.100
Lauterbach Trace32* (T32 scripts supported by Intel)	Windows* OS	24.12.34.200

Installation

Once you have downloaded an executable installation file from Intel® Products, the wizard will guide you through the installation procedure. Check the text below for additional information that may be useful during installation.

Installing on Windows* OS Host

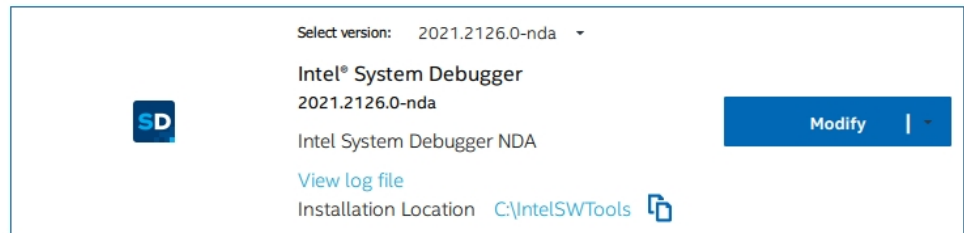
The default installation directory (referred to as <install-dir> in primary documentation) is C:\IntelSWTools\system_debugger\<>version<>-nda, where <version> is a four-digit number indicating the year and a workweek when the

release is out (for example, 2138).

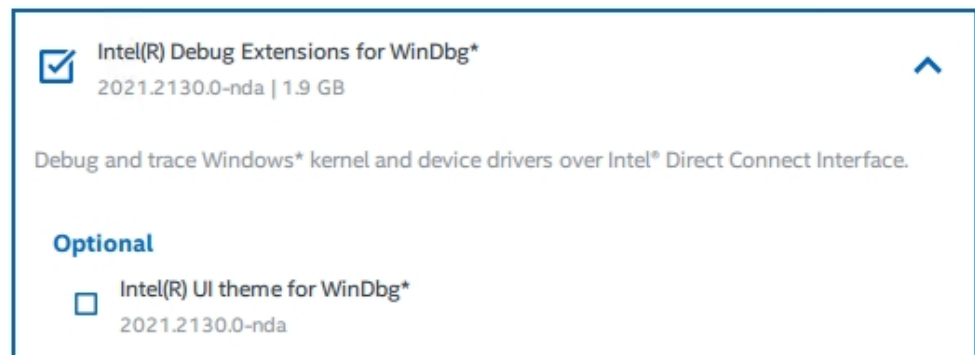
Selecting UI Theme for Intel® Debug Extensions for WinDbg

By default, a special theme designed by Intel is applied to your WinDbg installation. You can modify your theme selection (uninstall or reinstall it) after the product installation as follows:

1. Launch your installer again.
2. Select the latest version of Intel® System Debugger NDA, click **Modify**, and select **Add/Remove components**.



3. Expand the Intel® Debug Extensions for WinDbg component, (un)check the **Intel® UI theme for WinDbg*** box, and let the installer modify the product.



Without this component installed, WinDbg will use the default theme by Microsoft*, which will not affect product performance.

Installing on Windows* OS Host for On-target usage

On-target trace collection requires the installation of only a few components. It is recommended to install the System Trace package and its hard dependencies like DFLaunch and OpenIPC Debug Library.

In addition, On-target trace requires the installation of Intel® Trace Hub (Intel® TH) driver version 10.0.22000.549 or later. For information on how to get to the driver, please follow the System Trace documentation.

Installing on Linux* OS Host

Important: You need to install the product as root.

The default installation directory (referred to as <install-dir> in primary documentation) is `/opt/intel/oneapi/system_debugger/<version>-nda`, where <version> is a four-digit number indicating the year and a workweek when the release is out (for example, 2138).

New Features

Target Connection Assistant

- Various improvements to platform auto detection.
- Resolve token handling status code to a label.
- Use a single CLI command to start the Virtual Console.

Intel® System Debugger - System Debug

- Instruction Trace view will not refresh anymore when switching between the views.
- Source Lookup path will be automatically added in case of Perf-Based Processor Trace Decoder.

Intel® System Debugger - System Trace

- This will be the last release containing the sventx SDK. Please use the MIPI System Trace protocol instead, details see <https://github.com/MIPI-Alliance/public-mipi-sys-t>

Intelligent Debug & Validation Tool (IDV Tool)

- Added IDV Tool CLI support for generating analysis report in CSV format with waveform images.

Target Connection Assistant

- Fixed loading of debug & trace perspective as default IDE perspective.

Intel® System Debugger - System Debug

- Fixed Extended PCI configuration space detection. Before it has always been unavailable.
- Fixed "Browse" button opening issue for Linux.
- Fixed breakpoints added from C/C++ Event Breakpoint.

Intel® System Debugger - System Trace

- Fixed CSV export of Trace session from TraceCLI and Eclipse UI which contained redundant "Index" column.
- Fixed corrupted trace messages captured from Simics® which have been caused by unsynchronised timestamps.

- The HW tests dialog in System Trace contains unmaintained tests. The dialog already had a deprecation notice. It has now been removed.

Intel® Debug Extensions for WinDbg

- Fixed Intel Processor Trace® errata for Lunar Lake platforms.

Known Issues

Cannot launch Target Indicator, getting error about missing VCRUNTIME140_1.dll / Cannot do any tracing with "Failed to load platform" error (Windows* host only)

Issue: This should not be an issue, but in case it appears - it means Microsoft* C++ Redistributables package is broken.

Workaround:

1. Go to "Add/Remove Programs".
2. Uninstall all "Microsoft C++ Redistributables".
3. Uninstall Intel® System Debugger.
4. Install Intel® System Debugger again.

Trace and debug operations on Alder Lake S require changes in platforms OS workaround settings

Issue: Power state transition (S5, S4, WR) with probe mode detection method TapPolling causes debug tool hang. Meanwhile, TapPolling is required for run control on targets with OS running. Starting with U2122 release, Intel® System Debugger NDA enables you to manage probe mode detection logic based on use cases.

Workaround: Follow instructions at `<install_dir>/documentation/tca/usage/connecting_gui.html`. Depending on your goals, complete step 7 as follows:

- For BIOS debugging or FW/SW tracing, select "Target runs no OS" (Default). Debugging and tracing on platform that does power transitions requires this option selected.
- For working with Architectural Event Traces (AET) or WinDbg over Intel® Direct Connect Interface (Intel® DCI), select "Target runs an OS".

Disconnecting and reconnecting Intel System Debugger to Alder Lake S target with survivability flag enabled causes NorthPeak (NPK) hardware hang

Issue: During NPK tool disconnect and reconnect, NPK IP is power gated and then power un-gated and the default register value causes timeout and hang.

Workaround: Disable survivability mode if debugging across low power states is not required.

Low power debug is not supported for Alder Lake P due to missing Low Voltage Margining (LVM) overrides

Issue: Alder Lake P Best Known Configuration (BKC) does not support low power.

Workaround: N/A

Unicode error returned during target selection via console

Issue: When selecting a target with a command like `trace.interactive_setup()`, an error like *UnicodeEncodeError: 'latin-1' codec can't encode character 'u2122' in position 24: ordinal not in range(256)* is displayed.

Workaround: Before starting the console session, execute the following command:

- Windows: `set PYTHONIOENCODING=UTF8`
- Linux: `export PYTHONIOENCODING=UTF8`

Cannot connect Intel® System Debugger to Alder Lake P target with USB3 Debug Class (DbC)

Issue: Debug connection is not working with USB3 DbC connected to Alder Lake P. Reset break is also not functional.

Workaround: There are two ways to resolve connectivity issues with Intel System Debugger if USB3 DbC is detected:

- In the final step of establishing target connection ("Connection Summary" dialog box), expand the "Platform Debug Configuration" tab and uncheck the "Configure High Bandwidth Transport" box.
- Modify IFWI via Intel® Flash Image Tool (Intel® FIT) to enable USB2 port and avoid USB3 DbC detection.

Cannot establish USB DbC connection to debug the 10th generation Intel® NUC with Alder Lake M or Alder Lake P

Issue: When attempting to connect to the target using USB DbC, both Target Indicator and Windows* Device Manager report errors on the USB device (USB DbC probe).

Workaround: N/A

Target Indicator is not visible when being started on Linux* hosts with GNOME 3.26 and above (Linux* host only)

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 (Linux* host only)

Issue: Starting the Target Indicator executable from a file explorer (for example, 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](#).

Reset causes libusb error and loss of run control on Linux* host

Issue : Warm or cold reset of the target connected to Comet Lake CPU/Comet Lake PCH-H (CML-S) in user interface or command line (ipccli) causes `libusb` error and loss of run control. Cold reset is necessary to connect to the target again.

Workaround: Use the Intel® DCI USB DbC2 instead of USB DbC3 on the port used for the debug connection. You can configure this in the Intel® Flash Image Tool (Intel® FIT)

1. Launch the Intel® FIT (obtained from Intel® Validation Internet Portal) and change the following settings within your current configuration:
 - Debug > Direct Connect Interface Configuration > Direct Connect Interface (DCI) Enabled > Yes
 - Debug > Early USB DBC over Type-A Configuration > USB2 DbC port enable > USB2 Port <X>
where X identifies the post number. Depending on the target, set the following value:
 - 1 - to enable USB2 DbC on the vertical USB type-A port on CML-U DDR4 boards
 - 3 or 4 - to enable USB2 DbC on either of the two stacked USB type-A ports respectively on CML-U LP3 boards
 - 8 - to enable USB2 DbC on the vertical USB type-A port on CML-S boards with CMP-H or CNP-H chipset.
2. Build the image and flash the target with the new image.

Disconnecting a probe crashes the target (Windows* host only)

Issue: Intel® Silicon View Technology (Intel® SVT) Closed Chassis Adapter (CCA) and Intel® In-Target Probe (Intel® ITP) - XDP3 probes are supporting hot plug/unplug from a target, whereas for Intel® DCI DbC the connection is bidirectional. If a probe connection is lost, the debugger reports a Power Loss event.

Workaround: If the target was running, reconnect the probe (the disconnection has no effect on the running target). If the target was halted, reboot the target and restart the debugging session (the debug context is lost).

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.

Intel® System Debugger NDA Release Notes

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.

Issue: When you attempting to archive Intel® DAL logs using the commands `itp.daldebugloggerlevel("default")` followed by `itp.loggerarchive()`, a “Permission Denied” error is printed.

Workaround: Before every execution of the commands above, navigate to `<install-dir>\tools\DAL_<version>` and modify the `PublicLogging.log4net` file to replace “DALBase.log” with a location where you have write permissions (for example, `C:\Temp\DALBase.log`).

Cannot connect Intel® System Debugger to Raptor Lake CPU / Alder Lake PCH-S target with Intel® In-Target Probe (Intel® ITP) - XDP3

Issue: Debug connection is not working with Intel® ITP XPD3 connected to Raptor Lake CPU / Alder Lake PCH-S.

Workaround: Use a different debug probe (see Supported Platforms and Probes for available connection options).

Reset Break issues on Granite Rapids and Meteor Lake

Issue: Reset break does not work on properly Granite Rapids and Meteor Lake platforms. Planting and hitting reset break does not work reliable. Once it may work, other times it may fail sporadically to halt all cores.

Workaround:

- Meteor Lake users can try to use Intel® System Debugger 2246.
- For Granite Rapids there is currently no workaround.

Resume or step from Software Breakpoint issue on Granite Rapids D

Issue: Resume or step from Software Breakpoint issue on Granite Rapids D does not work properly. An attempt to resume execution or perform a step on a target halted in a Software Breaks does not work properly and can lead to a MCA exception.

Workaround: Use should use Hardware Breakpoints instead of Software Breakpoint.

Installation of DFLaunch fails

Issue: Installation of DFLaunch fails and the installer rolls back the installation.

Workaround: Check if there are existing virtual environments in DFLaunch with `dfmanager` and remove defect ones. Afterwards retry the installation.

Switching between Eclipse workspaces on Linux might fail

Issue: When using “File -> Switch Workspace” in the Eclipse IDE to switch to a different workspace, the IDE might not start with the new workspace and show an error instead. This happens only on Linux.

Workaround: The IDE can be started again and it will load the newly selected workspace. Another workaround is to close the IDE and start it again, then select a different workspace in the Eclipse IDE Launcher dialog. If this dialog does not appear because a default workspace is selected, then the "Window -> Preferences -> General -> Startup and Shutdown -> Workspaces -> Prompt for workspace on startup" option can be enabled to make the dialog appear again.

Target Connection Assistant

Sapphire Rapids 4-socket modular system is not detected automatically

Issue: Sapphire Rapids 4-socket system is not automatically detected by the Target Connection Assistant.

Workaround:

1. Launch Intel® System Debugger, start establishing target connection, and choose "Manually select target" (for instructions, see the Intel System Debugger documentation).
2. In the final step of the connection wizard, click the gear icon and update the new OpenIPC /Provider Configuration / Configuration Name to "SPR_EBG_XDPA_Modular_4Mhz"

Intel® System Debugger - System Debug

Intel® System Debugger: Debugging in SEAM root mode is known to currently be unstable

Issue: Debugging in SEAM root mode is known to currently be unstable

Workaround: N/A

Intel® System Debugger: TD guest debugging might require the target to be Red unlocked

Issue: TD guest debugging might require the target to be Red unlocked

Workaround: N/A

Intel® System Debugger: Instruction Trace view not updating during Step operations

Issue: When performing step operations, the Instruction Trace view may fail to update automatically, preventing users from seeing the most recent instructions executed.

Workaround: To view the instructions that were executed during the previous step operations, users can manually refresh the Instruction Trace view by clicking on the "Refresh" button provided within the view. This action will update the display with the latest instruction trace information.

Intel® System Debugger: Extended PCI configuration space is unavailable.

Issue: Extended PCI configuration space is unavailable.

Workaround: N/A

Intel® System Debugger - System Trace

Rocky Linux 8 Support

Issue: Trace does not support Rocky Linux 8 because of GLIBC version mismatch (/lib/x86_64-linux-gnu/libm.so.6: version `GLIBC_2.29' not found).

Workaround: Update to Rocky Linux 9

CentOS* 7 and OpenSuse* 15 host OS are not supported (Linux* host only)

Issue: CentOS* 7 and OpenSuse* 15 host OS are not supported for System Trace use cases.

Workaround: N/A

Various error messages referring to Interview_Decoder.dll or visa64.dll are displayed (Windows* host only)

Issue: If the host system has IVI tools installed, the tool erroneously tries to load visa64.dll from C:\Windows\System32 or Interview_Decoder.dll from <install-dir>\system_debugger_2020_ndasystem_tracebin.

Workaround: Do any of the following:

- Temporarily remove C:\Windows\System32\visa64.dll or uninstall the IVI tools.
- Check the information on the [Dynamic-Link Library Search Order](#) but be aware of the potential security implications.

On Discrete Graphic platforms (DG1/DG2), errors occur during connection attempt of trace capture start

Issue: Due to security policies in DG1 and DG2, access to North Peak Test Access Port over Lauterbach Processor Trace Interface connection is blocked, An error message "Failed Intel® Trace Hub hardware detection check!" appears.

Workaround: Currently, you can configure DG1/DG2 tracing only through firmware running on the target. To enable this tracing capability, launch Intel® Flash Image Tool (Intel® FIT), open Debug Settings, and configure the "Emergency mode". To start tracing, configure the firmware as described above, connect the debug probe, deselect "Configure target during capture start and stop", and press the "Start Capture" button in the Trace Tool. The error message "Error in connecting to the target. Target does not have power." will appear but you can safely ignore it.

Issue: Using TraceCLI to capture traces fails with "Unknown mode specified, falling back to port list" error message (<https://jira.devtools.intel.com/browse/DEBGGR-11176>).

Workaround: Use Eclipse-based System Trace instead of TraceCLI.

Issues when using CCA 2wire for trace streaming

Issue: Connection get lost or invalid/no trace packets are received when using CCA 2wire while trace streaming is active.

Workaround: Use a different probe for trace streaming or use MTB as trace destination

Issue when extracting Trace data from memory on Linux hosts using Eclipse GUI

Issue: Extraction Service fails to read data, example error message: Exception encountered while extracting: Failed to read 32 bytes @ 0xfe180000

Workaround: Set `export LD_LIBRARY_PATH=~/.inteloneapi/system_debugger/2410-nda/tools/OpenIPC/Bin/` before launching Eclipse

Message Distribution view does not work

Issue: Message distribution view does not show any data after decoding a capture. This view is no longer compatible with the new service-based Message viewer. The view will still work with legacy viewer if the capture is decoded with legacy usecase. However, the Message Distribution view will be deprecated in the future.

Intel® Debug Extensions for WinDbg

Compatibility issues of Intel® Debug Extensions for WinDbg for Windows* 10 SDK

Issue: WinDbg over Intel® DCI plugin is not compatible for Windows* SDK versions 1809 (10.0.17763) and 1903 (10.0.18362.1).

Workaround: Install the latest Windows* SDK version 22000.

Latest Windows* 11 SDK is not able to download symbols.

Issue: Windbg over Intel® DCI plugin is not able to download symbols with Windows* 11 SDK versions 22H2 (10.0.22621.0).

Workaround: Install previous Windows* SDK version 22000 (10.0.22000.194), or latest version (build 25197 or newer) from Windows Insider Preview Builds.

Intelligent Debug & Validation Tool (IDV Tool)

Captures are reported as starting later than the actual trigger

Issue: Sporadically captures are reported as starting later than the actual trigger.

Workaround: Trigger the same capture again.

Time difference between Analog and Digital samples

Issue: When capturing any channel as digital and analog a time difference between the two can be observed.

Workaround: None

Explanation: The analog values get sampled at a much lower speed and have to be transferred from the ADC to the IDV tool serially. Therefore an analog sample may

Intel® System Debugger NDA Release Notes

have to wait n cycles, with n being the number of active analog channels up to a maximum of 16.

Stop trigger does not work when manual start trigger is used.

Issue: When capturing signals the stop trigger condition will not work if a manual start trigger was used.

Workaround: Configure a start trigger condition.

IDV Python CLI samples are not installed

Issue: IDV Python CLI samples are not installed if customized IDV Tool only installation is selected

Workaround: Install System Debugger together with IDV Tool

Change History

This page contains update summary of three previous releases. To check full Release Notes for previous product versions:

- Online: open the [online book](#) and expand the Version drop-down menu on the top-left.
- Offline: go to the [Intel® System Debugger download page](#) at Intel® Products and select a required version from the drop-down menu.
- [Intel® System Debugger 2024 NDA Update 2414](#)
 - [New Features](#)
 - [Fixed Issues](#)
- [Intel® System Debugger 2024 NDA Update 2410](#)
 - [New Features](#)
 - [Fixed Issues](#)
- [Intel® System Debugger 2024 NDA Update 2406](#)
 - [New Features](#)
 - [Fixed Issues](#)

Intel® System Debugger 2024 NDA Update 2414

Check the summary below or see the [online version](#).

New Features

Target Connection Assistant

- Implemented PoC for on Target token creation and injection using Host Embedded Controller Interface(HECI). See related Platform Enablement section in user guide.
- Integrated debug info in device_info.json. This exposes properties like cse_trace_enabled for scripting purposes.

Intel® System Debugger - System Debug

- Implemented support for loading KASLR enabled Linux Kernels.
- Implemented support for Perf-Based Processor Trace Decoder. Follow user guide to create a Perf capture and decode it ISD source level debugger.

Intel® System Debugger - System Trace

- Deprecation notice added to Trace Hardware Tests dialog. The tests in this dialog are not maintained. This dialog will be removed in future releases.
- The new service based Message Viewer opens by default to show the trace messages in a decoded session. The legacy message viewer will still be available by selecting the "legacy" tagged usecases from the "Decode As" dialog or in the streaming preferences. Note that the previous sessions cannot be opened by new viewer since they are not compatible. Use legacy Message Viewer if your work requires to use message distribution of timeline viewer phases support.
- The legacy Message Viewer will show a deprecation notice that a new service based Message Viewer is available.
- Updated ITH on target driver to latest 10.0.22000.758.
- Improved Catalog Service part of Trace CLI, added "preview" label to catalog functions.

Intel® Debug Extensions for WinDbg

- Support to generate raw memory dump.
- Support to force type of memory read in full memory dump.

Intelligent Debug & Validation Tool (IDV Tool)

- Added delay option for rule analysis.
- Added reporting on error conditions of buffer.

Fixed Issues

IDE

- Fixed Welcome page on Windows Server* installations.
- Updated to Eclipse 2024-03 and other third party dependencies to mitigate CVEs.

Debug Token

- Fixed debug token creation for APL. Beginning with ISD 2406 ISD has been creating invalid debug tokens for APL. This has been fixed with ISD 2414.

Target Connection Assistant

- Fixed detection of platforms with locked Compute DIE. Platforms beginning with MTL have separate compute dies which report different detection information if they are not unlocked.

Intel® System Debugger - System Debug

- Fixed issue where stepping unstable for some architectures.

Intel® System Debugger - System Trace

- Fixed decoding of PMC traces on Alder Lake based platforms. Before the generated CSV file contained invalid new line characters.

Intel® Debug Extensions for WinDbg

- Fix disconnection from Simics® simulator instance.
- Fix extraction of crashlog data from memory dump.

Intelligent Debug & Validation Tool (IDV Tool)

- Fixed signal table column sort indicator.
- Improved installation of FTDI driver, activate close button after finish of installation.

Intel® System Debugger 2024 NDA Update 2410

Check the summary below or see the [online version](#).

New Features

Target Connection Assistant

- Initialize the Intel® System Debugger perspective when the IDE is started.
- Improvements to the Intel® System Debugger update notification dialog.

Intel® System Debugger - System Debug

- Enabled the platform registers in the remote hardware-based debug scenario. Platform registers (thread MSRs and device registers) are now usable in the same manner as in the local hardware-based debugging scenario.
- The optimization status of the modules and their sources (i.e. compiler flags for optimization) is now displayed in the Modules view.

- Added an option for filtering the Bootstrap Processor (BSP) thread within the Debug view.
- Added an option to load the modules of an EFI firmware volume (FV).

Intelligent Debug & Validation Tool (IDV Tool)

- Release IDV Tool CLI API Python package with IDV Tool Hardware capture and signal generator capabilities

Fixed Issues

Intel® Debug Extensions for WinDbg

- Fixed collection of full memory dump on the Sapphire Rapids platform.
- Fixed collection of full memory dump using XDP probe.
- Fixed print of registers value in regdb64 extension.

Intel® System Debugger 2024 NDA Update 2406

Check the summary below or see the [online version](#).

New Features

Target Connection Assistant

- Show green target state in connection view while being connected and the target available.
- Allow configuration of OpenIPC devices via IPC connection configurations.
- Fetch PMC firmware information on connect and after IPC reconfig event.

Intel® System Debugger - System Trace

- Added feature preview to TraceCLI command line interface to show live traces while the capture is running, see `'intel_systrace decode-live-preview -help'` and `'intel_systrace earlyboot-preview -help'` for more details.
- Implemented MIPI STP MasterID and ChannelID filtering to skip excessively high IDs (beyond hardware limits). In such cases, a warning about unreliable captures will be provided to the user.

Intel® Debug Extensions for WinDbg

- Support for checking Virtualization status
- Support for creating kernel memory dump

Intelligent Debug & Validation Tool (IDV Tool)

Intel® System Debugger NDA Release Notes

- Deliver FTDI driver with package and ask for installation if driver is missing

Fixed Issues

Target Connection Assistant

- Added decompose support for tokens which are not created by DTL (no DTL project ID).
- Hide warning on matching config name. OpenIPC supports config names ignoring case.

Intel® System Debugger - System Debug

- Fixed Linux kernel module loading from modules.order files.
- Fixed stack frame selection in the Debug view.
- Fixed endianness for Text column in I/O Port and MSR views.
- Fixed dropdown background for PCI Device selection in Linux OS.
- Fixed expanding feature for modules after re-launching the Debug session.

Intel® System Debugger - System Trace

- Improved decode performance and tool startup.
- Enabled directIO read operation for CCA probe using WinUSB driver, which reduce latency between sequential reads.
- Fixed missing BMG / ELG platform support from ISD U2350.

Intel® Debug Extensions for WinDbg

- Fixed !fullmemorydump command to use DMA.
- Updated installation message to point to latest compatible Microsoft SDK.

Intelligent Debug & Validation Tool (IDV Tool)

- Fixed issue with saving signal order

Related Documentation

Primary documentation (Get Started and User Guide) for Intel® System Debugger is available in the following formats:

- [Online at Resource & Design Center](#) (click the link or search for number 648476)
- Offline HTML located at <install_dir>/documentation/index.html
- Embedded Eclipse* Help. Go to Help (or press F1) and select User Guide.



Documentation for OpenIPC is available at `<install_dir>/tools/python3/Lib/site-packages/ipccli/html/index.html`.

Notices and Disclaimers

Intel technologies may require enabled hardware, software or service activation.

No product or component can be absolutely secure.

Your costs and results may vary.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

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

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks.

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.