# Intel® SoC Watch User Guide for Windows* OS

**(for Internal use only)**

October 2025

Version 2025.7

Intel Corporation

www.intel.com

Legal Information

**Intel Confidential**

# *Contents*

# Legal Information

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

No product or component can be absolutely secure.

Your costs and results may vary.



Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

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.

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.

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

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.

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos.

# Version History

These are the main releases of Intel® SoC Watch:

| Date | Revision | Description |
|---|---|---|
| April, 2019 | 2.10.1 | This release corrects an error in a JSON-formatted help file. |
| June, 2019 | 2.11 | Added feature to show settings for display panel low power modes (-f dpst).<br>Updates support for Tiger Lake PCH and adds Teton Glacier Optane support. Includes some support for Cascade Lake-Xeon.<br>Improves handling of unrecognized CPUs, reporting S-state when hibernation occurs, and other bug fixes. |
| September, 2019 | 2019.12 | Added support for Intel PCH code named Comet Lake V.<br>Initial support for Jasper Lake Plus .<br>Added features vr-pwr-state and display-rr.<br>Improved sata-lpm DevSlp Capabilities reporting, cpu-pkgc-dbg, and PCH metrics for various platforms. Modified pch-ip-active percentage calculation.<br>Modified hw-cpu-pstate reporting. |
| October, 2019 | 2019.13 | Added feature s0ix-subs-status to NDA for Intel platform code named Tiger Lake.<br>Fixed issue in ddr-bw and hw-cpu-cstate for Intel platform code named Tiger Lake.<br>Fixed issue in hw-cpu-pstate for Intel platform code named Ice Lake. |
| November, 2019 | 2020.1 | Added support for collecting metrics on Intel discrete graphics cards code named DG1.<br>Added NDA support for Intel platform code named Elkhart Lake and Intel PCH code named Mule Creek Canyon.<br>Added support for Intel platform code named Spring Hills.<br>Added support for Intel platform code named Tiger Lake -H.<br>Fixed issues in pcie-lpm, pcie-ltr and display-rr among others.<br>Fixed several features for Intel platform code named Lakefield. |
| February, 2020 | 2020.2 | Added collection of tool usage analytics.<br>Added support for Rocket Lake, Alder Lake, Cooper Lake, Tiger Lake -H PCH and JSP-N PCH. Extended support of Snow Ridge and Ice Lake Xeon to NDA along with various features. Added new internal feature hgs-feedback and pch-ivr-state. Added features pch-slps0, pch-slps0-dbg to External.<br>Enhanced Continuous Monitoring API.<br>Improved error messages and help output. Enhanced driver security. Improved many metrics. |
| June, 2020 | 2020.3 | Change in pch-ip-active behavior. Added feature s0ix-subs-dbg. Added (NDA) support for Intel platforms code named Cooper Lake-Xeon, Jasper Lake Plus, and Ice Lake Xeon -D. Added VMD support to PCIe features. Many bug fixes, including corrections in tcss-state, s0ix-subs-status, dpst, display-rr. |

| Date | Revision | Description |
|---|---|---|
| | | Added support for most metrics on Alder Lake. |
| July, 2020 | 2020.3.1 | Added NDA support for Intel platforms code named Rocket Lake and Sapphire Rapids.<br>Bug fixes |
| September, 2020 | 2020.3.2 | Corrected and extended support for Alder Lake.<br>Added NDA support for Intel platform code named Alder Lake and Intel® Arc Discrete Graphics Card.<br>Added soc-temp for Intel platform coded named Tiger Lake (B0) and updated pch-platform-ltr and pch-ip-status. Added ddr-bw and soc-temp for Intel platform coded named Tiger Lake -H.<br>Bug fixes including correcting hibernation detection.<br>Added cpu-pkgc-dbg for Intel platform code named Jasper Lake Plus. |
| October, 2020 | 2020.4 | Added external support for Intel platform code named Tiger Lake.<br>Added topology label in reports for some metrics.<br>Re-named feature cpu-gpu-concurrency to cpu-igpu-concurrency.<br>Removed support for older platforms. Also removed socperf.<br>This version is for external package release only. |
| November, 2020 | 2020.5 | Includes all changes noted above for v2020.4 release.<br>Changed hw-cpu-pstate to report frequencies per thread rather than per core.<br>Added term *integrated* to hw-igfx-cstate and hw-igfx-pstate report titles.<br>Fixes for PCH metrics on Intel platforms code named Tiger Lake and Alder Lake. |
| April, 2021 | 2021.1 | Added system name, OS name, and Intel PMT GUID to output reports.<br>Included throt-rsn and other sampled count results in Automation_Summary.<br>Re-ordered feature reports in summary output.<br>Added Intel PMT metrics support for additional SKUs of Intel platform code named Alder Lake.<br>Added Intel PMT features (for pkg C debug, bandwidth, LTR, and temperatures) and non-PMT features (for power limits) for Intel platforms code named Alder Lake and Tiger Lake.<br>Added DC6V state to display-state for Alder Lake.<br>Added basic support for Meteor Lake, Alder Lake -N, and Granite Rapids.<br>Added support for Raptor Lake (-S).<br>Added support for Intel® Arc Discrete Graphics Card .<br>Added NDA support for PCH on Intel platform code named Alder Lake (-P, -M).<br>Improvements/fixes for pch-ip-active, pch-ip-status, TCCold reporting, PCIe device discovery, empty summary reports, and more.<br>Modified WakeupAnalysis report and added percentage table. |
| May, 2021 | 2021.1 .1 | Fixed tcss-state for Intel platforms code named Tiger Lake -H and Alder Lake. |

| Date | Revision | Description |
|---|---|---|
| | | Fixed cpu-pkgc-dbg for Intel platform code named Alder Lake. |
| July, 2021 | 2021.2 | Extended Raptor Lake recognition, added ADL -N PCH, and ATS -M support.<br>Fixed pcie-lpm PEG and VMD issues.<br>Added dgfx-dstate for Intel discrete graphics code named DG2.<br>Added PCx LTR threshold report to soc-ltr for Intel platforms code name Tiger Lake and Alder Lake.<br>Added Intel PMT metrics perf-limit-rsn, ipu-state, llc-state, osreq-pkg-cstate, pkg-llc-flushed (internal only) for Tiger Lake and Alder Lake.<br>Enhanced WakeupAnalysis report with idle time per process.<br>Improved ia-throt-rsn support when Windows* OS secure modes are enabled.<br>Fixed panel-srr recognition of when PSR2 is enabled on Intel platform code named Alder Lake. |
| September, 2021 | 2021.3 | New feature s0ix-subs-req for Intel platform code named Alder Lake (-P).<br>Added internal support for Intel PMT metrics on Raptor Lake.<br>Bug fixes including correctness issues in pcie-lpm, pcie-ltr, cpu-pkgc-dbg, perf-limit-rsn, and several dgfx features. |
| October, 2021 | 2021.3.1 | Improved support for platforms with many sockets and/or discrete graphics cards.<br>Modified option --update-usage-consent. |
| November, 2021 | 2021.4 | Added NDA support for Intel platforms code named Raptor Lake (non-PMT features) and Alder Lake -N.<br>Added external support for Intel platforms code named Rocket Lake and Tiger Lake -H.<br>Added external support for Intel platforms code named Ice Lake -X and Cooper Lake -X.<br>Added some PCH features for Meteor Lake.<br>Multiple bug fixes including correctness issues in s0ix-subs-dbg, sata-lpm, xhci-lpm, and dgfx-soc-ltr. |
| December, 2021 | 2021.4.1 | Modified cpu-pkgc-dbg to report more detailed package C-state blocking reasons on Intel platform code named Alder Lake.<br>Improved tool stability. |
| February, 2022 | 2022.1 | Added external support for Intel platform code named Alder Lake and Intel® Arc Discrete Graphics Card .<br>Added support for more Meteor Lake features (cpu-pkgc-dbg, PCH metrics) and initial support for Lunar Lake.<br>Added NDA support for display-state.<br>Bug fixes, including handling when VTIO is enabled. |
| April, 2022 | 2022.2 | Added support for additional Meteor Lake features (from Intel PMT).<br>Bug fixes including correctness issues in Intel platform code named Alder Lake feature cpu-pkgc-dbg and PCH metrics. |
| June, 2022 | 2022.3 | Added support for additional Meteor Lake features.<br>Added support for Intel® Arc Discrete Graphics Card 256 EU. |

| Date | Revision | Description |
|---|---|---|
| | | Bug fixes including fix for collecting feature panel-srr irrespective of PSR flag status at initial collection time. |
| August, 2022 | 2022.4 | Multiple bug fixes including correctness issues in Meteor Lake features. |
| September, 2022 | 2022.5 | Enabled support for hgs-feedback for Meteor Lake, Raptor Lake and Alder Lake.<br>Bug fixes including data correctness in Intel PMT based residency metrics. |
| November, 2022 | 2022.6 | Report complete topology path on hybrid systems.<br>Added support for IOE DIE features in Meteor Lake.<br>Added support for PCH features in Meteor Lake -S.<br>Bug fixes for data correctness. |
| January, 2023 | 2023.0 | Deprecation of Google Analytics support.<br>Added support for Intel server platform code named Emerald Rapids server.<br>Added support for Intel platform code named Lunar Lake and additional metrics support (new PCH and Intel PMT metrics) for Intel platform code named Meteor Lake.<br>Added (NDA) PCH and CDie (from Intel PMT) metrics support for Intel platform code named Meteor Lake.<br>Bug fixes for data correctness. |
| February, 2023 | 2023.1 | Added (NDA) PCH and Intel PMT metrics support for Intel platform code named Meteor Lake (B-step).<br>Bug fixes for data correctness. |
| April, 2023 | 2023.2 | Added external support for Intel platform code named Raptor Lake.<br>Added external support for Intel server platform code named Sapphire Rapids -Xeon.<br>Added support for Intel platform code named Arrow Lake.<br>Added support for limited Intel PMT metrics on Intel platform code named Lunar Lake.<br>Added support for Intel discrete graphics card named Alchemist Refresh.<br>Added PCIe metrics support on Intel platform code named Meteor Lake -S.<br>Added support for external PCH metrics on Intel platform code named Meteor Lake -S.<br>Fixed issues in PCH metrics on Intel platform code named Meteor Lake.<br>Bug fixes including BSOD on systems with integrated graphics disabled when running igfx features. . |
| June, 2023 | 2023.3 | Added NDA support for Intel server platforms code named Emerald Rapids -Xeon and Granite Rapids -Xeon.<br>Added additional features support for Intel platform code named Lunar Lake (from PMT and PCH).<br>Added tcss feature support for Intel platform code named Meteor Lake.<br>Bug fixes including issues in vpu-bw, vpu-dstate-res, pch-ip-active, s0ix-subs-* features, sata-lpm. |

| Date | Revision | Description |
|---|---|---|
| August, 2023 | 2023.4 | Added support for Intel Platform code named Arrow Lake H (6+8) and Intel discrete graphics card code named Battlemage.<br>Deprecated support for pch-ip-status and sa-freq from MTL onwards.<br>Additional features support for Intel platform code named Lunar Lake.<br>Bug fixes including BSOD when VTIO is enabled, issues in s0ix-subs-* features, etc.<br>Notification about limiting the scope of certain group features including sys in future release. |
| August, 2023 | 2023.4.1 | Added limited NDA support for Intel platform code named Lunar Lake.<br>Fixed Intel PMT XML configuration file issue in Intel platform code named Lunar Lake.<br>Bug fixes. |
| September, 2023 | 2023.4.2 | Fixed an issue NDA only to avoid BSOD on Intel platform code named Lunar Lake. |
| October, 2023 | 2023.5 | Change in scope for certain group features including -f sys.<br>Added support for Intel platform code named Panther Lake.<br>Updated Intel Vision Processing Unit (VPU) nomenclature to Intel Neural Processing Unit (NPU).<br>Bug fixes |
| November, 2023 | 2023.6 | Added external support for Intel server platform code named Emerald Rapids-Xeon.<br>Added NDA support for Intel platform code named Arrow Lake and Intel server platform code named Sierra Forest-Xeon.<br>Additional feature support for Intel platform code named Lunar Lake.<br>Bug Fixes |
| December, 2023 | 2023.7 | Enhanced wakeup analysis histogram report<br>Additional features support for Intel platforms code named Lunar Lake and Arrow Lake.<br>Bug fixes including summary report not generated for longer duration collections, data correctness in telemetry features for Intel platform code named Lunar Lake. |
| January, 2024 | 2024.0 | Enabled (external) additional features support on Intel platform code named Meteor Lake.<br>Enabled (NDA) additional features support on Intel platforms code named Meteor Lake -S, Arrow Lake -S and Lunar Lake.<br>Enabled limited support for Intel platform code named Wildcat Lake.<br>Enabled limited support for Intel server platform code named Diamond Rapids.<br>Bug Fixes including reporting an error message when Intel discrete graphics card is in D3. |
| February, 2024 | 2024.1 | Enabled s0ix-subs-res support on Intel platform code named Panther Lake -P/-H. |

| Date | Revision | Description |
|---|---|---|
| | | Bug fixes including data correctness issues in NPU features on Intel platform code named Lunar Lake and Intel discrete graphics card named Battlemage. |
| April, 2024 | 2024.2 | Added a text disclaimer about validation for early access platforms.<br>Enabled additional PCD features support for Intel Platform code named Panther Lake -P/-H.<br>Introduced new feature switch tcss-state-res for Intel Platform code named Meteor Lake onwards.<br>Added warning to users for using deprecated or older telemetry GUIDs.<br>Deprecation of support for older atom based features.<br>Bug Fixes |
| May, 2024 | 2024.3 | Enabled support for Intel platform code named Arrow Lake U.<br>Enabled external support for Intel discrete graphics code named Ponte Vecchio.<br>Enabled limited support for Intel server platform code named Clearwater Forest.<br>Added support for latest telemetry GUIDs on Intel platforms code named Arrow Lake and Lunar Lake.<br>Enabled additional PCD features support on Intel platform code named Panther Lake -H/-P.<br>Bug Fixes |
| July, 2024 | 2024.4 | Enabled limited support for Intel platform code named Nova Lake.<br>Enabled support for Intel platform code named Arrow Lake S B0 stepping.<br>Enabled Intel PMT based and PCD based features on Intel platform code named Panther Lake.<br>Removed support for option pch-lvl from the feature pch-ip-active.<br>Bug fixes including data correctness issue in the feature sata-lpm. |
| July, 2024 | 2024.5 | Enabled limited support for Intel discrete graphics card named Battlemage G31.<br>Removed support for Intel discrete graphics card named Battlemage G10.<br>Added NDA support for Intel ARC discrete graphics card [320EU and 288EU] ( Intel discrete graphic card code named Battlemage G21) . |
| September, 2024 | 2024.6 | Added external support to Intel platforms code named Elkhart Lake, Arrow Lake and Lunar Lake.<br>Added limited NDA support for Intel platform code named Panther Lake.<br>Added additional features to Intel platform code named Panther Lake.<br>Enhanced the feature s0ix-subs-dbg in Intel platform code named Panther Lake to support B/D/F of the reported components.<br>Bug Fixes |
| October, 2024 | 2024.7 | Enabled support for the latest OOBMSM telemetry GUID on Intel ARC discrete graphics card [320EU and 288EU] (Intel discrete graphics card code named Battlemage G21) |

| Date | Revision | Description |
|---|---|---|
| | | Additional features external support on Intel Platform code named Lunar Lake<br>Added support for option --result-slice-range.<br>Enhanced summary report to include S0ix LPM capability of the system.<br>Bug Fixes |
| November, 2024 | 2024.8 | Bug Fixes |
| December, 2024 | 2024.9 | Enabled additional features on Intel platforms code named Panther Lake -H, Wildcat Lake and Nova Lake.<br>Added NDA support for feature s0ix-subs-dbg on Intel platform code named Panther Lake -P.<br>Bug Fixes. |
| January, 2025 | 2025.0 | Enabled additional features support on Intel platform code named Wildcat Lake.<br>Added NDA support for feature cpu-pkgc-cfg on Intel platforms code named Meteor Lake onwards.<br>Added support for latest telemetry GUIDs on Intel platform code named Arrow Lake.<br>Bug fixes including an issue with the help output. |
| February, 2025 | 2025.1 | Enabled external support for Intel® Arc™ B-series graphics.<br>Enabled external support for the feature memss-pstate on Intel platform code named Lunar Lake.<br>Enabled additional features NDA support on Intel platform code named Panther Lake. The support includes Intel PMT based features.<br>Bug fixes |
| April, 2025 | 2025.2 | Added external support for Intel server platforms code named Granite Rapids and Sierra Forest.<br>Enabled additional features support on Intel platform code named Nova Lake and Intel server platform code named Diamond Rapids.<br>Enabled limited support for Intel platform code named Bartlett Lake which is a derivative of Intel platform code named Raptor Lake.<br>Bug fixes including issues in TCSS, s0ix-subs-dbg features. |
| April, 2025 | 2025.3 | Added support for latest Pcode telemetry GUID on Intel support for Intel® Arc™ B-series graphics.<br>Enabled support for Intel discrete graphics card named Battlemage G31. |
| May, 2025 | 2025.4 | Added support for latest Pcode telemetry GUID on Intel support for Intel® Arc™ B-series graphics.<br>Added support for latest Pcode telemetry GUID for Intel discrete graphics card named Battlemage G31.<br>Enabled additional features support for Intel platforms code named Wildcat Lake and Nova Lake.<br>Bug fixes including refining pcie-ltr feature. |

| Date | Revision | Description |
|---|---|---|
| July, 2025 | 2025.5 | Enabled additional features support (including Intel PMT based features) on Intel platform code named Nova Lake.<br>Enabled additional features NDA support for Intel platform code named Panther Lake PCD P.<br>Enabled NDA support for Intel platforms code named Panther Lake PCD H and Wildcat Lake.<br>Re enabled support for PCIe based features on Windows OS 26100 or higher when Enhanced security is turned on.<br>Bug fixes. |
| August, 2025 | 2025.5.1 | Removed support for PCIe features when VTIO is enabled to avoid system BSOD. |
| September, 2025 | 2025.6 | Improved the topology report for hybrid platforms including core type.<br>Enabled NDA support for Intel discrete graphics card code named Battlemage G31.<br>Enabled NDA support for Intel PMT based features on Intel platform code named Wildcat Lake.<br>Enabled PCH based features support on Intel server platform code named Granite Rapids Workstation.<br>Bug fixes. |
| October, 2025 | 2025.7 | Added support for additional PCH / PCD features on Intel platform code named Nova Lake -H/-S.<br>Enabled external support for Intel platform code named Panther Lake.<br>Bug Fixes. |

# About Intel® SoC Watch

Intel® SoC Watch is a command line tool for monitoring and debugging system behaviors related to power consumption on Intel® architecture-based platforms. It reports active and low power states for the system/CPU/GPU/devices, processor frequencies and throttling reasons, IO subsystem low power states and latency tolerance, PCH activity, memory and display refresh states, wakeups, and other metrics that provide insight into the system's energy efficiency. The tool includes utility functions that include delaying the start of collection and launching an application prior to starting collection.

Data is collected from both hardware and OS sources. When using the default mode of collection, the tool collects data at normally occurring OS context-switch points so that the tool itself is not perturbing the system sleep states. Tool overhead when collecting during idle scenarios can be < 1%, however active workloads with a high-rate of context switching will increase the overhead. A minimum collection interval is used to control the rate of collection.

Intel SoC Watch writes a summary report file (.csv) at the end of collection on the system under analysis (target system), allowing immediate access to results. Additional result files can be specified including: an import file (.pwr) for Intel® VTune™ Profiler that can be used for visualization of correlated timelines for all the collected metrics with powerful zoom and filtering functions, and a time trace file (.csv) that can be viewed as a timelines in tools like Microsoft* Excel*.

## Related Information

See the Intel® SoC Watch Release Notes for information on new features as well as known issues.

For online help, including information about importing results into Intel® VTune™ Profiler, see the Energy Analysis User Guide (https://software.intel.com/en-us/energy-analysis-user-guide).

# 2 Installation

See the *Intel® SoC Watch Release Notes* for supported platforms and installation instructions.

# Getting Started with Intel® SoC Watch

3

1. Launch a command prompt with administrative privileges. Intel SoC Watch requires administrative access to load a device driver.
2. Run the `socwatch` command with one or more required options to collect energy data.
3. Review the reports generated by Intel SoC Watch to analyze energy consumption.
4. [Optional] Import the results into Intel VTune Profiler to review a visual representation of the data collected.

Listed below are a few basic commands that will have you looking at Intel SoC Watch reports quickly. Use the Options Quick Reference section to learn about all the commands, their options, and abbreviations.

To collect and report CPU C-state and P-state residency data and system sleep state residency for the platform, replace `<duration>` with the number of seconds you want to collect data and use the command below. Intel SoC Watch will write a summary report to the file `SOCWatchOutput.csv`.

```
socwatch -t <duration> -f cpu -f sstate
```

If you want to name the output files something other than the default `SOCWatchOutput`, replace `<string>` in the command below with the name you want. After collection, look for the `<string>.csv` file to find the summary report.

> **Tip**
> Include a folder name in the string to group the files under a folder. The folder will be created if it doesn't exist.

```
socwatch -t <duration> -f cpu -f sstate -o <string>
```

To collect a broad set of system metrics for a high level look at platform power behavior, replace `<duration>` with the number of seconds you want to collect data in the following command. Intel SoC Watch will write a summary report to `SOCWatchOutput.csv`.

```
socwatch -t <duration> -f sys
```

To get detailed reports available for any metric being collected (e.g., idle and wakeup analysis and timelines), include the `--max-detail` (`-m`) option during collection and specify additional report types using the `--result` option. In the example below, all metrics in group name `sys` are collected for 60 seconds, including over time data and any detailed data for metrics that offer it. A trace file is generated (`-r int`) in addition to the summary report. All the results are in files named `run1` under the folder `sysDetail`.

```
socwatch -t 60 -f sys -m -r int -o sysDetail/run1
```

Note that trace (timeline) reports for long collections can take many minutes to generate. In this case, you may wish to generate only summary reports at the end of collection and later request the trace file be generated as shown below using the `--input` (`-i`) option. In this example, the sys group of metrics is collected for 60 minutes with maximum detail, but only summary report is generated initially. The second command causes the results to be re-processed and timeline reports generated. All of the new reports have the same base name and location unless a `-o` is included to give the re-processed results a new name.

**Tip**
Use the `-o` option in conjunction with `-i` to change the base name.

```
socwatch -t 3600 -f sys -m -o myResult
socwatch -i myResult -r int
```

To generate an export file containing all supported metrics that can be opened for viewing in Intel VTune Profiler, include the `-r vtune` option alongside any other collection parameters. The `--max-detail` option is required to generate a complete set of trace data for all metrics. The following command will collect platform data for 30 seconds and produce a file called `run1.pwr` that can be imported to Intel VTune Profiler.

```
socwatch -t 30 -f sys -r vtune -m -o run1
```

# Entering Connected Standby (Modern Standby)

If running on a system that supports Connected Standby and has WDTF installed, use the `--auto-connected-standby` (`-z`) option to automatically put the system in Connected Standby and then exit standby when the collection time completes. You can manually exit standby early and collection will stop at that time and generate the reports. You can also enter Connected Standby manually after starting a collection.

**NOTE**
In order to use the `--auto-connected-standby` (`-z`) option, the Windows* OS Driver Test Framework (WDTF) must be installed. See the *Installation Notes* section in the Release Notes for instructions.

Your platform must be enabled for *S0 Low Power Idle* state for this option to successfully put the system into Modern Standby after the WDTF is installed. The following command should list a sleep state such as *Standby (S0 Low Power Idle) Network Connected*, as available on this system if Modern Standby is possible: `powercfg /a`

Example: The following command collects CPU C-state residency for 60 seconds after putting the system into Connected Standby. It will delay collection and the transition to standby by the `--startdelay` (`-s`) time of 20 seconds. (There is no option to delay the start of collection after entering standby.)

```
socwatch -s 20 -t 60 -f cpu-cstate -z
```

# 4 Options Quick Reference

Invoke Intel SoC Watch with administrator privilege, using the following syntax:

```
socwatch <general options><post-processing options><collection options>
```

- Order of options does not matter unless specifically noted.
- Help is displayed if no option is specified.
- All features are not available on all systems, so the help text is dynamic, meaning it displays only the collection options that are supported by the system on which it is run. The metrics available differ because of changes in the system's hardware architecture support. This User's Guide contains a list of all metrics across all systems.
- You can specify feature names that are not available or not enabled on a particular system. When the tool starts, it will display console messages regarding features that cannot be collected, but collection will proceed if at least one feature is valid on that system.

Intel SoC Watch terminates data collection for one of three reasons (whichever occurs first):

1. the `--time` option was specified and the timer elapsed,
2. the `--program` option was used and the specified program exited,
3. a Ctrl-C interrupt was entered in the command window.

The location and name of the results files is displayed at the end of a collection. The summary report will be there with that name and a `.csv` extension. Raw data files and additional files based on post-processing options specified on the command line are located there as well, all with the same base name (default name is `SoCWatchOutput`).

**NOTE**
Result files are replaced if the same name is used for multiple collections.

## General Options

The following options display information about the tool or system on which it is run.

| Abbreviation | Option Name | Description |
|---|---|---|
| `-c` | `--config-file` *<config_file_name>* | Specify alternative configuration file or path to a directory containing one or more configuration files. The argument can include an absolute or relative path name. A configuration file contains architectural information on how to read metrics during collection. Incorrect specifications can result in a system fault. If this option is not specified, `SOCWatchConfig.txt` is used. |
| `-d [level]` | `--debug [level]` | Controls display of error, warning and information messages on the console. Higher levels are used to debug collection and processing issues. Supported levels:<br>0: Only display fatal error messages<br>1: Also display error messages [default]<br>2: Also display warning messages |

| Abbreviation | Option Name | Description |
|---|---|---|
| | | 3: Also display debug messages<br>4: Also display informational messages |
| | `--disable-cstrace` | Uninstall *cstracedrv.sys*. System reboot is required after executing this option to complete the uninstall. The *cstracedrv* driver is automatically loaded the first time feature -f connected-standby is executed on a platform, or by using the --enable-cstrace option. It remains installed after the collection ends (unlike the *socwatch* driver) and is automatically reloaded after a system reboot. The uninstall must be completed by rebooting, before this driver can be installed again (meaning the connected-standby feature will fail until after a reboot). |
| | `--enable-alts` | Enable alternative key for terminating collection [ALT-S]. |
| | `--enable-cstrace` | Install *cstracedrv.sys*. System reboot is required after executing this option. Use this option (and reboot) prior to use of the -f connected-standby feature in order to get complete session information for the connected standby analysis. If the first use of -f connected-standby is used to install this driver, the session information reported will be incomplete until after a reboot. Note: The *cstracedrv* must be manually uninstalled, see --disable-cstrace. |
| | `--export-help` | Write help output to JSON formatted file. |
| `-h` | `--help` | Display tool usage information and exit. The help shown is specific to the system on which it is run. Only metrics supported by the system architecture will be listed. |
| | `--info` | Display additional low-power system settings. (includes LPIT table) |
| `-l` | `--log <filename>` | Redirect all console output, including errors, to specified file. |
| | `--print-fms` | Display CPU ID as Family.Model.Stepping and exit. |
| `-v` | `--version` | Display tool version information and exit. |

# Post-processing Options

The following options affect how results are reported and where they are stored.

| Abbreviation | Option Name | Description |
|---|---|---|
| `-i` | `--input <filename>` | Specify the collection name (with full path) of an existing collection to generate reports. |

| Abbreviation | Option Name | Description |
|---|---|---|
| -o | --output *<filename>* | Specify the output collection name (default "SoCWatchOutput"). Specifying `console` as the filename will cause the summary results to also write to stdout. If the filename already exists, the previous results will be replaced. |
| -r | --result *<result_type>* | Specify the type of result to generate. This option can be repeated to get multiple types of reports. Following are the result types that can be specified:<br>• `sum` Write summary reports to .csv file. [default]<br>• `int` Write over-time data to _trace.csv file.<br>• `vtune` Generate .pwr file for import to Intel VTune Profiler.<br>• `json` Generate .swjson file.<br>• `auto` Write summary results as a single line to file Automation_Summary.csv in current directory. Appends results, does not overwrite. If column headers for the new result changed, new headers will be inserted. Use to generate sets of data in a single file for comparison. |
| | --result-slice-range *<rangeStartTime in msec><rangeEndTime in msec>* | Specify a time range in milliseconds bounding a subset of the result file to post-process. Time range specified should be relative to collection start time and only one result slice may be processed at a time. Use with the -i option to specify input file to post-process. |

## Collection Options

These options affect what is collected and how it is collected.

| Abbreviation | Option Name | Description |
|---|---|---|
| -f | --feature *<name>* | Specify which metric to collect, choose from the group names or individual names listed in the tables below. This option can be repeated to collect multiple metrics in a single run. Most features can be collected simultaneously, exceptions noted in the table of feature names. |
| -m | --max-detail | Collect all data available for each feature specified. This will cause snapshot metrics to be sampled. Use of this option can increase tool overhead, so best used only when timeline of the data is needed or when collecting across system entry to hibernation.<br>Without this option, the tool collects data at the minimum required by the data source for best accuracy.<br>Data may be traced, sampled, or snapshot. |

| Abbreviation | Option Name | Description |
|---|---|---|
| | | • Traced data is obtained at state transition points resulting in accurate summary and timeline results.<br>• Sampled data is read at OS context switch points (or at timed intervals if polling option is used). This is less accurate as changes that take place between samples will not be measured. Metrics that come from hardware status/ state data must be sampled.<br>• Snapshot data can be read at the beginning and end of the collection and the difference gives an accurate result with lowest overhead, but no timeline. Only metrics that come from hardware accumulators can be snapshot.<br>The algorithm used to determine the collection method for each data type is as follows:<br>If `-m` is specified:<br>if the data can be traced, trace it; else sample it.<br>If `-m` is not specified:<br>if the data can be snapshot, snapshot it; else if the data can be traced, trace it; else sample it. |
| `-n` | `--interval <milliseconds>` | Specify the time in milliseconds that should pass before reading next hardware data sample (default 100 ms). For default collection mode, this is the minimum time between sampling at context switch points. When `--polling` option is used, this is actual time between samples.<br>The minimum polling interval is 1ms. However, using low polling intervals will result in higher overhead and may fail to measure some metrics (e.g. bandwidths) with intervals shorter than the default. |
| | `--no-post-processing` | Do not generate the summary file or other result files at the end of collection. Use -i option to process the intermediate results files and generate summary or other result file types at a later time. |
| | `--option <opt-name=value>` | Specify additional info for a metric. See `pch-ip-active` and `pch-ip-lat-limit` for valid option names and values. |
| | `--polling` | Make data collection occur at regular intervals rather than at context switch points. Use the `--interval` option to set the interval period (default: 100ms). Use of this option significantly increases |

| Abbreviation | Option Name | Description |
|---|---|---|
| | | perturbation of sleep states because it employs a timer which will interrupt sleep states, increase wakeup counts, and change timer resolution. |
| -p | --program *<application> <parameters>* | Specify the name of an executable to be started automatically prior to collection. The name can be followed by zero or more arguments that will be passed to the program.<br>**NOTE** This option must occur at the end of the command line, everything following the executable name will be given to it as arguments.<br>**NOTE** The executable can only be launched as a desktop application (not as Windows Modern UI application). |
| | --program-delay *<seconds>* | Specify number of seconds to wait before starting the program specified by -p. Has no effect if -p not used. |
| -s | --startdelay *<seconds>* | Specify number of seconds to wait before starting collection of data.<br>If used with -p and --program-delay, this delay is applied after the program starts.<br>If used with --auto-connected-standby, this delays entry to standby. |
| -t | --time *<seconds>* | Specify collection duration in seconds. Collection will stop when this time has elapsed unless Ctrl-C is entered or an executable specified with --program option exits prior to the specified duration. |
| -z | --auto-connected-standby | Automatically enter Connected Standby for the duration of the collection. Will automatically exit Connected Standby when the -t specified time expires. If system is woken from Connected Standby prior to the end of the duration, the collection will stop as well. If --start-delay is specified, it occurs prior to entering Connect Standby.<br>**NOTE** Requires Windows* Driver Test Framework, WDTF to be installed on the system under test. |

## Feature Names (Individual)

The available feature names for the `--feature` option and their collection methods are listed below. You can specify multiple feature names individually or using group names described in the Feature Group Names section.

Note that every feature listed is not available on every platform supported by Intel SoC Watch. The `--help` option is dynamic, only showing features available for the platform on which it is run. Use it to determine which features are supported. You can specify unsupported features on the command line and the tool will simply display a message for those that cannot be collected, but continue with collection if there is at least one that is supported.

Collection methods are indicative of a metric's level of accuracy and overhead. Traced collection provides high accuracy along with precise transition points between states. Sampled collection is least accurate since transitions can occur which are never noted. Sampled data needs to be read at intervals throughout the collection period which increases tool overhead. Increasing the sampling rate (reading at closer intervals) will improve accuracy but increase overhead. Snapshot collection means the data comes from an accumulator so it can be collected only at the start and end of the collection period and give perfect accuracy. This gives accuracy and the lowest overhead. If the `--max-detail` (`-m`) option is given, the Snapshot metrics will instead be read at the same intervals as the Sampled metrics throughout the collection, so that you can generate a trace file to see how it changed overtime.

The list of supported features are:

| Name | Collection Methods | Description |
|---|---|---|
| `acpi-dstate` | Trace | Device D-state residencies, from OS event trace. If a device has multiple components, the component F-state residency report is included.<br>The list of devices included in the report are filtered to only include selected ACPI devices. To get the complete device list, disable the filter using option `--option no-device-filter`. |
| `acpi-sstate` | Trace | ACPI Sx system state residency (S1-S4), from OS event trace. |
| `ccf-temp` | Snapshot | Converged coherent fabric temperature data. [from Intel (R) PMT] |
| `cclk-pstate` | Sampled | Core clock P-states approximated residency. [from Intel (R) PMT] |
| `ccp-volt` | Sampled | Core clock voltage. [from Intel (R) PMT] |
| `ccp-pstate` | Sampled | Converged Core Perimeter IA Module (0-4) P-states approximated residency. [from Intel (R) PMT] |
| `ccp-volt` | Sampled | Converged Core Perimeter IA Module (0-4) voltage. [from Intel (R) PMT] |
| `cdie-ccp-llc-bw` | Sampled | Converged Core Perimeter (per Core/Module) to LLC Read and Write bandwidth for Compute Die. [from Intel (R) PMT] |
| `cdie-ccp-pstate` | Sampled | CDie Converged Core Perimeter IA Module (0-7) P-states approximated residency for Compute Die. [from Intel (R) PMT] |

| Name | Collection Methods | Description |
|---|---|---|
| cdie-ccp-volt | Sampled | CDie Converged Core Perimeter IA Module (0-7) voltage for Compute Die. [from Intel (R) PMT] |
| cdie-core-cstate-cnt | Sampled | Entrance count for Core C-state C6 for Compute Die. [from Intel (R) PMT] |
| cdie-core-cstate-res | Snapshot | Residency for Core C-states (CCx) for Compute Die. [from Intel (R) PMT] |
| cdie-core-temp | Snapshot | CDie temperature data. [from Intel (R) PMT] |
| cdie-cstate-cnt | Sampled | Entrance count for Die C-states for Compute Die. [from Intel (R) PMT] |
| cdie-cstate-dbg | Snapshot | Reasons blocking entry to lower-power Die C-states and wakeup from Die C-states for Compute Die. [from Intel (R) PMT] |
| cdie-cstate-res | Snapshot | Residency for Die C-states (DCx.y) for Compute Die. [from Intel (R) PMT] |
| cdie-hammer-throt-res | Sampled | Hammer throttle residency for CDie. [from Intel (R) PMT] |
| cdie-llc-flushed-cnt | Sampled | Entrance count for LLC Flushed state for Compute Die. [from Intel (R) PMT] |
| cdie-llc-flushed-res | Snapshot | Residency for LLC Flushed state for Compute Die. [from Intel (R) PMT] |
| cdie-ltr | Static | Compute Die LTR Thresholds for states DC2.1, DC2.2, DC3.1, DC3.2, DC6. [from Intel (R) PMT] |
| cdie-perf-limit-rsn-cnt | Sampled | Compute Die performance limit reasons transition count. [from Intel (R) PMT] |
| cdie-perf-limit-rsn-res | Sampled | Compute Die performance limit reasons. [from Intel (R) PMT] |
| cdie-pwr | Sampled | Compute Die power statistics. [from Intel (R) PMT] |
| cdie-ring-cstate-cnt | Sampled | Entrance count for Ring C-state C6 for Compute Die. [from Intel (R) PMT] |
| cdie-ring-cstate-res | Snapshot | Residency for Ring C-states for Compute Die. [from Intel (R) PMT] |
| cdie-ring-pstate | Sampled | Ring P-states approximated residency for Compute Die. [from Intel (R) PMT] |
| cdie-ring-volt | Sampled | Ring voltage requirement for Compute Die. [from Intel (R) PMT] |
| cdie-vccia-pwr | Sampled | Power statistics from voltage rails for CPUs in Compute Die. [from Intel (R) PMT] |
| core-cstate-res | Snapshot | Residency for Core C-states (CCx). [from Intel (R) PMT] |

| Name | Collection Methods | Description |
|---|---|---|
| cluster-cstate-res | Snapshot | Residency for Cluster C-states. Reports residency for hybrid cores. [from Intel (R) PMT]. |
| comp-max-temp | Sampled | Maximum temperature statistics for system components, from hardware status data. Cannot be collected with edram-state or vx-pwr-state due to hardware limitations. |
| connected-standby | Connected Standby Idle Resiliency period summary, including entry/exit events and latency, plus certain hardware metrics collected at these events.<br><br>**NOTE**<br>Automatically loads *cstracedrv* driver the first time this feature is collected on the platform which requires a system enabled to run a test-signed driver. This driver is automatically reloaded after a reboot to provide the most complete session information for the connected standby analysis. See option --enable-cstrace to preload this driver. See option --disable-cstrace to unload the driver. | |
| core-auto-freq | Sampled | Reports the per core autonomous mode frequency target. [from Intel (R) PMT] |
| core-max-freq | Sampled | Reports the per core maximum allowed frequency (P-Alpha). [from Intel (R) PMT] |
| core-scale | Sampled | Reports per core scalability percentages that are calculated by the Punit as unstalled clocks divided by total active clocks. [from Intel (R) PMT] |
| core-temp | Sampled | IA core temperature statistics, from hardware status data. |
| cpu-igpu-concurrency | Snapshot | Concurrent active time of CPU and integrated GPU, from hardware accumulators. |

| Name | Collection Methods | Description |
|---|---|---|
| cpu-pkgc-cfg | Sampled | Platform settings that define CPU Package C-state entry behavior. |
| cpu-pkgc-dbg | Snapshot | Counts for reasons blocking entry to Package C-states and wakeups from Package C-states, from hardware accumulators.<br>**NOTE**<br>This data is retrieved from Intel (R) PMT on Intel Platforms starting from Meteor Lake. |
| cpu-pkgc-dbg-res | Snapshot | Residency for Package C-state blocking cause and category, and count for wake causes. [from Intel (R) PMT] |
| cpu-pkg-cstate-cnt | Sampled | Entrance count for Package C-states (PCx). [from Intel (R) PMT] |
| cpu-pkg-cstate-res | Snapshot | Residency for Package C-states (PCx). [from Intel (R) PMT] |
| d2d-pstate | Sampled | Die to Die (D2D) P-states approximated residency. [from Intel (R) PMT] |
| ddr-bw | Sampled | Total bandwidth per memory channel, from hardware accumulators.<br>The hardware accumulator data is always collected over time due to frequent overflow, so snapshot is not available. |
| ddr-comp-bw | Sampled | DDR memory bandwidth by component. [from Intel (R) PMT] |
| ddr-virtual-bw | Sampled | Total bandwidth per memory channel [from Intel (R) PMT] |
| device-acpi-calls | Trace | Summary of device ACPI call namespaces by process groups (System, Idle, Other), from OS event trace. |
| dgfx-adm-perf-limit-rsn | Snapshot | Discrete Graphics ADM Performance Limit Reasons. [from Intel (R) PMT] |
| dgfx-adm-cstate | Snapshot | Discrete Graphics Adamantine (ADM) C-state (RCx) residency. [from Intel (R) PMT] |
| dgfx-afm-cstate | Snapshot | Discrete Graphics AFM C-state (RCx) residency. [from Intel (R) PMT] |
| dgfx-audio-dstate | Snapshot | Discrete Graphics AUDIO D-state (Dx) residency. [from Intel (R) PMT] |
| dgfx-bw | Snapshot | Discrete graphics bandwidth. [from Intel (R) PMT] |
| dgfx-cstate | Snapshot | Discrete graphics C-state residency (RC6). [from Intel (R) PMT] |

| Name | Collection Methods | Description |
|---|---|---|
| dgfx-cstate-status | Sampled | Discrete graphics C-states count (RC0, RC1, RC6). [from Intel (R) PMT] |
| dgfx-current | Sampled | Discrete graphics maximal virus current for Unslice and Slice. [from Intel (R) PMT] |
| dgfx-ddr-bw | Snapshot | Discrete graphics display VC1 bandwidth. [from Intel (R) PMT] |
| dgfx-dstate | Sampled | Discrete graphics D-state residency for D3Hot and D3Cold-VRAM-SRR. [from Intel (R) PMT] |
| dgfx-display-ltr | Sampled | Discrete Graphics display latency tolerance reporting values [from Intel (R) PMT] |
| dgfx-gt-cstate | Sampled | Discrete Graphics GT C-state (RCx) residency. [from Intel (R) PMT] |
| dgfx-gt-perf-limit-rsn | Snapshot | Discrete Graphics GT Performance Limit Reasons. [from Intel (R) PMT] |
| dgfx-media-cstate | Sampled | Discrete Graphics Media C-state (RCx) residency. [from Intel (R) PMT] |
| dgfx-media-perf-limit-rsn | Snapshot | Discrete Graphics media Performance Limit Reasons. [from Intel (R) PMT] |
| dgfx-mods-res | Sampled | Discrete Graphics modern standby residency. [from Intel (R) PMT] |
| dgfx-pcie-dstate | Sampled | Discrete Graphics PCIE USP controller D-state (Dx) residency. [from Intel (R) PMT] |
| dgfx-pcie-link-state | Snapshot | Discrete Graphics PCIE link state residency. [from Intel (R) PMT] |
| dgfx-pkg-cstate | Snapshot | Discrete graphics Package C-state residency (PCx). [from Intel (R) PMT] |
| dgfx-pkg-cstate-status | Sampled | Discrete graphics C-states counts for: Package C-state (PCx), display state (DCx), PCIe state (Cx), and memory subsystem (Cx). [from Intel (R) PMT] |
| dgfx-pkg-pwr-limit-exceeded | Snapshot | Discrete graphics package power limit exceeded residency (enabled on DG2 128 EU A-Step, DG2 512 EU B-Step only). [from Intel (R) PMT] |
| dgfx-pkgc-dbg | Snapshot | Discrete graphics Package C-state reasons blocking entry to lower-power C-states and wakeup from C-states. [from Intel (R) PMT] |
| dgfx-psys-pwr-limit-exceeded | Snapshot | Discrete graphics platform power limit exceeded residency (enabled on DG2 128 EU A-Step, DG2 512 EU B-Step only). [from Intel (R) PMT] |
| dgfx-pwr | Sampled | Discrete graphics package and PSYS energy usage. [from Intel (R) PMT] |

| Name | Collection Methods | Description |
|---|---|---|
| dgfx-sgunit-dstate | Snapshot | Discrete Graphics SGUNIT D-state (Dx) residency. [from Intel (R) PMT] |
| dgfx-soc-ltr | Sampled | Discrete graphics latency tolerance values for components. [from Intel (R) PMT] |
| dgfx-temp | Sampled | Discrete graphics temperatures for: system agent (SA), image processing unit (IPU), and graphics processor (GT). [from Intel (R) PMT] |
| dgfx-vccin-pwr | Sampled | Discrete graphics VCCIN energy usage. [from Intel (R) PMT] |
| dgfx-volt | Sampled | Discrete graphics voltage values for Unslice and Slice. [from Intel (R) PMT] |
| dgfx-vram-perf-limit-rsn | Snapshot | Discrete Graphics VRAM Performance Limit Reasons. [from Intel (R) PMT] |
| dgfx-vram-srr | Snapshot | Discrete Graphics VRAM self refresh residency. [from Intel (R) PMT] |
| display-bw | Sampled | Display bandwidth Reads and Writes. [from Intel (R) PMT] |
| display-pstate | Sampled | Display (CDCLK) P-states approximated residency. [from Intel (R) PMT] |
| display-rr | Sampled | Summary of display resolution (for eDP only), link rate, and refresh rate (LRR). |
| display-state | Snapshot | Display state DC5/DC6 entrance count report, from hardware accumulators. DC6 data is not available Meteor Lake onwards. |
| display-vc1-bw | Sampled | Display VC1 bandwidth (non-coherent read). [from Intel (R) PMT] |
| display-volt | Sampled | Display voltage requirement. [from Intel (R) PMT] |
| dpst | Sampled | Summary of display panel low power settings (for eDP only), from hardware status data and OS static data. Includes Intel ® Display Power Saving Technology (DPST), Enhanced Power Savings Mode (EPSM), and south backlight duty cycle frequency percentage. |
| dram-bw | Snapshot | Total DDR bandwidth with memory slice breakdown, from hardware accumulators. Available in place of ddr-bw on platforms where hardware signals not available.<br>Supported on Intel Atom® Processor-based SoCs for systems code Broxton-M, Apollo Lake, and Gemini Lake. |
| dram-pwr | Sampled | Calculate DRAM power consumption, from PUNIT telemetry hardware accumulators. |

| Name | Collection Methods | Description |
|---|---|---|
| | | This data can be collected on Intel server platforms. |
| edram-state | Snapshot, Sampled | EDRAM active residency, from hardware accumulator and EDRAM max temperature, from hardware status data. Cannot be collected with comp-max-temp or vx-pwr-state due to hardware limitations. |
| ext-pch-ip-active | Sampled | EXT PCH active residencies for each IP, from hardware accumulators. |
| ext-pch-platform-ltr | Sampled | EXT PCH IP platform latency tolerance requirement |
| ext-pch-slps0 | Sampled | EXT PCH SLP_S0 residency, from hardware accumulator. |
| ext-pch-slps0-cfg | Sampled | EXT Die System's configuration settings for SLP_S0 entry requirements. |
| ext-pch-slps0-dbg | Sampled | EXT Die Blocking reasons for SLP_S0, from hardware status data. |
| ext-s0ix-subs-dbg | Sampled | EXT residency for meeting S0ix requirements that block entry to S0ix.y states |
| ext-s0ix-subs-res | Sampled | EXT Die system S0ix.y state residency |
| ext-s0ix-subs-req | Sampled | EXT IP requirements for entry to S0ix.y states |
| ext-s0ix-subs-status | Sampled | EXT Die Status of components in states required for entry in S0ix sub-states, from hardware status data. |
| gt-temp | Sampled | Graphics module temperature. [from Intel (R) PMT] |
| hw-cpu-hwp | Sampled | HWP capabilities, from hardware status data. |
| hw-cpu-pstate | Sampled | CPU P-state operating frequency residencies, from hardware status data. |
| hw-dgfx-pstate | Sampled | Discrete graphics processor P-state operating frequency for Unslice and Slice, from hardware converged telemetry aggregator sampler. [from Intel (R) PMT] |
| hbo-bw | Sampled | Home Agent Box (block in hardware that handles all coherent traffic including snoop and non snoop traffic) bandwidth. [from Intel (R) PMT] |
| idi-bw | Sampled | Cluster 1 (off ring cores) bandwidth. [from Intel (R) PMT] |
| hw-igfx-cstate | Snapshot* | Integrated GPU C-state residency (RC6), from hardware accumulators. *Always sampled due to short overflow time period. |
| hw-igfx-cstate-cnt | Sampled | Entrance count for integrated GPU C-state (RC6). [from Intel (R) PMT] |

| Name | Collection Methods | Description |
|---|---|---|
| hw-igfx-pstate | Sampled | Integrated GPU P-states approximated residency, from hardware status data. |
| ia-throt-rsn | Sampled | Reasons for throttling the CPU frequency, from hardware status data. |
| idi-bw | Sampled | Iceland I/O bandwidth. [from Intel (R) PMT] |
| igfx-busy | Sampled | Time integrated GPU engine truly busy. [from Intel (R) PMT] |
| igfx-pstate | Sampled | Integrated GPU P-states approximated residency. [from Intel (R) PMT] |
| igfx-rc0-idle | Sampled | Time after all streams indicated idle until RC6 entry requested. [from Intel (R) PMT] |
| igfx-rc6-entry-ovhd | Sampled | Time integrated GPU took to go through the RC6 flow (including context save) after all conditions to enter RC6 were met. [from Intel (R) PMT] |
| igfx-rc6-wake-ovhd | Sampled | Time integrated GPU spent between end of RC6 wake sequence and engine starting to execute. [from Intel (R) PMT] |
| igfx-throt-rsn | Sampled | Reasons for throttling the integrated GPU frequency, from hardware status data. |
| igfx-volt | Sampled | Integrated GPU module voltage. [from Intel (R) PMT] |
| io-bw | Sampled | Total IO bandwidth Reads and Writes. [from Intel (R) PMT]<br>**NOTE**<br>This data is retrieved from Intel (R) PMT on Intel Platforms starting from Meteor Lake. |
| ioe-pch-ip-active | Sampled | IOE PCH active residencies for each IP, from hardware accumulators. |
| ioe-pch-platform-ltr | Sampled | IOE PCH IP platform latency tolerance requirement |
| ioe-pch-slps0 | Sampled | IOE PCH SLP_S0 residency, from hardware accumulator. |
| ioe-pch-slps0-cfg | Sampled | IOE Die System’s configuration settings for SLP_S0 entry requirements. |
| ioe-pch-slps0-dbg | Sampled | IOE Die Blocking reasons for SLP_S0, from hardware status data. |
| ioe-s0ix-subs-dbg | Sampled | IOE residency for meeting S0ix requirements that block entry to S0ix.y states |
| ioe-s0ix-subs-res | Sampled | IOE Die system S0ix.y state residency |
| ioe-s0ix-subs-req | Sampled | IOE IP requirements for entry to S0ix.y states |

| Name | Collection Methods | Description |
|---|---|---|
| ioe-s0ix-subs-status | Sampled | IOE Die Status of components in states required for entry in S0ix sub-states, from hardware status data. |
| ipu-bw | Sampled | Image Processing Unit bandwidth. [from Intel (R) PMT] |
| ipu-cstate-cnt | Sampled | Entrance count for Imaging Processing Unit (IPU) Input System (IS) and Processing System (PS) C-state C6. [from Intel (R) PMT] |
| ipu-cstate-res | Snapshot | Residency for Imaging Processing Unit (IPU) Input System (IS) and Processing System (PS) C-states. [from Intel (R) PMT] |
| ipu-pstate | Sampled | Imaging Processing Unit (IPU) P-states approximated residency for Input System (IS) and Processing System (PS). [from Intel (R) PMT] |
| ipu-volt | Sampled | Imaging Processing Unit (IPU) voltage requirement for Input System (IS) and Processing System (PS). [from Intel (R) PMT] |
| ipu-state | Snapshot | Imaging Processing Unit (IPU) C6 residency for Input System (IS) and Processing System (PS). [from Intel (R) PMT] |
| llc-bw | Sampled | Per Core LLC bandwidth and GT LLC bandwidth. [from Intel (R) PMT] |
| llc-state | Snapshot | Last Level Cache residency in C3 (LLC retention) and C6 (LLC flushed and off). [from Intel (R) PMT] |
| llc-temp | Sampled | Last level Cache temperature. [from Intel (R) PMT] |
| lpss-ltr | Sampled | Low Power Subsystem (LPSS) Latency tolerance reporting (LTR) for hardware and software mechanisms, from hardware status data. |
| media-cstate-cnt | Sampled | Entrance count for Media C-state C6. [from Intel (R) PMT] |
| media-cstate-res | Snapshot | Residency for Media C-states. [from Intel (R) PMT] |
| media-pstate | Sampled | Media P-states approximated residency. [from Intel (R) PMT] |
| media-volt | Sampled | Media voltage requirement. [from Intel (R) PMT] |
| mem-state | Snapshot | Residency for low power memory states. [from Intel (R) PMT] |
| memss-pstate | Sampled | Memory subsystem (QCLK) P-states approximated residency. [from Intel (R) PMT] |
| memss-volt | Sampled | Memory subsystem voltage requirement. [from Intel (R) PMT] |
| mufasa-bw | Sampled | System cache bandwidth. [from Intel (R) PMT] |

| Name | Collection Methods | Description |
|---|---|---|
| ngu-pstate | Sampled | Next Gen Uncore (NGU) P-states approximated residency. [from Intel (R) PMT] |
| ngu-volt | Sampled | Next Gen Uncore (NGU) voltage requirements. [from Intel (R) PMT] |
| noc-cce-bw | Sampled | CCE to NOC bandwidth on Bridge0/1. [from Intel (R) PMT] |
| noc-cdie-bw | Sampled | Compute Die to NOC bandwidth on Bridge0/1. [from Intel (R) PMT] |
| noc-d2d-bw | Sampled | Die to Die I/O to NOC bandwidth on Bridge0/1. [from Intel (R) PMT] |
| noc-display-bw | Sampled | Display to NOC bandwidth on Bridge0/1. [from Intel (R) PMT] |
| noc-gcd-bw | Sampled | Graphics Die to NOC bandwidth on Bridge0/1. [from Intel (R) PMT] |
| noc-hac-bw | Sampled | Home Agent Controller (HAC) to NOC bandwidth on Bridge0/1. [from Intel (R) PMT] |
| noc-idi2cmi-bw | Sampled | ICI2CMI to NOC bandwidth on Bridge0/1. [from Intel (R) PMT] |
| noc-ioc-bw | Sampled | IO Cache to NOC bandwidth on Bridge0/1. [from Intel (R) PMT] |
| noc-ipu-bw | Sampled | Image Processing Unit (IPU)to NOC bandwidth on Bridge0/1. [from Intel (R) PMT] |
| noc-media-bw | Sampled | Media to NOC bandwidth on Bridge0/1. [from Intel (R) PMT] |
| noc-ms-bw | Sampled | Memory switch to NOC bandwidth on Bridge0/1. [from Intel (R) PMT] |
| noc-volt | Sampled | Network on Chip (NOC) voltage requirements. [from Intel (R) PMT] |
| noc-pstate | Sampled | Network on Chip (NOC) P-states approximated residency. [from Intel (R) PMT] |
| npu-bw | Sampled | NPU memory bandwidth. [from Intel (R) PMT] |
| npu-context-priority | Sampled | Residency in NPU context priority (RealTime,Normal,Focus,Idle). [from Intel (R) PMT] |
| npu-cstate-cnt | Sampled | Entrance count for NPU C-state C6. [from Intel (R) PMT] |
| npu-cstate-res | Snapshot | Residency for NPU C-states. [from Intel (R) PMT] |
| npu-dstate-cnt | Sampled | Entrance count for NPU D-states (D0/D0ix). [from Intel (R) PMT] |
| npu-dstate-res | Sampled | Residency for NPU D-states (D0/D0ix). [from Intel (R) PMT] |

| Name | Collection Methods | Description |
|---|---|---|
| npu-pstate | Sampled | NPU P-states approximated residency. [from Intel (R) PMT] |
| npu-pwr | Snapshot | NPU power statistics. [from Intel (R) PMT] |
| osreq-pkg-cstate-blocked | Snapshot | OS requested Package C-state blocked residency, as percentage of total collection duration. [from Intel (R) PMT] |
| osreq-pkg-cstate-res | Snapshot | OS requested Package C-state residency, as percentage of total collection duration. [from Intel (R) PMT] |
| panel-srr | Sampled | Panel self-refresh residency report, from hardware status data. |
| partial-slp-res | Sampled | SoC N based IP power gated residency report [from Intel (R) PMT] |
| partial-slp-cnt | Sampled | SoC N based IP power gated entrance count [from Intel (R) PMT] |
| pcd-dmi-ip-active | Snapshot | Active residencies for DMI PMA components. These residency counters do not increment once the system enters PC6 or lower. |
| pch-ip-active \| pcd-ip-active | Snapshot | PCH active residencies for each IP, from hardware accumulators.<br>To change the frequency at which this data is collected, include `--option pch-int=<milliseconds>`. Use of intervals below the default 5 seconds for PCH active metrics has higher impact on overhead and may reduce time in low power states. Use the `--interval` option for all other metrics. |
| pcd-sxix-subs-res | Snapshot | Sxix sub-state residencies from PCH \| PCD, from hardware accumulators. The sub-state residency can be reported only for S4 states.<br>This data can only be collected on Intel ® platforms code named Nova Lake and newer. |
| pch-ip-lat-limit | Snapshot | PCH IP latency limit residencies, from hardware accumulators. Exactly three IPs can be monitored at the same time. By default SPA, SPB, and SPC will be monitored.<br>To change which IPs to monitor, include `--option lat-limit-ip=<IP1, IP2, IP3>`, where each IP is one of the following: SPA, SPB, SPC, SATA, GBE, XHCI, ME, EVA, HD-AUDIO, ESPI, LPSS, CAM, SCC, ISH. The exact list of IPs can vary for different PCH. Use the `--help` option to see which IP are available on your target platform. |
| pch-ip-status | Sampled | PCH IP blocks power gated statistics, from hardware status data. |

| Name | Collection Methods | Description |
|---|---|---|
| pch-ivr-state | Snapshot | PCH Idle Voltage Reduction (IVR) residency in the idle voltage states, from hardware accumulators. |
| pch-pcieg5-ip-active \| pcd-pcieg5-ip-active | Snapshot | Active residencies for each PCIe Gen 5 PMA component in PCH/PCD chipset. These residency counters do not increment once the system enters PC6 or lower. |
| pch-platform-ltr | Sampled | PCH IP Latency Tolerance Requirement (LTR) for snoop and no snoop, from hardware status data. |
| pch-ps-on | Snapshot* | Power supply off residency, from hardware accumulator. *Always sampled due to short overflow time period. |
| pch-s0ix-subs-res \| pcd-s0ix-subs-res | Snapshot | S0ix sub-state residencies from PCH \| PCD, from hardware accumulators.<br>This data can only be collected on Intel ® platforms code named Nova Lake and newer. |
| pch-s0ix-subs-dbg \| pcd-s0ix-subs-dbg | Snapshot | S0ix sub-state debug blockers and breakers from PCH \| PCD, from hardware accumulators.<br>This data can only be collected on Intel ® platforms code named Nova Lake and newer. |
| pch-slps0\| pcd-slps0 | Snapshot | PCH SLP_S0 residency, from hardware accumulator. |
| pch-slps0-cfg | Sampled | System's configuration settings for SLP_S0 entry requirements. |
| pch-slps0-dbg | Sampled | Blocking reasons for SLP_S0, from hardware status data. |
| pcie-bw | Sampled | PCIe bandwidth. [from Intel (R) PMT] |
| pcie-lpm | Sampled | PCIe Link Power Management (LPM) Lx state residencies, from hardware status data. Includes root port and end point device capabilities. L1 sub-states are included if supported by the device (Gen 1, Gen 2, Gen 3). |
| pcie-ltr | Sampled | PCIe Latency Tolerance Reporting (LTR) for snoop and no-snoop requests, from hardware status. Includes root port and end point device capabilities. |
| pcieg5-ip-active | Snapshot | Active residencies for each PCIe Gen 5 PMA component. These residency counters do not increment once the system enters PC6 or lower. |
| pe-freq | Sampled | Power Efficient (PE) frequency for IA and GT. [from Intel (R) PMT] |
| perf-limit-rsn | Snapshot | Performance limiting reason (Thermal/Power/EDP/ Other) per IP (Core or Module, GT, LLC, IPU, Pkg) residency and PROCHOT and VRHOT residency. [from Intel (R) PMT] |

| Name | Collection Methods | Description |
|---|---|---|
| pkg-llc-flushed | Snapshot | Residency in Package LLC Flushed state. [from Intel (R) PMT] |
| pkg-llc-flushed-cnt | Sampled | Entrance count for Package LLC Flushed state. [from Intel (R) PMT] |
| pkg-pwr | Snapshot | Total SoC/Package power consumption, from hardware accumulator. |
| pkg-pwr-limit-exceeded | Snapshot | Time when package power limit (PLx) was exceeded. [from Intel (R) PMT] |
| pl-limited-res | Snapshot | Time when SoC was package power limited (PLx). [from Intel (R) PMT] |
| psys-pwr-limit-exceeded | Snapshot | Time when platform power limit (PSYS PLx) was exceeded. [from Intel (R) PMT] |
| psyspl-limited-res | Sampled | PSYS PL limited residency. [from Intel (R) PMT] |
| psys-temp | Snapshot | PSYS temperature data. [from Intel (R) PMT] |
| pwr-limits | Sampled | Reports package power limits (PL1-PL4) and platform (PSYS) power limits (PSYS PL1-PL2), Tau values (time allowed in a particular power limit), and the TjMax TCC offset. Both MSR and MMIO power limit settings are reported. This data can be collected on Intel Core® Processor-based SoCs for systems code named Tiger Lake and newer. |
| pwr-limit-locks | Sampled | Reports package and platform power limit locks and enable status information. This data can be collected on Intel Core® Processor-based SoCs for systems code named Tiger Lake and newer. |
| ring-throt-rsn | Sampled | Reasons for throttling the ring clock frequency, from hardware status data. |
| cdie-ring-volt | Sampled | Ring voltage requirement for Compute Die. [from Intel (R) PMT] |
| soc-hammer-throt-res | Sampled | Hammer throttle residency for SoC Die. [from Intel (R) PMT] |
| soc-vccia-pwr | Sampled | Power statistics from voltage rails for CPUs in SoC Die. [from Intel (R) PMT] |
| s0i3-sstate | Snapshot | S0i3 system state residencies, from PUNIT telemetry hardware accumulators.<br>This data can only be collected on Intel Atom ® Processor-based SoCs for systems code named Apollo Lake and Gemini Lake. |
| s0i3-sstate-dbg | Snapshot | Blocking reasons for entry to system state S0i3 (wakeup reasons for exit from S0i3 are included on some platforms), from PUNIT and PMC telemetry |

| Name | Collection Methods | Description |
|---|---|---|
| | | hardware accumulators. This data can only be collected on Intel Atom ® Processor-based SoCs for systems code named Apollo Lake and Gemini Lake. |
| s0ix-subs-dbg | Snapshot | Residency of components in states that are blocking entry to S0ix sub-states or causing wake-ups from S0ix sub-states, from hardware accumulators.<br>This data can only be collected on Intel ® platforms code named Tiger Lake and newer platforms.<br>Starting from Intel ® platform code named Panther Lake -H/-P, this feature will also report the IPs association with the s0ix sub-state that they block / break from including the Bus/Device/Function (B/D/F) ID's when applicable. |
| s0ix-subs-res | Snapshot | S0ix sub-state residencies, from hardware accumulators.<br>This data can only be collected on Intel ® platforms code named Lakefield, Elkhart Lake, Tiger Lake and newer. |
| s0ix-subs-req | Static | Requirements for entry to S0ix.y states for each IP, including the Bus/Device/Function (B/D/F) ID's when applicable.<br>This data can only be collected on Intel ® platforms code named Alder Lake (-P only) and newer until Lunar Lake. |
| s0ix-subs-status | Sampled | Status of components in states required for entry in S0ix sub-states, from hardware status data.<br>This data can only be collected on Intel ® platforms code named Lakefield, Elkhart Lake, Tiger Lake and newer. |
| sa-freq | Sampled | System Agent (SA) clock frequencies, from hardware status data. |
| sa-temp | Sampled | System Agent temperature, from hardware status data. This data can only be collected on Intel Atom® Processor-based SoCs for systems code named, Broxton-M, Apollo Lake and Gemini Lake. |
| santa-bw | Sampled | Cluster 0 (ring cores) bandwidth. [from Intel (R) PMT] |
| sata-lpm | Sampled | SATA Link Power Management (LPM) states residencies, from hardware status data. Includes device capabilities. |
| soc-ccp-pstate | Sampled | SoC Converged Core Perimeter IA Module 0 P-state approximated residency for SoC Die. [from Intel (R) PMT] |
| soc-ccp-volt | Sampled | SoC Converged Core Perimeter IA Module 0 voltage. [from Intel (R) PMT] |

| Name | Collection Methods | Description |
|---|---|---|
| soc-core-cstate-cnt | Sampled | Entrance count for SoC Core C6 state. [from Intel (R) PMT] |
| soc-core-cstate-res | Sampled | Residency for SoC Core C-states. [from Intel (R) PMT] |
| soc-ltr | Sampled | SOC LTR Threshold and Latency Tolerance data for Display, IPU, PCH, TypeC subsystem, and PCIe. [from Intel (R) PMT] |
| soc-pch-ip-active | Sampled | SoC PCH active residencies for each IP, from hardware accumulators. |
| soc-pch-slps0 | Sampled | SoC PCH SLP_S0 residency, from hardware accumulator. |
| soc-pch-platform-ltr | Sampled | SoC PCH IP platform latency tolerance requirement |
| soc-pch-ps-on | Snapshot* | SoC Power supply off residency, from hardware accumulator. *Always sampled due to short overflow time period. |
| soc-pch-slps0-cfg | Sampled | SoC System's configuration settings for SLP_S0 entry requirements. |
| soc-pch-slps0-dbg | Sampled | SoC Blocking reasons for SLP_S0, from hardware status data. |
| soc-perf-limit-rsn-cnt | Sampled | SoC performance limit reasons transition count. [from Intel (R) PMT] |
| soc-perf-limit-rsn-res | Sampled | SoC performance limit reasons. [from Intel (R) PMT] |
| soc-pl-limited-cnt | Sampled | SoC PL limited count. [from Intel (R) PMT] |
| soc-pl-limited-res | Sampled | SoC PL limited residency. [from Intel (R) PMT] |
| soc-psyspl-limited-cnt | Sampled | SoC PSYS PL limited count. [from Intel (R) PMT] |
| soc-psyspl-limited-res | Sampled | SoC PSYS PL limited residency. [from Intel (R) PMT] |
| soc-pwr | Sampled | SoC power statistics. [from Intel (R) PMT] |
| soc-s0ix-subs-dbg | Sampled | SoC residency for meeting S0ix requirements that block entry to S0ix.y states |
| soc-s0ix-subs-res | Sampled | SoC system S0ix.y state residency |
| soc-s0ix-subs-req | Sampled | SoC IP requirements for entry to S0ix.y states |
| soc-s0ix-subs-status | Sampled | Status of components in states required for entry in S0ix sub-states, from hardware status data. |
| soc-temp | Sampled | SoC component temperatures (e.g., IA, GT, LLC, PCH, Module, PCIe, SA, IPU, TypeC). [from Intel (R) PMT] |

| Name | Collection Methods | Description |
|---|---|---|
| tcss-state | Snapshot | TypeC subsystem state (TCx) residency and entrance count, from hardware accumulators. |
| tcss-state-res | Snapshot | TypeC subsystem state (TCx) residency, from hardware accumulators. This wakes up IOM to sample the data. To avoid IOM from waking up, use the option `--option disable-iom-wakeup`. |
| tcss-state-cnt | Snapshot | TypeC subsystem state (TCx) residency entrance counts, from hardware accumulators. |
| tcss-state-dbg | Snapshot | TypeC subsystem state (TCx) blocking time, blocking and wake causes from hardware accumulators. |
| tcss-ltr-cnt | Snapshot | Reports the number of instances when IPsTypeC subsystem achieve minimum LTR. |
| tcss-cfg-status | Sampled | TypeC subsystem (IOM, TBT) configuration status, from hardware status data. |
| timer-resolution | Trace | Timer resolution change events from OS trace events. |
| typec-bw | Sampled | TypeC Subsystem bandwidth. [from Intel (R) PMT] |
| vccia-pwr | Snapshot | VCCIA power statistics. [from Intel (R) PMT] |
| vccgt-pwr | Snapshot | VCCGT iGPU power statistics. [from Intel (R) PMT] |
| vr-temp | Snapshot | Temperature data from various voltage rails. [from Intel (R) PMT] |
| xhci-lpm | Sampled | XHCI Link Power Management state (Ux) residencies, from hardware status data. Includes capabilities of the devices directly attached to an XHCI controller operating in USB 2.0 or 3.0 modes. |

## Feature Group Names

The following features are groupings of the previously described features. These group names can be used to simplify command lines to collect multiple features concurrently. For example, `-f cpu` can replace the `-f cpu-cstate -f cpu-pstate` in a command line.

If a group includes a feature that is not enabled on the target platform, that feature will be ignored and collection continue, as long as there is one feature that can be collected.

All features are not supported on all platforms, a group will only include the supported features. Use the `--help` option on the target platform to see the list of group names and specific features included each group.

The list of supported feature groups are:

| Name | Description |
|---|---|
| bw-all | All bandwidth related metrics |
| ccp | Converged Core Perimeter (CCP) metrics |
| cdie | Compute Die (CDie) metrics |

| Name | Description |
|---|---|
| chipset-all | All PCH related metrics (all PCH active groups) and S0ix state metrics |
| core-freq-select-rsn | Collects core frequency features that form the basis for decision on the P-state frequency selected |
| cpu | cpu-hw + cpu-os |
| cpu-hw | Most CPU metrics obtained from hardware data sources |
| cpu-os | All CPU metrics obtained from OS event traces |
| cpu-pkg-cstate | Includes cpu-pkg-cstate-cnt |
| cpu-pstate | CPU P-state metrics |
| device | All I/O device metric. |
| dgfx | All discrete graphics metrics. |
| display | panel-srr+ dpst + display-rr+ display-state |
| gfx | All graphics metrics from hardware and OS, for integrated and discrete graphics processor.<br>gfx-hw + gfx-os<br>Graphics metrics for Meteor Lake. |
| gfx-hw | Most GPU metrics obtained from hardware data sources, for integrated and discrete graphics processor. |
| gfx-os | All GPU metrics obtained from OS event traces |
| hw-dgfx-cstate | dgfx-cstate + dgfx-cstate-status |
| io | Input Output (IO) device metrics. |
| ipu | Image Processing Unit (IPU) metrics |
| media | Media metrics |
| ngu | Next Gen Uncore (NGU) metrics |
| lpss | Low power subsystem I/O metrics |
| npu | Neural Processing Unit (NPU) metrics |
| npu-dstate | Neural Processing Unit (NPU) D-state metrics |
| os-req-pkg-cstate | osreq-pkg-cstate-blocked + osreq-pkg-cstate-res |
| pcie | PCIe metrics such as link state residencies and latency tolerance requirements |
| pch-ip-active | PCH active residencies for each IP, from hardware accumulators. Includes SoC, IOE and EXT dies. |
| pch-slps0-cfg | Includes SoC, IOE and EXT die SLP-S0 entry requirements |
| pch-slps0-dbg | Includes SoC, IOE and EXT die SLP-S0 blocking reasons |
| platform-ltr | Latency tolerance requirement (LTR) metrics |

| Name | Description |
|---|---|
| power | Power/energy metrics |
| pwr-limit-exceeded | Metrics reporting time when a power limit was exceeded |
| sata | SATA related metrics |
| sa-pstate | System Agent frequency and throttling reasons |
| soc-cstate | Includes SoC die and core cstate and count metrics |
| s0ix-subs-dbg | Includes SoC, IOE and EXT die s0ix debug reasons |
| s0ix-subs-req | Includes SoC, IOE and EXT die requirements for entry to S0ix.y states for each IP |
| s0ix-subs-res | Includes SoC, IOE and EXT die substate residencies |
| s0ix-subs-status | Includes SoC, IOE and EXT die substate status |
| sys | Spectrum of metrics commonly used to get general information about platform power behavior. Provides information about physical components (CPU, GPU) and not include firmware component information |
| tcss | TypeC sub-system metrics. |
| temp | Most temperature metrics |
| throt | Frequency throttling reason metric. |
| vr-pwr | Power statistics from voltage rails. |
| xhci | XHCI related metrics |

# 5 Feature Reference

This section contains detailed descriptions for the Feature Name options listed in the Options Quick Reference chapter.

## hw-cpu-cstate

Feature hw-cpu-cstate reports CPU time in C-states. C-states (aka sleep states) indicate levels of power-saving that are achieved by turning off internal function for a period of time when the CPU is not needed. C0 is the active state, as the C-state number increases so does the power savings. When the CPU needs to become active, the functions turned off need to be reversed. The time to prepare the CPU to resume the active state is *latency*. Latency is the cost of power-savings and can be seen by the platform user as sluggish response time. The higher the C-state number, the greater the power-savings and the longer the latency period for returning to C0 (active). Example: C1 saves tiny amount of power and has a tiny latency, whereas C6 saves more power and has a longer latency. Modern-day OS give hints to the platform regarding when it will need a core to be active and the hardware power management code must then chose the C-state for each Core, using logic that combines many factors it is tracking so that the Core will be in that state long enough for the power savings to exceed cost of entering/exiting and not result in latency visible as sluggish response/performance to a user interaction with the platform.

Multiple CPU cores are bundled with a GPU and memory cache into a Package for resource management. (Servers typically have multiple packages.) . So there is a Package-level C-state (PCx), in addition to the Core-level (CCx). There may also be a Module, which is a group of Atom cores and their cache, which can have its own Module-level C-states (MCx), also done for resource management. (There can be multiple Modules in a Package.) Atom cores are tuned for energy efficiency (E-Cores), whereas big cores are high performance (P-Cores). Heterogenous (aka Hybrid) platforms like Alder Lake have OS scheduling algorithms tuned to run background tasks or apps that don't require fast processing on the Atom cores.

Each CPU generation defines a set of C-states which currently range from C1 to C10 (C0 is always the active state), but not all the levels between are used, and those used for Package, Module, and Core can differ as well. For example Alder Lake defines PC2, PC3, PC6, PC7, PC8, PC9, PC10 for Package, but only CC1, CC6, CC7 for Core. There are no Module C-states, since the Atoms were not grouped for this platform.

The hw-cpu-cstate reports residency per C-state as a percentage and time value for each package, module, core on the target platform. This data comes from hardware counters that accumulate the total residency per C-state. [The C0 active time is inferred, so issues in the residency counters over/under counting show up in C0 as under/over counting. On platforms where a C1 state is valid but a counter cannot be read, the C0+C1 state must be inferred.]

If you are analyzing an idle workload, you would use this report to see if the Package entered the deepest sleep state for most of the idle period (allowing time for entry after starting the collection). On an Alder Lake, you would want to see a high PC10 residency. If not, check the core level C-states. All of the Cores in a Module must be in the same sleep state at the same time for the Module to enter that state. All of the cores in the Package must be in an idle state before the Package while try to go to a sleep state. On an Alder Lake, the Atom cores would need high residency in CC6 and the big cores in CC7. If they are not, the cores are too active for PC10 to be possible. Use the os-cpu-cstate feature to get a report on the processes that were active during a collection to see what is keeping the cores active. If the cores are sleeping, then look at the cpu-pkgc-dbg report for reasons why the deeper Package C-states were blocked or awakened from.

To see how C-state residency increased over time, include the --max-detail (-m) option during collection and choose a result format for viewing either in VTune (--result vtune) or in CSV format (--result int).

Since this data comes from an accumulator counter that will not overflow quickly, the default behavior is to only collect at the start and end of collection since there is no loss of accuracy and it reduces collection overhead. The -m option causes all features that could have been collected only at start/end to be collected at every sample point in order to see what was happening in between.

## os-cpu-cstate

This is a Windows only feature.

The os-cpu-cstate residency report shows the operating system's view of the percentage of time spent in each processor C-state per package, per module, per core, and per logical processor (when Intel HT Technology is enabled). It is based on OS trace events that log OS requests to the hardware to enter one of the ACPI defined (hardware agnostic) C-states C1, C2, or C3, which are implemented as MWAIT requests. The ACPI C-states are mapped to hardware core-specific C-states by BIOS. Typically, ACPI C1 maps to CC1, ACPI C2 will chose an intermediate core C-state or is unused, and ACPI C3 always maps to the deepest core C-state, such as CC7.

The hardware and OS residencies for C-state may or may not be similar. The hardware data shows the actual time in the C-states. The OS data shows what the OS requested, but these are considered hints, not must-do. The hardware knows much more than the OS about which state a core should try to enter, so modern OS may not try to guess and simply indicate the ACPI C1 or C3, and let the hardware make its choice. Also, auto-demotion is generally enabled, in which case the OS will not know if its sleep state request is over-ruled by the hardware. Therefore, the OS view of residency is only of interest to learn how close OS requests align with actual C-states attained. But this feature also summarizes interrupt/wake events and which processes were running during the collection - making it ideal to learn what was happening to keep the cores active (0 or low time in CC6/CC7). [Windows only] Also, running this feature with --max-detail option will result in the generation of a csv-formatted file named WakeupAnalysis. This generates extensive summary tables regarding wake/interrupt behavior of the processes running during the collection that can be used in finding poor behavior of applications that results in no time to enter deeper C-states.

Note that the OS events trace sleep requests are per logical processor. The per Core, Module, and Package level residency is extrapolated based on the understanding that the Core residency will be the shallowest of its logical processors' sleep states, the Module residency is the shallowest of its cores' sleep states, and the Package is the shallowest of its modules' sleep states (or core's sleep states if no modules). The actual residency shown in the hw-cpu-cstate report will generally be less for the deep sleep states because there are factors other than CPU processor activity that prevent entry into deeper states at the Module and Package level.

Example: The following report was generated on a platform with architecture code named XXX using the command shown. The report below shows the SOCWatchOutput.csv opened in Microsoft* Excel.

```
socwatch -t 20 -f os-cpu-cstate
```

## pch-slps0

This feature assists in determining the duration during which the system enters the deepest Package C state (PC10) and transitions into the idle sleep state (s0ix states). At this stage, the s0ix substates are achieved. While it is expected that the s0ix substate residencies (s0ix.y) align with the slp_s0 residency, observations reveal that the slps0 residency can exceed the aggregated s0ix substate residency. This discrepancy arises because the slps0 residency counter increments once the s0ix Low Power Mode (LPM) state machine is triggered, whereas the s0ix substate residencies only increment when the desired substate is achieved by the state machine.

In scenarios involving dis-aggregated dies, this is treated as a group feature. The per-die slp-s0 feature is supported. For instance, in the Meteor Lake client platform, which features two dis-aggregated PMC dies, the group feature pch-slps0 encompasses the individual features soc-pch-slps0 and ioe-pch-slps0.

## s0ix-subs-res

This feature reports the time spent in S0ix states. S0ix states are substates within the ACPI S0 state, achieved when certain conditions are met, such as the system entering a lower CPU C-state and slps0 being asserted. Since the Meteor Lake platform, the deepest low-power mode states are limited to s0i2.0, s0i2.1, and s0i2.3. The s0i2.0 state represents the base state where SoC idleness is achieved, and the latency tolerance requirements for entry include Deepest Package C exit plus chipset base S0ix wake. The s0i2.1 state includes all the conditions of s0i2.0 along with VNN removal. Finally, the s0i2.2 state indicates that most non-RTC clocks are off.

In cases of dis-aggregated PCH / PCD dies, the s0ix residency is reported per die.

# s0ix-subs-dbg

This feature reports the reasons that can prevent the system from entering deeper s0ix substates (blocker reasons) or cause wakeups from s0ix substates (breaker reasons). The data is read from hardware counters that are updated in the PMC space. It is reported as a percentage of time during which the blocker or breaker reason affected the system behavior.

The system can be blocked from entering s0ix substates or woken up from deeper s0ix substates due to the following broad reasons:

1. **Components/IP Blocks with Active Clocks:** Some components or IP blocks do not have their clocks turned off, preventing entry into deeper s0ix states.
2. **Non-Power-Gated Components/IP Blocks:** Components or IP blocks that are not power gated can hinder s0ix state transitions.
3. **D3 State Not Achieved:** Components or IP blocks that have not achieved the D3 (off) state can be a barrier to s0ix entry.
4. **VNN Removal State Not Achieved**: Failure to reach the VNN removal state can impact s0ix transitions.
5. **VNN Wake Not Achieved:** Components or IP blocks that have not achieved VNN wake status may prevent successful s0ix entry.
6. **Resource-Constrained Components/IP Blocks:** Some components or IP blocks may be resource constrained, affecting s0ix behavior.
7. **Other Miscellaneous Reasons:** Various other factors can also influence s0ix state transitions.

It’s important to note that not all reasons act as blockers or breakers for a particular s0ix state. Users are recommended to refer to the feature s0ix-subs-req, which identifies the specific reasons affecting a particular s0ix substate

In Intel platforms starting from Panther Lake, this feature only reports the components that are a blocker or a breaker to a particular s0ix state. The feature also reports the s0ix state that can be affected by the component and the B/D/F of the component.

In case of dis-aggregated PCH / PCD dies, the s0ix debug reasons are reported per die.

# npu

## npu-dstate-res

This features reports the residency of various power states the NPU FW was on during the collection. The various power states reported include

D0: This is the full operational, normal operating power state of NPU. High-speed clocks are utilized to achieve maximum performance.

D0i2 Active: In this low-power state, NPU hardware still performs tasks but at a lower power level compared to a fully active D0 state. High-speed clocks may be gated.

D0i2 Idle: In this low-power state, NPU does not process tasks actively but remains ready to resume full operation when needed. Non-essential clocks are gated to reduce power consumption.

D0i3 /D3: This is the lowest power state supported that allows NPU to conserve energy when idle. No active tasks will be performed. Most of NPU internal logic is power-gated. The residency value reported is inferred based on the other power states residency.

# *Viewing Intel SoC Watch Results with Intel® VTune™ Profiler*

6

You can analyze Intel SoC Watch data graphically using the Intel® VTune™ Profiler GUI. Intel® VTune™ Profiler provides a dynamic timeline view for interacting with Intel SoC Watch data and provides powerful filtering of data for in-depth analysis of a platform's power management behavior.

For detailed instructions, refer to the Energy Analysis section of the Intel® VTune™ Profiler Help.

# 7 Frequently Asked Questions

**What is Intel(R) SoC Watch, how to use it?**

Intel(R) SoC Watch is a platform power tool which provides insights into system power behavior such as CPU, GPU, firmware power states, bandwidth, power, debug causes.

The tool is used for platform power-performance validation, next generation modelling, platform enabling for OEM/ODM clients.

The tool support client, server and DG platforms. In general cases, the data is sampled at context switch points thereby the tool has a low overhead. Available features are specific to the target platform and system configuration.

This a command line interface tool.

**How do I use the tool for active power analysis?**

General system information can be collected by using "-f sys" option. For Discrete Graphics information, use "-f sys -f dgfx"

**How can I use the tool for debugging?**

The tool provides varied power related information. Users can collect various system and firmware related power data for analysis.

To collect general system information:

For active power debugging: Use "-f sys -f cpu -f power -f bw-all "

For low power debugging: start with "-f sys -f cpu -f chipset-all -f pcie".

Use the group switch "-f dgfx" to collect Discrete GPU related information.

**What products and OS are supported?**

The tool supports all x86 client platforms, servers, desktop SKUs, discrete graphics card. The tool can also work on various OS (windows, Linux, ChromeOS, Android (limited support)).

**What features are supported by the tool. Is this platform specific?**

The tool supports a diverse range of metrics, including cstate, firmware residency, power, bandwidth, and frequency. The available features may vary depending on the operating system, platform, and system configuration. Refer to Feature Names (Individual) for feature names and descriptions. To determine the features supported on a specific system, users should run the <help> command. The tool’s help is dynamic and will display only the features supported on the system.

**Is the tool running on an OS under virtualization supported?**

No

**Are there external dependencies to be able to use the tool?**

Intel(R) Platform monitoring technology driver is an inboxed Intel driver that provides information from telemetry aggregator via telemetry SRAM. SoC Watch queries the PMT driver to report the telemetry aggregator data.

**What is the overhead of using the tool?**

The overhead of using the SoC Watch can vary depending on several factors, including the specific metrics being collected, the duration of the collection, the sampling interval, the platform being monitored, and the system's overall workload during the collection period.

Here are some general points regarding the overhead associated with using the tool:

Sampling Interval: A shorter sampling interval can lead to higher overhead because the tool needs to wake up more frequently to collect data. Conversely, a longer interval may reduce overhead but at the cost of potentially missing short-duration events.

Number of Metrics: Collecting a large number of metrics simultaneously can increase overhead since more data needs to be gathered and processed.

Collection Duration: Longer collection periods can result in more data to be stored and processed, which can increase the overhead, especially during the post-processing phase.

System Configuration: Systems with more cores or complex configurations may experience higher overhead due to the increased amount of data to be collected and processed.

Tool Version: Different versions of the tool may have optimizations or changes that affect overhead. It's generally recommended to use the latest version of the tool for the best performance and accuracy.

Post-Processing: SoC Watch may take additional time to process the collected data after the actual monitoring period has ended. This post-processing time can be significant, especially for collections with OS event trace data.

System Impact: While SoC Watch is designed to minimize its impact on the system, it may still affect the system's power state transitions (e.g., preventing certain low-power states) due to its periodic wake-ups for data collection.

Driver and Firmware: TheSoC Watch driver and the system's firmware can also influence the overhead. For example, certain firmware versions may have bugs that affect the accuracy of the data collected, which could lead to additional overhead as the tool attempts to compensate or work around these issues.

System State: If the system is in a low-power state or performing power-sensitive operations, the overhead of running any monitoring tool, including SoC Watch, may be more noticeable.