



# NSIGHT COMPUTE

v2023.3.0 | September 2023

## Release Notes



# TABLE OF CONTENTS

<b>Chapter 1. Release Notes.....</b>	<b>1</b>
1.1. Updates in 2023.3.....	1
1.2. Updates in 2023.2.2.....	2
1.3. Updates in 2023.2.1.....	2
1.4. Updates in 2023.2.....	2
1.5. Updates in 2023.1.1.....	3
1.6. Updates in 2023.1.....	4
1.7. Updates in 2022.4.1.....	5
1.8. Updates in 2022.4.....	6
1.9. Updates in 2021.2.9.....	7
1.10. Older Versions.....	7
1.10.1. Updates in 2022.3.....	7
1.10.2. Updates in 2022.2.1.....	8
1.10.3. Updates in 2022.2.....	9
1.10.4. Updates in 2022.1.1.....	10
1.10.5. Updates in 2022.1.....	10
1.10.6. Updates in 2021.3.1.....	11
1.10.7. Updates in 2021.3.....	11
1.10.8. Updates in 2021.2.8.....	13
1.10.9. Updates in 2021.2.7.....	13
1.10.10. Updates in 2021.2.6.....	13
1.10.11. Updates in 2021.2.5.....	13
1.10.12. Updates in 2021.2.4.....	13
1.10.13. Updates in 2021.2.3.....	13
1.10.14. Updates in 2021.2.2.....	13
1.10.15. Updates in 2021.2.1.....	14
1.10.16. Updates in 2021.2.....	14
1.10.17. Updates in 2021.1.1.....	15
1.10.18. Updates in 2021.1.....	16
1.10.19. Updates in 2020.3.1.....	17
1.10.20. Updates in 2020.3.....	17
1.10.21. Updates in 2020.2.1.....	19
1.10.22. Updates in 2020.2.....	19
1.10.23. Updates in 2020.1.2.....	20
1.10.24. Updates in 2020.1.1.....	20
1.10.25. Updates in 2020.1.....	21
1.10.26. Updates in 2019.5.3.....	22
1.10.27. Updates in 2019.5.2.....	22
1.10.28. Updates in 2019.5.1.....	22
1.10.29. Updates in 2019.5.....	23

1.10.30. Updates in 2019.4.....	24
1.10.31. Updates in 2019.3.1.....	25
1.10.32. Updates in 2019.3.....	25
1.10.33. Updates in 2019.2.....	26
1.10.34. Updates in 2019.1.....	27
<b>Chapter 2. Known Issues.....</b>	<b>29</b>
<b>Chapter 3. Support.....</b>	<b>33</b>
3.1. Platform Support.....	33
3.2. GPU Support.....	34
3.3. Library Support.....	34
3.3.1. OptiX.....	34
3.4. System Requirements.....	35

## LIST OF TABLES

Table 1	Platforms supported by NVIDIA Nsight Compute .....	33
Table 2	GPU architectures supported by NVIDIA Nsight Compute .....	34

# Chapter 1.

## RELEASE NOTES

### 1.1. Updates in 2023.3

#### General

- ▶ NVIDIA Nsight Compute now supports collecting many metrics by [sampling the GPU's performance monitors \(PM\)](#) periodically at fixed intervals. The results can be visualized on a [timeline](#).
- ▶ Added WSL profiling support on Windows 10 WSL with OS build version 19044 and greater. WSL profiling is not supported on Windows 10 WSL for systems that exceed 1 TB of system memory.
- ▶ Rule outputs are prioritized to improve the accuracy of estimated speedups. The [Summary](#) page now shows the most actionable optimization advices when a result row is selected.
- ▶ Improved the handling and reporting for unavailable metrics during collection and when applying rules.

#### NVIDIA Nsight Compute

- ▶ Added support to see the source files of two profile results side by side using [Source Comparison](#). This allows you to quickly identify source differences and understand changes in metric values.
- ▶ The [Summary](#) page is now the default page when a report is opened. Previous behavior can be enabled in the [options](#) dialog.
- ▶ On the [Summary](#) and [Raw](#) pages, values from all/selected rows are automatically aggregated in the column header for applicable metrics. Selected individual cells are aggregated in the bottom status bar.
- ▶ Added **Launch Name** and **Device** options in the filter dialog launched by **Apply Filters** button in the [report header](#). This allows you to now filter results in the Summary and Raw pages.
- ▶ Added support for [source view profiles](#) that persist the Source page configuration and allow you to re-apply it to other reports.
- ▶ The [Metric Details](#) tool window now supports querying metrics beyond the current report by using the **chip:<chipname>** tag in the search.

- ▶ Added support for *CUDA Graph Edge Data* (such as port and dependency type) and *CUDA Graph Conditional Handles* in the [Resources](#) tool window.
- ▶ The [Acceleration Structure Viewer](#) and [Resources](#) tool window now support OptiX Opacity Micromaps.

### NVIDIA Nsight Compute CLI

- ▶ Tracking and profiling all child processes (`--target-processes all`) is now the default for `ncu`.
- ▶ Improved reporting of requested but unavailable metrics. Metrics requested in section files are by default considered optional and only cause a warning to be shown.

### Resolved Issues

- ▶ Support for tracking child processes launched with `system()` is available on Linux ppc64le.
- ▶ Improved the behavior of following SASS navigation links on the Source page.
- ▶ Fixed issues with profiling CUDA graphs in graph-profiling mode when nodes are associated with a non-current CUDA context.
- ▶ Fixed an issue in L2 bandwidth calculations in the hierarchical roofline sections.

## 1.2. Updates in 2023.2.2

### Resolved Issues

- ▶ Fixed possible crash when profiling CUDA graphs on multiple GPUs.
- ▶ Fixed the detection mechanism of the C2C interface, i.e. metric `c2clink__present`. The fix requires the display driver shipping with this release or any newer driver.

## 1.3. Updates in 2023.2.1

### Resolved Issues

- ▶ Fixed a crash during application replay while having the temporary directory located on a network file system (NFS).
- ▶ Improved detection mechanism for C2C interface. Added caching of the detected configuration to reduce overhead.

## 1.4. Updates in 2023.2

### General

- ▶ Extended the rules system to show estimates of the potential speedup that can be achieved by addressing the corresponding performance bottleneck. These speedups allow prioritizing applicable rules and help focusing first on optimization strategies with the highest potential performance gain.

- ▶ Added support for rules to highlight individual source lines. Lines with global/local memory access with high excessive sector counts and shared accesses with many bank conflicts are automatically detected and highlighted.
- ▶ Added the ability to query metric attributes in NvRules API.
- ▶ Added support for creating instanced metrics through the NvRules API.
- ▶ Added *sharedBankConflicts* sample CUDA application and document to show how to use NVIDIA Nsight Compute to optimize shared memory access patterns.
- ▶ For Orin+ mobile chips on the Linux aarch64 platform, added metrics (`mcc_*`) support for memory controller channel (MC Channel) unit which connects to the DRAM.

## NVIDIA Nsight Compute

- ▶ Added hyperlinks to the SASS View of the Source Page for instructions that reference others by address or offset. This enables to quickly jump to the target instruction of a branch.
- ▶ Improved the search bar in the Metric Details tool window. The search string now matches any part of the metric names, and the matching results are shown in a sorted order.
- ▶ Added a visual indication of scale of the metric value changes when the baselines are used. The background bars in the table cells of the Details Page allow to quickly identify which metrics values increased or decreased the most. The color scheme can be configured in the [Baselines tool window](#).
- ▶ Added support for rules toggle button on the Summary Page. Allows to hide the bottom pane with the rules output for the selected kernel launch.
- ▶ Added support for allowing users to configure properties on [Summary Page](#) using [Metrics/Properties profile option](#).
- ▶ Added percentage bars on [Summary Page](#).

## NVIDIA Nsight Compute CLI

- ▶ Added support for tracking child processes launched with `posix_spawn(p)` when using `--target-processes all`.
- ▶ Added support for tracking child processes launched with `system()` on Windows and Linux (aarch64, x86\_64) when using `--target-processes all`.

## Resolved Issues

- ▶ Fixed table alignment in the output of the NVIDIA Nsight Compute CLI on Windows when printing Unicode characters.
- ▶ Fixed view corruption in the Source Page after switching from the collapsed view to the expanded view.
- ▶ Fixed missing tooltip descriptions for some SASS instructions.
- ▶ Fixed potential crash when copying from the Resources tool window using CTRL+C.
- ▶ Fixed possible crash when restoring sections in the Sections tool window.

# 1.5. Updates in 2023.1.1

## NVIDIA Nsight Compute

- ▶ Added new configuration options to set the default view mode and precision for the Source page.

## Resolved Issues

- ▶ Added support for the **DT\_RUNPATH** attribute when intercepting calls to **dlopen**. Fixed issue for applications or libraries relying on **DT\_RUNPATH** not finding all dynamic libraries when launched by NVIDIA Nsight Compute.
- ▶ Improved interaction between custom additional metrics and the selected metric set. Adding custom metrics no longer forces switching to the custom metric set.
- ▶ Added ability to gracefully skip folders with insufficient access permissions while importing source code.
- ▶ Fixed the calculation of the peak values for the L1 and L2 cache bandwidths in the hierarchical roofline charts.
- ▶ Fixed issue that prevented modules loaded with function **optixModuleCreateFromPTX** showing up in the *Optix: Modules* table of the *Resources* tool window.
- ▶ Fixed handling of deprecated functions when querying function pointers from the OptiX interception library.
- ▶ Fixed that sometimes sections or rules couldn't be easily selected in the tool window.
- ▶ Fixed issue with *Reset Application Data* that prevented some setting from correctly resetting.
- ▶ Fixed potential crash of NVIDIA Nsight Compute when *Reset Application Data* was executed multiple times in a row.
- ▶ Fixed a crash when saving or loading baselines for non-kernel results.
- ▶ Fixed that memory written while executing a CUDA graph was not properly restored in single-pass graph profiling.
- ▶ Fixed potential memory leak while collecting SW counters for modules with unpatched kernel functions.

## 1.6. Updates in 2023.1

### General

- ▶ Added support for the CUDA toolkit 12.1.
- ▶ Added a new **app-range** replay mode to profile ranges without API capture by relaunching the entire application multiple times.
- ▶ Added *sharedBankConflicts* sample CUDA application and document to show how NVIDIA Nsight Compute can be used to analyze and identify the shared memory bank conflicts which result in inefficient shared memory accesses. Refer to the **README.TXT** file, sample code and document under **extras/samples/sharedBankConflicts**.
- ▶ Jupyter notebook samples are available in the Nsight training [github repository](#).
- ▶ The equivalent of the **high-level Python report interface** is now available in rule files.

### NVIDIA Nsight Compute



- ▶ Added support for profiling individual metrics in [Interactive Profile activity](#). A new input field for metrics was added in the [Metric Selection](#) tool window.
- ▶ Files on remote systems can be opened directly from the [menu](#).
- ▶ Metric- and section-related entries in the menu, [Profile activity](#) and [Metric Selection](#) tool window were renamed to make them more clear.
- ▶ CPU and GPU [NUMA topology metrics](#) can be collected on applicable systems. Topology information is shown in a new [NUMA Affinity section](#).
- ▶ Added content-aware suggestions to the Details page to provide suggestions based on the selected profiling options.
- ▶ Added support for [re-resolving source files](#) on the Source page.
- ▶ Not-issued warp stall reasons are removed from the Source Counters section tables and hidden by default on the Source page. Users should focus on regular warp stall reasons by default and only inspect not-issued samples if this distinction is needed.
- ▶ Added support to search missing CUDA source files to permanently import into the report using [Source Lookup options](#) in the [Interactive Profile activity](#).
- ▶ The [source page](#) now shows metric values as percentages by default. New buttons are added to support switching between different value modes.

### NVIDIA Nsight Compute CLI

- ▶ Added support for config files in the current working or user directory to set default ncu parameters. See the [General options](#) for more details.
- ▶ Added [--range-filter command line option](#) which allows to select subset of enabled profile ranges.
- ▶ Added new [--source-folders command line option](#) that allows to recursively search for missing CUDA source files to permanently import into the report.

### Resolved Issues

- ▶ Fixed performance issues on the Summary and Raw pages for large reports.
- ▶ Improved support for non-ASCII characters in filenames.
- ▶ Fixed an issue with delayed updates of assembly analysis information on the Source page's Source and PTX views.
- ▶ Fixed potential crashes when using the Python report interface.

## 1.7. Updates in 2022.4.1

### General

- ▶ Improved the documentation for the NvRules API.
- ▶ The python report interface links libstdc++ statically.

### Resolved Issues

- ▶ Fixed an issue that enabled profiling on CUDA Graph uploads.
- ▶ Fixed formatting issues during unit conversion of metric instances.
- ▶ Fixed an issue that could lead to a crash during application replay.
- ▶ Fixed an issue that could lead to a crash in the python report interface.
- ▶ Fixed typos in the metrics reference documentation and descriptions.

## 1.8. Updates in 2022.4

### General

- ▶ Added support for the CUDA toolkit 12.0.
- ▶ Added support for profiling [CUDA graphs](#) as complete workloads instead of as single kernel nodes. Enable this using the *Graph Profiling* option in the [activities](#). Similarly to [range replay](#) results, selected metrics are not available when profiling graphs.
- ▶ Added support for profiling on Windows Subsystem for Linux (WSL2). See the [System Requirements](#) for more details.
- ▶ Added support to query [Metric Attributes](#) in the Python report interface.
- ▶ Deprecated `nv-nsight-cu` and `nv-nsight-cu-cli` aliases are removed in favor of `ncu-ui` and `ncu`.

### NVIDIA Nsight Compute

- ▶ The [Source page](#) now loads disassembly and static analysis results asynchronously in the background.
- ▶ Added a new [Metric Details](#) tool window to inspect metric information such as raw value, unit, description or instance values. Open the tool window and select a metric on the *Details* or *Raw* page or lookup any metric in the focused report directly in the tool window's search bar.
- ▶ In the Source page PTX view, the source name will be shown as a list of comma-separated files.
- ▶ Added flexibility with NVTX based filtering in the *Next Trigger* filter, similar to the command line. Filters can now use `nvtx-include` and `nvtx-exclude` expressions by adding the `nvtx-include:` or `nvtx-exclude:` prefix.
- ▶ NVTX views now show the payload type.
- ▶ Simplified the command line generated by the [Profile activity](#).
- ▶ Reduced the number of steps required to re-run the [Profile activity](#).
- ▶ The way to rename [Baselines](#) in-place has been improved.
- ▶ The [Resources](#) tool window now shows the CUDA Dynamic Parallelism state for CUDA functions and modules.
- ▶ OptiX traversable handles can now be exported as [Graphviz](#) DOT or SVG files for visualization from the Resources tool window.
- ▶ All OptiX build, instance and geometry flags can be viewed in the [Acceleration Structure Viewer](#).
- ▶ Added OptiX-specific highlight filters to the Acceleration Structure Viewer.
- ▶ Added support for user-specified index strides to the Acceleration Structure Viewer.

### NVIDIA Nsight Compute CLI

- ▶ Added new option `--graph-profiling graph` to enable profiling of complete CUDA graphs as single workloads.
- ▶ Added new option `--filter-mode per-launch-config` to enable filtering of kernel launches for each GPU launch parameter separately.

- ▶ Added support to print section body item metrics on the details page with the new `--print-details` [command line option](#).
- ▶ Added support to select what to show in Metric Name column on the details page with the new `--print-metric-name` [command line option](#).
- ▶ Removed deprecated options: `--units`, `--fp`, `--summary` and `--kernel-base`
- ▶ Added support to print launch, session, process and device attributes on session page with the new `--page session` option.
- ▶ Added `--kill yes` support for application replay mode.

### Resolved Issues

- ▶ Fixed an issue that NVIDIA Nsight Compute could crash when continuing profiling after transposing the *Raw* page table.
- ▶ Fixed an issue that caused closing a report document to be delayed by pending source analysis.
- ▶ Fixed support for profiling applications with older OptiX versions.
- ▶ Fixed display of OptiX module inputs for IR and built-in modules.

## 1.9. Updates in 2021.2.9

### NVIDIA Nsight Compute

- ▶ Clarify when not all metrics for the roofline chart could be collected on the current chip.

## 1.10. Older Versions

### 1.10.1. Updates in 2022.3

#### General

- ▶ Added support for the CUDA toolkit 11.8.
- ▶ Added support for the Ada GPU architecture.
- ▶ Added support for the Hopper GPU architecture.
- ▶ Added support for [OptiX 7.6](#).
- ▶ Added *uncoalescedGlobalAccesses* sample CUDA application and document to show how the NVIDIA Nsight Compute profiler can be used to analyze and identify the memory accesses which are uncoalesced and result in inefficient DRAM accesses. Refer to the README, sample code and document under `extras/samples/uncoalescedGlobalAccesses`.
- ▶ Added [Metrics Reference](#) in the documentation that lists metrics not available through `--query-metrics`.
- ▶ Reduced the overhead of collecting SASS-patching based metrics.
- ▶ On [Multi-Instance GPU \(MIG\)](#) configurations, NVIDIA Nsight Compute cannot lock clocks anymore. Users are expected to lock clocks externally using `nvidia-smi`.

### NVIDIA Nsight Compute

- ▶ Wrapper script **nv-nsight-cu** is deprecated in favor of **ncu-ui** and will be removed in a future release.
- ▶ Source page supports range replay results.
- ▶ Added a second chart on the Compute Workload Analysis section to avoid mixing metrics with different meaning.
- ▶ NVIDIA Nsight Compute now tracks traversable handles created with **optixAccelRelocate**.
- ▶ NVIDIA Nsight Compute now tracks traversable handles created as updates from others.
- ▶ The Acceleration Structure viewer now reports unsupported inputs.
- ▶ The Acceleration Structure viewer now supports opening multiple traversable handles.
- ▶ The Acceleration Structure viewer now uses OptiX naming for displayed elements.

### NVIDIA Nsight Compute CLI

- ▶ Wrapper script **nv-nsight-cu-cli** is deprecated in favor of **ncu** and will be removed in a future release.
- ▶ Added new option **--filter-mode per-gpu** to enable filtering of kernel launches on each GPU separately.
- ▶ Added new option **--app-replay-mode relaxed** to produce profiling results for valid kernels even if the number of kernel launches is inconsistent across application replay passes.
- ▶ Added a documentation section on supported [environment variables](#).
- ▶ Improved the performance when loading existing reports on the command line.

### Resolved Issues

- ▶ Fixed an issue when resolving files on the Source page.
- ▶ Fixed an issue when profiling OptiX applications.
- ▶ Fixed an issue in the OptiX traversable handle management caused by clashing handle values.
- ▶ Fixed an issue in the Acceleration Structure viewer causing the display of invalid memory when viewing AABB buffers.

## 1.10.2. Updates in 2022.2.1

### Resolved Issues

- ▶ Fixed an issue that caused some tooltips to not show up for the charts on the Details page.
- ▶ Fixed the incorrect reporting of the accessed bytes for LDGSTS (access) traffic in the L1TEX memory table.
- ▶ Fixed an issue that resulted in an empty view on the Source page after resolving multiple source files at once.
- ▶ Fixed a failure to connect to remote machines over SSH due to a mismatch in the configuration of data compression.
- ▶ Fixed a potential failure to profile kernels on multiple devices of the same type of chip. The failure occurred on the attempt to profile on the second device.

## 1.10.3. Updates in 2022.2

### General

- ▶ Added support for the CUDA toolkit 11.7.
- ▶ Improved performance for profiling and metric query.
- ▶ Added Linux (aarch64 sbsa) as a supported [host platform](#).
- ▶ The NVIDIA Nsight Compute CLI stores the command line arguments, which can be viewed in the [Session](#) report page.
- ▶ Added an API to query the version of the [Python Report](#) and [NvRules](#) interfaces.
- ▶ Added an API to query the PTX in the [Python Report](#) and [NvRules](#) interfaces.

### NVIDIA Nsight Compute

- ▶ The [Acceleration Structure Viewer](#) allows inspection of acceleration structures built using the OptiX API for debugging and performance optimization.
- ▶ The [Source page](#) column chooser now supports to enable or disable groups of metrics. Note that not all metrics are enabled anymore by default to make the view easier to use.
- ▶ The [Resources](#) tool window now links to the exact target resource instances for *CUDA* resource types.
- ▶ The [Resources](#) tool window now shows the instanced nodes for *CUDA* graphs.
- ▶ The [Resources](#) tool window now shows the loading state and number of loaded functions for *CUDA Modules*.
- ▶ The [Resources](#) tool window now shows the graph node enablement state for applicable instanced graph nodes.
- ▶ The [Resources](#) tool window now shows the graph node priorities for instanced kernel graph nodes.
- ▶ Added regex support in the *Next Trigger* filter for NVTX based filtering. The *Next Trigger* filter now considers the NVTX config as a regular expression if the **regex:** prefix is specified.
- ▶ Added regex support in the report's *Filter Results* dialog.
- ▶ Added [keyboard shortcuts](#) to navigate between the pages in a report.
- ▶ The behavior for selecting sets and sections is now consistent between the [Sections/ Rules Info window](#) and the [non-interactive profile activity](#).
- ▶ Reports can now be opened directly from the welcome dialog.

### NVIDIA Nsight Compute CLI

- ▶ Added support for collecting [sampling-based](#) warp stalls in range replay mode.
- ▶ Added regex support in [NVTX filtering](#).
- ▶ The metric type is shown when querying metrics.

### Resolved Issues

- ▶ Reduced overhead of connecting to the host UI for non-interactive remote profiling sessions.
- ▶ Fixed issues with persisting the Source page state when collapsing or switching between results.

- ▶ Fixed an issue that locked GPU clocks were not reset when terminating the NVIDIA Nsight Compute CLI while profiling a kernel.
- ▶ Fixed issues with selecting and copying text from the Details page tables.
- ▶ Fixed an issue with opening report files in the UI on MacOSX.
- ▶ Fixed an issue with the *Freeze API* option.

## 1.10.4. Updates in 2022.1.1

### General

- ▶ Filtering kernel launches or profile results based on NVTX domains/ranges now takes registered strings in the payload field into account, if the range name is empty.
- ▶ Added support for the suffix `.max_rate` for ratio metrics.

### Resolved Issues

- ▶ Fixed a crash during the disassembly of the kernel's SASS code for the Source page.
- ▶ Fixed a crash on exit of the NVIDIA Nsight Compute UI.
- ▶ Fixed a hang during profiling when CPU call stack collection is enabled.
- ▶ Fixed missing to flush UVM buffers before taking memory checkpoints during [Range Replay](#).
- ▶ Fixed tracking of memory during [Range Replay](#), if the CUDA context has any device mapped memory allocations.
- ▶ Fixed the maximum available shared memory sizes in the [Occupancy Calculator](#) for NVIDIA Ampere GPUs.
- ▶ Fixed that the shared memory usage of the kernel is incorrectly initialized when opening the [Occupancy Calculator](#) from a profile report.

## 1.10.5. Updates in 2022.1

### General

- ▶ Added support for the CUDA toolkit 11.6.
- ▶ Added support for GA103 chips.
- ▶ Added a new [Range Replay](#) mode to profile ranges of multiple, concurrent kernels. Range replay is available in the NVIDIA Nsight Compute CLI and the non-interactive Profile activity.
- ▶ Added a new rule to detect non-fused floating-point instructions.
- ▶ The Uncoalesced Memory access rules now show results in a dynamic table.
- ▶ Unix Domain Sockets and Windows Named Pipes are used for local connection between the host and target processes on x86\_64 Linux and Windows, respectively.
- ▶ The [NvRules API](#) now supports querying action names using different function name bases (e.g. demangled).

### NVIDIA Nsight Compute

- ▶ The default [report page](#) is now chosen automatically when opening a report.
- ▶ Added coverage for ECC (Error Correction Code) operations in the L2 Cache table of the Memory Analysis section.

- ▶ Added a new **L2 Evict Policies** table to the Memory Analysis section.
- ▶ The **Occupancy Calculator** now updates automatically when the input changes.
- ▶ Added new metric *Thread Instructions Executed* to the Source page.
- ▶ Added tooltips to the **Register Dependency** columns in the Source page to identify the associated register more conveniently.
- ▶ Improved the selection of Sections and Sets in the Profile activity connection dialog.
- ▶ NVLink utilization is shown in the NVLink Tables section.
- ▶ NVLink links are colored according to the measured throughput.

### NVIDIA Nsight Compute CLI

- ▶ **--kernel-regex** and **--kernel-regex-base** options are no longer supported. Alternate options are **--kernel-name** and **--kernel-name-base** respectively, added in 2021.1.0.
- ▶ Added support to resolve CUDA source files in the **--page source** output with the new **--resolve-source-file** **command line option**.
- ▶ Added new option **--target-processes-filter** to filter the processes being profiled by name.
- ▶ The CPU Stack Trace is shown in the NVIDIA Nsight Compute CLI output.

### Resolved Issues

- ▶ Fixed the calculation of aggregated average instruction execution metrics in non-SASS views on the Source page.
- ▶ Fixed that atomic instructions are counted as both loads and stores in the Memory Analysis tables.

## 1.10.6. Updates in 2021.3.1

### Resolved Issues

- ▶ Fixed that kernels with the same name and launch configuration were in some scenarios associated with the wrong profiling results during application replay.
- ▶ Fixed an issue with binary forward compatibility of the report format.
- ▶ Fixed an issue with applications calling into the CUDA API during process teardown.
- ▶ Fixed an issue profiling application using pre-CUDA API 3.1 contexts.
- ▶ Fixed a crash when resolving files on the Source page.
- ▶ Fixed that opening reports with large embedded CUBINs would hang the UI.
- ▶ Fixed an issue with remote profiling on a target where the UI is already launched.

## 1.10.7. Updates in 2021.3

### General

- ▶ Added support for the CUDA toolkit 11.5.
- ▶ Added a new rule for detecting inefficient memory access patterns in the L1TEX cache and L2 cache.
- ▶ Added a new rule for detecting high usage of system or peer memory.



- ▶ Added new `IAction::sass_by_pc` function to the the [NvRules API](#).
- ▶ The [Python-based report interface](#) is now available for Windows and MacOS hosts, too.
- ▶ Added Hierarchical Roofline section files in a new "roofline" section set.
- ▶ Added support for collecting CPU call stack information.

## NVIDIA Nsight Compute

- ▶ Added support for new remote profiling [SSH connection and authentication options](#) as well as local SSH configuration files.
- ▶ Added an [Occupancy Calculator](#) which can be opened directly from a profile report or as a new activity. It offers feature parity to the CUDA Occupancy Calculator [spreadsheet](#).
- ▶ Added new [Baselines tool window](#) to manage (hide, update, re-order, save/load) baseline selections.
- ▶ The Source page views now support multi-line/cell selection and copy/paste. Different colors are used for highlighting selections and correlated lines.
- ▶ The search edit on the Source page now supports *Shift+Enter* to search in reverse direction.
- ▶ The [Memory Workload Analysis Chart](#) can be configured to show throughput values instead of transferred bytes.
- ▶ The *Profile* activity now supports the `--devices` option.
- ▶ The *NVLink Topology* diagram displays per NVLink metrics.
- ▶ Added a new tool window showing the CPU call stack at the location where the current thread was suspended during interactive profiling activities.
- ▶ If enabled, the *Call Stack / NVTX* page of the profile report shows the captured CPU call stack for the selected kernel launch.

## NVIDIA Nsight Compute CLI

- ▶ Added support for printing source/metric content with the new `--page source` and `--print-source` [command line options](#).
- ▶ Added new option `--call-stack` to enable collecting the CPU call stack for every profiled kernel launch.

## Resolved Issues

- ▶ Fixed that `memory_*` metrics could not be collected with the `--metrics` option.
- ▶ Fixed that selection and copy/paste was not supported for section header tables on the Details page.
- ▶ Fixed issues with the Source page when collapsing the content.
- ▶ Fixed that the UI could crash when applying rules to a new profile result.
- ▶ Fixed that PC Sampling metrics were not available for *Profile Series*.
- ▶ Fixed that local profiling did not work if no non-loopback address was configured for the system.
- ▶ Fixed termination of remote-launched applications. On QNX, terminating an application profiled via *Remote Launch* is now supported. Canceling remote-launched *Profile* activities is now supported.



## 1.10.8. Updates in 2021.2.8

### General

- ▶ Updated Python libraries to version 3.10.5.

## 1.10.9. Updates in 2021.2.7

### General

- ▶ Enabled stack canaries with random canary values for L4T builds.

## 1.10.10. Updates in 2021.2.6

### Resolved Issues

- ▶ Fixed an issue causing a hang on QNX after pressing **ctrl+c** while profiling a multi-process application.

## 1.10.11. Updates in 2021.2.5

### Resolved Issues

- ▶ Improve the handling of the performance monitor reservation on mobile target GPUs.

## 1.10.12. Updates in 2021.2.4

### Resolved Issues

- ▶ Fixed an issue that prevented remote interactive profiling of kernels on NVIDIA GA10b chips.

## 1.10.13. Updates in 2021.2.3

### General

- ▶ Added support for the NVIDIA GA10b chip.

### Resolved Issues

- ▶ Improved error message on QNX for failure to deploy stock section and rules files.

## 1.10.14. Updates in 2021.2.2

### General

- ▶ Changes for profiling support on NVIDIA virtual GPUs (vGPUs) for an upcoming GRID/vGPU release.

### Resolved Issues

- ▶ Fixed hang issue on QNX when using the `--target-processes all` option while profiling shell scripts.

## 1.10.15. Updates in 2021.2.1

### General

- ▶ Reduced the memory overhead when loading reports in the [Python Report Interface](#).

### Resolved Issues

- ▶ Fixed that links in the *Memory Allocations* Resource view were not working correctly.
- ▶ Fixed that NVTX state might not be correctly reset between interactive profiling activities.
- ▶ Fixed that the UI could crash when opening baselines from different GPU architectures.

## 1.10.16. Updates in 2021.2

### General

- ▶ Added support for the CUDA toolkit 11.4.
- ▶ Added support for OptiX version 7.3.
- ▶ Added support for profiling on [NVIDIA virtual GPUs](#) (vGPUs) on an upcoming GRID/vGPU release.
- ▶ Added a new [Python-based report interface](#) for interacting with report files from Python scripts.
- ▶ Added a new rule to warn users when sampling metrics were selected, but no sampling data was collected.
- ▶ Renamed *SOL* to *Throughput* in the Speed of Light section.
- ▶ Renamed several `memory_*` metrics used on the *Source* page, to better reflect the measured value. See the [Source page](#) documentation for more details.

### NVIDIA Nsight Compute

- ▶ Added support for opening [cubin files](#) in a [Standalone Source Viewer](#) without profiling the application.
- ▶ Moved the output of all rules so that it is visible even if a section's body is collapsed. Visibility of the rules' output can be toggled by a new button in the report header.
- ▶ The profiler report header now shows the report name for each baseline when ambiguous.
- ▶ Rules can define *Focused Metrics* that were most important for triggering their result output. Metrics are provided per result message which additional information, such as the underlying conditions and thresholds.
- ▶ [Memory tables](#) show tooltips for cells with derived metric calculations.
- ▶ Added a knowledge base service to show more comprehensive background information on metric names and descriptions in their tooltips.
- ▶ Following a link in the Source Counters hot spot tables automatically selects the corresponding metric in the Source page.

- ▶ Added new columns for visualizing register dependencies in the SASS view of the [Source page](#).
- ▶ Functions in the SASS view are now sorted by name.
- ▶ Added support for OptiX 7.x resource tracking in the interactive profile activity. The *Resources* tool window will show information on instantiated **optixDeviceContexts**, **optixModules**, **optixProgramGroups**, **optixPipelines** and **optixDenoiser** objects.
- ▶ Added support for new CUDA graph memory allocation APIs.
- ▶ Improved consistency between command line parameters and the *Next Trigger* filter in the API Stream window for handling of regex inputs. The *Next Trigger* filter now considers kernel/API name as a regular expression only if string has **regex:** as prefix.
- ▶ Added ability to select font settings in the options dialog.
- ▶ Added ability to configure the metrics shown on the summary page via the options dialog.
- ▶ The selected heatmap color scale now also applies to the *Memory chart*.
- ▶ The ncu-ui script now checks for missing library dependencies, such as OpenGL or Qt.

### NVIDIA Nsight Compute CLI

- ▶ Added environment variable **NV\_COMPUTE\_PROFILER\_DISABLE\_STOCK\_FILE\_DEPLOYMENT=1** to skip deployment of section and rule files.

### Resolved Issues

- ▶ Fixed a performance issue in the NVIDIA Nsight Compute CLI when using **--page raw --csv --units auto**.
- ▶ Fixed that the SSH passphrase key is no longer persisted in the project file.
- ▶ Fixed state of restore button in connection dialog. The button now supports restoring the default settings, if current setting differ from the default.
- ▶ Fixed that the complete GPU name can be shown in the NVLINK topology diagram on MacOS.
- ▶ Fixed that collapsing the Source view reset the selected metrics.
- ▶ Fixed that correlated lines could differ between filtered and unfiltered views of the executed functions.
- ▶ Fixed that two application icons were shown in the MacOS dock.
- ▶ Improved HiDPI awareness.

## 1.10.17. Updates in 2021.1.1

### General

- ▶ Updated OpenSSL library to version 1.1.1k.

### NVIDIA Nsight Compute

- ▶ Remote source resolution can now use the IP address, in addition to the hostname, to find the necessary SSH target.

## NVIDIA Nsight Compute CLI

- ▶ Added support for the existing command line options for kernel filtering while importing data from an existing report file using `--import`.
- ▶ Option `-k` is not considered as deprecated option `--kernel-regex` anymore.

## Resolved Issues

- ▶ Fixed failure to profile kernels from applications that use the CUDA graphics interop APIs to share semaphores.
- ▶ Fixed wavefront metric in the L1TEX table for writes to shared memory on GA10x chips.
- ▶ Fixed an issue resulting in incomplete data collection for the interactive profile activity after switching from single-pass mode to collecting multiple passes in the same session.
- ▶ Fixed values shown in the minimap of the Source page when all functions are collapsed.
- ▶ Fixed an issue causing names set by the NVTX naming APIs of one application to be applied to all subsequent sessions of the same instance of NVIDIA Nsight Compute.
- ▶ Fixed behavior of horizontal scroll bars when clicking in the source views on the Source page.
- ▶ Fixed appearance of multi-line entries in column chooser on the Source page.
- ▶ Fixed enablement state of the reset button on the Connection dialog.
- ▶ Fixed potential crash of NVIDIA Nsight Compute when windows size becomes small while being on the Source page.
- ▶ Fixed potential crash of NVIDIA Nsight Compute when relative paths for section/rules files could not be found.
- ▶ Fixed potential crash of NVIDIA Nsight Compute after removing baselines.

## 1.10.18. Updates in 2021.1

### General

- ▶ Added support for the CUDA toolkit 11.3.
- ▶ Added support for the [OptiX 7 API](#).
- ▶ **GpuArch** enumeration values used for filtering in section files were renamed from architecture names to compute capabilities.
- ▶ NVTX states can now be accessed via the [NvRules API](#).
- ▶ Added a rule for the *Occupancy* section.

### NVIDIA Nsight Compute

- ▶ Added support for new CUDA asynchronous allocator attributes in the *Memory Pools* resources view.
- ▶ Added a topology chart and link properties table in the NVLink section.
- ▶ The selected metric column is scrolled into view on the *Source* page when a new metric is selected.
- ▶ Users can choose the *Source* heatmap color scale in the *Options* dialog.

## NVIDIA Nsight Compute CLI

- ▶ Added file-based **application replay** as the new default application replay mode. File-based replay uses a temporary file for keeping replay data, instead of allocating them in memory. This keeps the required memory footprint close to constant, independent of the number of profiled kernels. Users can switch between buffer modes using the `--app-replay-buffer` option.
- ▶ CLI output now shows NVTX color and message information.
- ▶ `--kernel-regex` and `--kernel-regex-base>` options are deprecated and replaced by `--kernel-name` and `--kernel-regex-base`, respectively.
- ▶ All options which support regex need to provide **regex:** as a prefix before an argument to match per the regex, e.g `<option> <regex:expression>`

### Resolved Issues

- ▶ Fixed that baselines were not updated properly on the *Comments* page.
- ▶ Fixed that NVTX ranges named using their payloads can be used in **NVTX filtering** expressions.
- ▶ Fixed crashes in MacOSX hosts when terminating the target application.
- ▶ The NVLINK(**nv1\***) metrics are now added back.

## 1.10.19. Updates in 2020.3.1

### General

- ▶ Added support for LDSM instruction-level metrics.

### NVIDIA Nsight Compute

- ▶ LDSM instruction-level metrics are shown in the *Source* page and memory tables.
- ▶ Improved reporting and documentation for collecting *Profile Series*.
- ▶ Frozen columns in the *Source* page are automatically scrolled into view.

### Resolved Issues

- ▶ Fixed an issue when profiling multi-threaded applications.
- ▶ Fixed an issue that NVIDIA Nsight Compute would not automatically restart when using *Reset Application Data*.
- ▶ Fixed issues with target applications using libstdc++.
- ▶ Fixed an issue when collecting single-pass metrics in multiple Nsight Compute instances.
- ▶ Fixed an issue when using *Kernel ID* and setting *Launch Capture Count* as non-zero in the UI's *Profile* activity.
- ▶ Fixed an issue that prevented different users on the same Linux system to use NVIDIA Nsight Compute in shared instance mode.
- ▶ Fixed an issue that prevented resources from being properly renamed using NVTX information in the UI.

## 1.10.20. Updates in 2020.3

### General

- ▶ Added support for *derived metrics* in section files. Derived metrics can be used to create new metrics based on existing metrics and constants. See the [Customization Guide](#) for details.
- ▶ Added a new *Import Source* (**--import-source**) option to the UI and command line to permanently import source files into the report, when available.
- ▶ Added a new section that shows selected *NVLink* metrics on supported systems.
- ▶ Added a new **launch\_\_func\_\_cache\_\_config** metric to the *Launch Statistics* section.
- ▶ Added new branch efficiency metrics to the *Source Counters* section, including **smsp\_\_sass\_average\_branch\_targets\_threads\_uniform.pct** to replace nvprof's **branch\_efficiency**, as well as instruction-level metrics **smsp\_\_branch\_targets\_threads\_divergent**, **smsp\_\_branch\_targets\_threads\_uniform** and **branch\_inst\_executed**.
- ▶ A warning is shown if kernel replay starts staging GPU memory to CPU memory or the file system.
- ▶ Section and rule files are deployed to a versioned directory in the user's home directory to allow easier editing of those files, and to prevent modifying the base installation.
- ▶ Removed support for NVLINK(**nv1\***) metrics due to a potential application hang during data collection. The metrics will be added back in a future version of the driver/tool.

## NVIDIA Nsight Compute

- ▶ Added support for *Profile Series*. Series allow you to profile a kernel with a range of configurable parameters to analyze the performance of each combination.
- ▶ Added a new *Allocations* view to the *Resources* tool window which shows the state of all current memory allocations.
- ▶ Added a new *Memory Pools* view to the *Resources* tool window which shows the state of all current memory pools.
- ▶ Added coverage of peer memory to the *Memory Chart*.
- ▶ The *Source* page now shows the number of excessive sectors requested from L1 or L2, e.g. due to uncoalesced memory accesses.
- ▶ The *Source* column on the *Source* page can now be scrolled horizontally.
- ▶ The kernel duration **gpu\_\_time\_duration.sum** was added as column on the *Summary* page.
- ▶ Improved the performance of *application replay* when not all kernels in the application are profiled.

## NVIDIA Nsight Compute CLI

- ▶ Added a new **--app-replay-match** option to select the mechanism used for matching kernel instances across application replay passes.
- ▶ An error is shown if **--nvtx-include/exclude** are used without **--nvtx**.

## Resolved Issues

- ▶ The *Grid Size* column on the *Raw* page now shows the CUDA grid size like the *Launch Statistics* section, rather than the combined grid and block sizes.
- ▶ The *Branch Resolving* warp stall reason was added to the PC sampling metric groups and the *Warp State Statistics* section.

- ▶ The *API Stream* tool window shows kernel names according to the selected Function Name Mode.
- ▶ Fixed that an incorrect line could be shown after a heatmap selection on the *Source* page.
- ▶ Fixed incorrect metric usage for system memory in the *Memory Chart*. Previously, all requested memory of L2 from system memory was reported instead of only the portion that missed in L2.

## 1.10.21. Updates in 2020.2.1

### Resolved Issues

- ▶ Fixed several issues related to auto-profiling in the UI.
- ▶ Fixed a metric collection issue when profiling kernels on different GPU architectures with application replay.
- ▶ Fixed a performance problem related to profiling large process trees.
- ▶ Fixed that occupancy charts would not render correctly when comparing against baselines.
- ▶ Fixed that no memory metrics were shown on the *Source* page for **LDGSTS** instructions.
- ▶ Fixed the automatic sorting on the *Summary* and *Raw* pages.
- ▶ Fixed an issue that would cause the NVIDIA Nsight Compute CLI to consume too much memory when importing or printing reports.
- ▶ Long kernel names are now elided in the *Details* page source hot spot tables.
- ▶ Fixed that function names in the *Resources* tool window were demangled differently.

## 1.10.22. Updates in 2020.2

### General

- ▶ Added support for the NVIDIA Ampere GPUs with compute capability 8.6 and CUDA toolkit 11.1.
- ▶ Added support for application replay to collect metric results across multiple application runs, instead of replaying individual kernels.
- ▶ Added new **launch\_\_device\_id** metric.
- ▶ Added support for NVLink (**nv1\***) metrics for GPUs with compute capabilities 7.0, 7.5 and 8.0
- ▶ Added documentation for memory charts and tables in the [Profiling Guide](#).

### NVIDIA Nsight Compute

- ▶ Updated menu and toolbar layout.
- ▶ Added support for zoom and pan on roofline charts.
- ▶ The *Resources* tool window shows the current CUDA stream attributes.
- ▶ The memory chart shows a heatmap for link and port utilization.
- ▶ The hot-spot tables in the *Source Counters* section now show values as percentages, too.
- ▶ On-demand resolve of remote CUDA-C source is now available for MacOS hosts.



- ▶ Metric columns in the *Summary* and *Raw* pages are now sortable.
- ▶ Added a new option to set the number of recent API calls shown in the *API Stream* tool window.

### NVIDIA Nsight Compute CLI

- ▶ CLI output now shows NVTX payload information.
- ▶ CSV output now shows NVTX states.
- ▶ Added a new **--replay-mode** option to select the mechanism used for replaying a kernel launch multiple times.
- ▶ Added a new **--kill** option to terminate the application once all requested kernels were profiled.
- ▶ Added a new **--log-file** option to decide the output stream for printing tool output.
- ▶ Added a new **--check-exit-code** option to decide if the child application exit code should be checked.

### Resolved Issues

- ▶ The profiling progress dialog is not dismissed automatically anymore after an error.
- ▶ The inter-process lock is now automatically given write permissions for all users.
- ▶ All project extensions are enabled in the default dialog filter.
- ▶ Fixed handling of targets using *tcsch* during remote profiling.
- ▶ Fixed handling of quoted application arguments on Windows.

## 1.10.23. Updates in 2020.1.2

### General

- ▶ The NVIDIA Nsight Compute installer for Mac is now code-signed and notarized.
- ▶ Disabled the creation of the Python cache when executing rules to avoid permission issues and signing conflicts.

### Resolved Issues

- ▶ Fixed the launcher script of the NVIDIA Nsight Compute CLI to no longer fail if **uname -p** is not available.
- ▶ Fixed the API parameter capture for function **cuDeviceGetLuid**.

## 1.10.24. Updates in 2020.1.1

### General

- ▶ Metrics passed to **--metrics** on the NVIDIA Nsight Compute CLI or in the respective *Profile* activity option are automatically expanded to all first-level sub-metrics if required. See the documentation on **--metrics** for more details.
- ▶ Added new rules for detecting inefficiencies of using the sparse data compression on the NVIDIA Ampere architecture.
- ▶ The version of the NVIDIA Nsight Compute target collecting the results is shown in the *Session* page.



- ▶ Added new `launch__grid_dim_[x,y,z]` and `launch__block_dim_[x,y,z]` metrics.

### NVIDIA Nsight Compute

- ▶ The *Break on API Error* functionality has been improved when auto profiling.

### NVIDIA Nsight Compute CLI

- ▶ The full path to the report output file is printed after profiling.
- ▶ Added and corrected metrics in the *nvprof Metric Comparison* table.

### Resolved Issues

- ▶ Documented the *breakdown:* metrics prefix.
- ▶ Fixed handling of escaped domain delimiters in NVTX filter expressions.
- ▶ Fixed issues with the occupancy charts for small block sizes.
- ▶ Fixed an issue when choosing a default report page in the options dialog.
- ▶ Fixed that the scroll bar could overlap the content when exporting the report page as an image.

## 1.10.25. Updates in 2020.1

### General

- ▶ Added support for the NVIDIA GA100/SM 8.x GPU architecture
- ▶ Removed support for the Pascal SM 6.x GPU architecture
- ▶ Windows 7 is not a supported host or target platform anymore
- ▶ Added a rule for reporting uncoalesced memory accesses as part of the *Source Counters* section
- ▶ Added support for report name placeholders %p, %q, %i and %h
- ▶ The [Kernel Profiling Guide](#) was added to the documentation

### NVIDIA Nsight Compute

- ▶ The UI command was renamed from `nv-nsight-cu` to `ncu-ui`. Old names remain for backwards compatibility.
- ▶ Added support for roofline analysis charts
- ▶ Added linked hot spot tables in section bodies to indicate performance problems in the source code
- ▶ Added section navigation links in rule results to quickly jump to the referenced section
- ▶ Added a new option to select how kernel names are shown in the UI
- ▶ Added new memory tables for the L1/TEX cache and the L2 cache. The old tables are still available for backwards compatibility and moved to a new section containing deprecated UI elements.
- ▶ Memory tables now show the metric name as a tooltip
- ▶ Source resolution now takes into account file properties when selecting a file from disk
- ▶ Results in the profile report can now be filtered by NVTX range
- ▶ The Source page now supports collapsing views even for single files

- ▶ The UI shows profiler error messages as dismissible banners for increased visibility
- ▶ Improved the baseline name control in the profiler report header

### NVIDIA Nsight Compute CLI

- ▶ The CLI command was renamed from **nv-nsight-cu-cli** to **ncu**. Old names remain for backwards compatibility.
- ▶ Queried metrics on GV100 and newer chips are sorted alphabetically
- ▶ Multiple instances of NVIDIA Nsight Compute CLI can now run concurrently on the same system, e.g. for profiling individual MPI ranks. Profiled kernels are serialized across all processes using a system-wide file lock.

### Resolved Issues

- ▶ More C++ kernel names can be properly demangled
- ▶ Fixed a **free() : invalid pointer** error when profiling applications using `pytorch > 19.07`
- ▶ Fixed profiling IBM Spectrum MPI applications that require PAMI GPU hooks (`--smpiargs="-gpu"`)
- ▶ Fixed that the first kernel instruction was missed when computing **sass\_\_inst\_executed\_per\_opcode**
- ▶ Reduced surplus DRAM write traffic created from flushing caches during kernel replay
- ▶ The *Compute Workload Analysis* section shows the IMMA pipeline on GV11b GPUs
- ▶ Profile reports now scroll properly on MacOS when using a trackpad
- ▶ Relative output filenames for the Profile activity now use the document directory, instead of the current working directory
- ▶ Fixed path expansion of `~` on Windows
- ▶ Memory access information is now shown properly for RED assembly instructions on the Source page
- ▶ Fixed that user **PYTHONHOME** and **PYTHONPATH** environment variables would be picked up by NVIDIA Nsight Compute, resulting in locale encoding issues.

## 1.10.26. Updates in 2019.5.3

### General

- ▶ More C++ kernel names can be properly demangled

## 1.10.27. Updates in 2019.5.2

### General

- ▶ Bug fixes

## 1.10.28. Updates in 2019.5.1

### General

- ▶ Added support for Nsight Compute Visual Studio Integration

## 1.10.29. Updates in 2019.5

### General

- ▶ Added *section sets* to reduce the default overhead and make it easier to configure metric sets for profiling
- ▶ Reduced the size of the installation
- ▶ Added support for CUDA Graphs Recapture API
- ▶ The NvRules API now supports accessing correlation IDs for instanced metrics
- ▶ Added breakdown tables for *SOL SM* and *SOL Memory* in the Speed Of Light section for Volta+ GPUs

### NVIDIA Nsight Compute

- ▶ Added a snap-select feature to the Source page heatmap help navigate large files
- ▶ Added support for loading remote CUDA-C source files via SSH on demand for Linux x86\_64 targets
- ▶ Charts on the Details page provide better help in tool tips when hovering metric names
- ▶ Improved the performance of the Source page when scrolling or collapsing
- ▶ The charts for Warp States and Compute pipelines are now sorted by value

### NVIDIA Nsight Compute CLI

- ▶ Added support for GPU cache control, see **--cache-control**
- ▶ Added support for setting the kernel name base in command line output, see **--kernel-base**
- ▶ Added support for listing the available names for **--chips**, see **--list-chips**
- ▶ Improved the stability on Windows when using **--target-processes all**
- ▶ Reduced the profiling overhead for small metric sets in applications with many kernels

### Resolved Issues

- ▶ Reduced the overhead caused by demangling kernel names multiple times
- ▶ Fixed an issue that kernel names were not demangled in CUDA Graph Nodes resources window
- ▶ The connection dialog better disables unsupported combinations or warns of invalid entries
- ▶ Fixed metric *thread\_inst\_executed\_true* to derive from *smsp\_not\_predicated\_off\_thread\_inst\_executed* on Volta+ GPUs
- ▶ Fixed an issue with computing the theoretical occupancy on GV100
- ▶ Selecting an entry on the Source page heatmap no longer selects the respective source line, to avoid losing the current selection
- ▶ Fixed the current view indicator of the Source page heatmap to be line-accurate
- ▶ Fixed an issue when comparing metrics from Pascal and later architectures on the Summary page
- ▶ Fixed an issue that metrics representing constant values on Volta+ couldn't be collected without non-constant metrics

## 1.10.30. Updates in 2019.4

### General

- ▶ Added support for the Linux PowerPC target platform
- ▶ Reduced the profiling overhead, especially if no source metrics are collected
- ▶ Reduced the overhead for non-profiled kernels
- ▶ Improved the deployment performance during remote launches
- ▶ Trying to profile on an unsupported GPU now shows an "Unsupported GPU" error message
- ▶ Added support for the `%i` sequential number placeholder to generate unique report file names
- ▶ Added support for `smsp__sass_*` metrics on Volta and newer GPUs
- ▶ The `launch__occupancy_limit_shared_mem` now reports the device block limit if no shared memory is used by the kernel

### NVIDIA Nsight Compute

- ▶ The *Profile* activity shows the command line used to launch `ncu`
- ▶ The heatmap on the Source page now shows the represented metric in its tooltip
- ▶ The *Memory Workload Analysis Chart* on the Details page now supports baselines
- ▶ When applying rules, a message displaying the number of new rule results is shown in the status bar
- ▶ The Visual Profiler Transition Guide was added to the documentation
- ▶ Connection dialog activity options were added to the documentation
- ▶ A warning dialog is shown if the application is resumed without Auto-Profile enabled
- ▶ Pausing the application now has immediate feedback in the toolbar controls
- ▶ Added a *Close All* command to the *File* menu

### NVIDIA Nsight Compute CLI

- ▶ The `--query-metrics` option now shows only metric base names for faster metric query. The new option `--query-metrics-mode` can be used to display the valid suffixes for each base metric.
- ▶ Added support for passing response files using the `@` operator to specify command line options through a file

### Resolved Issues

- ▶ Fixed an issue that reported the wrong executable name in the Session page when attaching
- ▶ Fixed issues that chart labels were shown elided on the Details page
- ▶ Fixed an issue that caused the cache hit rates to be shown incorrectly when baselines were added
- ▶ Fixed an illegal memory access when collecting `sasp__*_histogram` metrics for applications using PyTorch on Pascal GPUs
- ▶ Fixed an issue when attempting to collect all `smsp__*` metrics on Volta and newer GPUs

- ▶ Fixed an issue when profiling multi-context applications
- ▶ Fixed that profiling start/stop settings from the connection dialog weren't properly passed to the interactive profile activity
- ▶ Fixed that certain *smsp\_\_warp\_cycles\_per\_issue\_stall\** metrics returned negative values on Pascal GPUs
- ▶ Fixed that metric names were truncated in the **--page details** non-CSV command line output
- ▶ Fixed that the target application could crash if a connection port was used by another application with higher privileges

## 1.10.31. Updates in 2019.3.1

### NVIDIA Nsight Compute

- ▶ Added ability to send bug reports and suggestions for features using *Send Feedback* in the *Help* menu

### Resolved Issues

- ▶ Fixed calculation of theoretical occupancy for grids with blocks that are not a multiple of 32 threads
- ▶ Fixed intercepting child processes launched through Python's `subprocess.Popen` class
- ▶ Fixed issue of NVTX push/pop ranges not showing up for child threads in NVIDIA Nsight Compute CLI
- ▶ Fixed performance regression for metric lookups on the Source page
- ▶ Fixed description in rule covering the IMC stall reason
- ▶ Fixed cases where baseline values were not correctly calculated in the Memory tables when comparing reports of different architectures
- ▶ Fixed incorrect calculation of baseline values in the Executed Instruction Mix chart
- ▶ Fixed accessing instanced metrics in the NvRules API
- ▶ Fixed a bug that could cause the collection of unnecessary metrics in the Interactive Profile activity
- ▶ Fixed potential crash on exit of the profiled target application
- ▶ Switched underlying metric for **SOL FB** in the GPU Speed Of Light section to be driven by **dram\_\_throughput.avg.pct\_of\_peak\_sustained\_elapsed** instead of **fbpa\_\_throughput.avg.pct\_of\_peak\_sustained\_elapsed**

## 1.10.32. Updates in 2019.3

### General

- ▶ Improved performance
- ▶ Bug fixes
- ▶ Kernel launch context and stream are reported as metrics
- ▶ PC sampling configuration options are reported as metrics
- ▶ The default base port for connections to the target changed
- ▶ Section files support multiple, named Body fields

- ▶ NvRules allows users to query metrics using any convertible data type

### NVIDIA Nsight Compute

- ▶ Support for filtering kernel launches using their NVTX context
- ▶ Support for new options to select the connection port range
- ▶ The Profile activity supports configuring PC sampling parameters
- ▶ Sections on the Details page support selecting individual bodies

### NVIDIA Nsight Compute CLI

- ▶ Support for stepping to kernel launches from specific NVTX contexts
- ▶ Support for new `--port` and `--max-connections` options
- ▶ Support for new `--sampling-*` options to configure PC sampling parameters
- ▶ Section file errors are reported with `--list-sections`
- ▶ A warning is shown if some section files could not be loaded

### Resolved Issues

- ▶ Using the `--summary` option works for reports that include invalid metrics
- ▶ The full process executable filename is reported for QNX targets
- ▶ The project system now properly stores the state of opened reports
- ▶ Fixed PTX syntax highlighting
- ▶ Fixed an issue when switching between manual and auto profiling in NVIDIA Nsight Compute
- ▶ The source page in NVIDIA Nsight Compute now works with results from multiple processes
- ▶ Charts on the NVIDIA Nsight Compute details page uses proper localization for numbers
- ▶ NVIDIA Nsight Compute no longer requires the system locale to be set to English

## 1.10.33. Updates in 2019.2

### General

- ▶ Improved performance
- ▶ Bug fixes
- ▶ Kernel launch context and stream are reported as metrics
- ▶ PC sampling configuration options are reported as metrics
- ▶ The default base port for connections to the target changed
- ▶ Section files support multiple, named Body fields
- ▶ NvRules allows users to query metrics using any convertible data type

### NVIDIA Nsight Compute

- ▶ Support for filtering kernel launches using their NVTX context
- ▶ Support for new options to select the connection port range
- ▶ The Profile activity supports configuring PC sampling parameters
- ▶ Sections on the Details page support selecting individual bodies

### NVIDIA Nsight Compute CLI

- ▶ Support for stepping to kernel launches from specific NVTX contexts
- ▶ Support for new **--port** and **--max-connections** options
- ▶ Support for new **--sampling-\*** options to configure PC sampling parameters
- ▶ Section file errors are reported with **--list-sections**
- ▶ A warning is shown if some section files could not be loaded

### Resolved Issues

- ▶ Using the **--summary** option works for reports that include invalid metrics
- ▶ The full process executable filename is reported for QNX targets
- ▶ The project system now properly stores the state of opened reports
- ▶ Fixed PTX syntax highlighting
- ▶ Fixed an issue when switching between manual and auto profiling in NVIDIA Nsight Compute
- ▶ The source page in NVIDIA Nsight Compute now works with results from multiple processes
- ▶ Charts on the NVIDIA Nsight Compute details page uses proper localization for numbers
- ▶ NVIDIA Nsight Compute no longer requires the system locale to be set to English

## 1.10.34. Updates in 2019.1

### General

- ▶ Support for CUDA 10.1
- ▶ Improved performance
- ▶ Bug fixes
- ▶ Profiling on Volta GPUs now uses the same metric names as on Turing GPUs
- ▶ Section files support descriptions
- ▶ The default sections and rules directory has been renamed to *sections*

### NVIDIA Nsight Compute

- ▶ Added new profiling options to the options dialog
- ▶ Details page shows rule result icons in the section headers
- ▶ Section descriptions are shown in the details page and in the sections tool window
- ▶ Source page supports collapsing multiple source files or functions to show aggregated results
- ▶ Source page heatmap color scale has changed
- ▶ Invalid metric results are highlighted in the profiler report
- ▶ Loaded section and rule files can be opened from the sections tool window

### NVIDIA Nsight Compute CLI

- ▶ Support for profiling child processes on Linux and Windows x86\_64 targets
- ▶ NVIDIA Nsight Compute CLI uses a temporary file if no output file is specified
- ▶ Support for new **--quiet** option
- ▶ Support for setting the GPU clock control mode using new **--clock-control** option

- ▶ Details page output shows the NVTX context when `--nvtx` is enabled
- ▶ Support for filtering kernel launches for profiling based on their NVTX context using new `--nvtx-include` and `--nvtx-exclude` options
- ▶ Added new `--summary` options for aggregating profiling results
- ▶ Added option `--open-in-ui` to open reports collected with NVIDIA Nsight Compute CLI directly in NVIDIA Nsight Compute

### Resolved Issues

- ▶ Installation directory scripts use absolute paths
- ▶ OpenACC kernel names are correctly demangled
- ▶ Profile activity report file supports a relative path
- ▶ Source view can resolve all applicable files at once
- ▶ UI font colors are improved
- ▶ Details page layout and label elision issues are resolved
- ▶ Turing metrics are properly reported on the Summary page
- ▶ All byte-based metrics use a factor of 1000 when scaling units to follow SI standards
- ▶ CSV exports properly align columns with empty entries
- ▶ Fixed the metric computation for `double_precision_fu_utilization` on GV11b
- ▶ Fixed incorrect 'selected' PC sampling counter values
- ▶ The SpeedOfLight section uses 'max' instead of 'avg' cycles metrics for Elapsed Cycles



# Chapter 2.

## KNOWN ISSUES

### Installation

- ▶ The installer might not show all patch-level version numbers during installation.
- ▶ Some command line options listed in the help of a `.run` installer of NVIDIA Nsight Compute are affecting only the archive extraction, but not the installation stage. To pass command line options to the embedded installer script, specify those options after `--` in the form of `-- -<option>`. The available options for the installer script are:

```
-help           : Print help message
-targetpath=<PATH> : Specify install path
-noprompt       : No prompts. Implies acceptance of the EULA
```

For example, specifying only option `--quiet` extracts the installer archive without any output to the console, but still prompts for user interaction during the installation. To install NVIDIA Nsight Compute without any console output nor any user interaction, please specify `--quiet -- -noprompt`.

- ▶ After using the SDK Manager to install the NVIDIA Nsight Compute tools, their binary path needs to be manually added to your `PATH` environment variable.
- ▶ See also the [System Requirements](#) for more installation instructions.

### Launch and Connection

- ▶ Launching applications on remote targets/platforms is not supported for several combinations. See [Platform Support](#) for details. Manually launch the application using command line `ncu --mode=launch` on the remote system and connect using the UI or CLI afterwards.
- ▶ In the NVIDIA Nsight Compute connection dialog, a remote system can only be specified for one target platform. Remove a connection from its current target platform in order to be able to add it to another.
- ▶ Loading of CUDA sources via SSH requires that the remote connection is configured, and that the hostname/IP address of the connection matches the target (as seen in the report session details). For example, prefer `my-machine.my-domain.com`, instead of `my-machine`, even though the latter resolves to the same.

- ▶ Other issues concerning remote connections are discussed in the documentation for [remote connections](#).
- ▶ Local connections between NVIDIA Nsight Compute and the launched target application might not work on some ppc64le or aarch64 (sbsa) systems configured to only support IPv6. On these platforms, the `NV_COMPUTE_PROFILER_LOCAL_CONNECTION_OVERRIDE=uds` environment variable can be set to use *Unix Domain Sockets* instead of *TCP* for local connections to workaround the problem. On x86\_64 Linux, Unix Domain Sockets are used by default, but local TCP connections can be forced using `NV_COMPUTE_PROFILER_LOCAL_CONNECTION_OVERRIDE=tcp`.

## Profiling and Metrics

- ▶ Profiling of 32-bit processes is not supported.
- ▶ Profiling kernels executed on a device that is part of an SLI group is not supported. An "Unsupported GPU" error is shown in this case.
- ▶ Profiling a kernel while other contexts are active on the same device (e.g. X server, or secondary CUDA or graphics application) can result in varying metric values for L2/FB (Device Memory) related metrics. Specifically, L2/FB traffic from non-profiled contexts cannot be excluded from the metric results. To completely avoid this issue, profile the application on a GPU without secondary contexts accessing the same device (e.g. no X server on Linux).
- ▶ In the current release, profiling a kernel while any other GPU work is executing on the same MIG compute instance can result in varying metric values for all units. NVIDIA Nsight Compute enforces serialization of the CUDA launches within the target application to ensure those kernels do not influence each other. See [Serialization](#) for more details. However, GPU work issued through other APIs in the target process or workloads created by non-target processes running simultaneously in the same MIG compute instance will influence the collected metrics. Note that it is acceptable to run CUDA processes in other MIG compute instances as they will not influence the profiled MIG compute instance.
- ▶ On Linux kernels settings `fs.protected_regular=1` (e.g. some Ubuntu 20.04 cloud service provider instances), root users may not be able to access the [inter-process lock file](#). See the [FAQ](#) for workarounds.
- ▶ Profiling only supports up to 32 device instances, including instances of MIG partitions. Profiling the 33rd or higher device instance will result in indeterminate data.
- ▶ Enabling certain metrics can cause GPU kernels to run longer than the driver's watchdog time-out limit. In these cases the driver will terminate the GPU kernel resulting in an application error and profiling data will not be available. Please disable the driver watchdog time out before profiling such long running CUDA kernels.
  - ▶ On Linux, setting the X Config option Interactive to false is recommended.
  - ▶ For Windows, detailed information on disabling the Windows TDR is available at <https://docs.microsoft.com/en-us/windows-hardware/drivers/display/timeout-detection-and-recovery>
- ▶ Collecting device-level metrics, such as the NVLink metrics (`nv1*`), is not supported on [NVIDIA virtual GPUs \(vGPUs\)](#).

- ▶ As of CUDA 11.4 and R470 TRD1 driver release, NVIDIA Nsight Compute is supported in a vGPU environment which requires a vGPU license. If the license is not obtained after 20 minutes, the reported performance metrics data from the GPU will be inaccurate. This is because of a feature in vGPU environment which reduces performance but retains functionality as specified [here](#).
- ▶ Profiling on [NVIDIA live-migrated virtual machines](#) is not supported and can result in undefined behavior.
- ▶ Profiling with enabled multi-process service (MPS) can result in undefined behavior.
- ▶ The NVLink Topology section is not supported for a configuration using NVSwitch.
- ▶ NVIDIA Nsight Compute does not support per-NVLink metrics.
- ▶ NVIDIA Nsight Compute does not support the *Logical NVLink Throughput* table.
- ▶ Setting a reduced NvLink Bandwidth mode does not impact the reported peak values for NvLink metrics. All peak values and corresponding percentages are calculated off the non-reduced NvLink bandwidth. Reconfiguring the NvLink Bandwidth mode using `nvidia-smi` while profiling may lead to undefined tools' behavior.
- ▶ Profiling CUDA graph kernel nodes that can launch device graphs or are part of device-launchable graphs is not supported. Use [Graph Profiling](#) mode instead.
- ▶ Profiling in [Graph Profiling](#) mode is performed on the context that is specified by the stream handle for the graph launch. Consequently, only memory allocated on this context is saved off and restored during graph replay and only kernel nodes executed on this context are profiled.
- ▶ On CUDA drivers older than 530.x, profiling on Windows Subsystem for Linux (WSL) is not supported if the system has multiple physical NVIDIA GPUs. This is not affected by setting `CUDA_VISIBLE_DEVICES`.
- ▶ On CUDA drivers older than 545.x, profiling on Windows 10 WSL is not supported.
- ▶ Profiling on Windows 10 WSL is not supported for systems that exceed 1 TB of system memory.
- ▶ Collecting software counters through PerfWorks currently forces all functions in the module of the profiled kernel to be loaded. This increases the host and device memory footprint of the target application for the remainder of the process lifetime.

## Compatibility

- ▶ Applications calling blocking functions on std input/output streams can result in the profiler to stop, until the blocking function call is resolved.
- ▶ NVIDIA Nsight Compute can hang on applications using RAPIDS in versions 0.6 and 0.7, due to an issue in cuDF.
- ▶ Profiling child processes launched via `clone()` is not supported.
- ▶ Profiling of Cooperative Groups kernels launched with `cuLaunchCooperativeKernelMultiDevice` is not yet supported.
- ▶ On Linux systems, when profiling `bsd-csh` scripts, the original application output will not be printed. As a workaround, use a different C-shell, e.g. `tcsh`.
- ▶ Attempting to use the `--clock-control` option to set the GPU clocks will fail when profiling on a GPU partition. Please use `nvidia-smi` (installed with NVIDIA display driver) to control the clocks for the entire GPU. This will require administrative privileges when the GPU is partitioned.

- ▶ On Linux aarch64, NVIDIA Nsight Compute does not work if the *HOME* environment variable is not set.
- ▶ NVIDIA Nsight Compute versions 2020.1.0 to 2020.2.1 are not compatible with CUDA driver version 460+ if the application launches Cooperative Groups kernels. Profiling will fail with error "UnknownError".
- ▶ Collecting CPU call stack information on Windows Server 2016 can hang NVIDIA Nsight Compute in some cases. Currently, the only workaround is to skip CPU call stack collection on such systems by not specifying the option **--call-stack**.
- ▶ When profiling a script, **--target-processes all** may target utility executables such as *xargs*, *uname* or *ls*. To avoid profiling these, use the **--target-processes-filter** option accordingly.
- ▶ On mobile platforms, **--kill** option is not supported with application replay mode.
- ▶ NVIDIA Nsight Compute might show invalid characters for Unicode names and paths on Windows 10. As a workaround, use a third-party terminal emulator, e.g. Git bash.

### User Interface

- ▶ The API Statistics filter in NVIDIA Nsight Compute does not support units.
- ▶ File size is the only property considered when resolving source files. Timestamps are currently ignored.
- ▶ Terminating or disconnecting an application in the *Interactive Profiling* activity while the API Stream View is updated can lead to a crash.
- ▶ See the [OptiX library support section](#) for limitations concerning the [Acceleration Structure Viewer](#).
- ▶ After updating from a previous version of NVIDIA Nsight Compute on Linux, the file load dialog may not allow column resizing and sorting. As a workaround, the `~/config/QtProject.conf` file can be edited to remove the *treeViewHeader* entry from the `[FileDialog]` section.

# Chapter 3.

## SUPPORT

Information on supported platforms and GPUs.

### 3.1. Platform Support

Host denotes the UI can run on that platform. Target means that we can instrument applications on that platform for data collection. Applications launched with instrumentation on a target system can be connected to from most host platforms. The reports collected on one system can be opened on any other system.

Table 1 Platforms supported by NVIDIA Nsight Compute

	Host	Targets
Windows	Yes	Windows*, Linux (x86_64)
Windows Subsystem for Linux (WSL2)	Yes	Windows Subsystem for Linux (WSL2) as part of the Linux (x86_64) package.
Linux (x86_64)	Yes	Windows*, Linux (x86_64), Linux (ppc64le), Linux (aarch64 sbsa)
Linux (ppc64le)	No	Linux (ppc64le)
Linux (aarch64 sbsa)	Yes	Linux (aarch64 sbsa)
Linux (x86_64) (Drive SDK)	Yes	Windows*, Linux (x86_64), Linux (aarch64), QNX
MacOSX 10.15+	Yes	Windows*, Linux (x86_64), Linux (ppc64le)
Linux (aarch64)	No	Linux (aarch64)
QNX	No	QNX

Target platforms marked with \* do not support remote launch from the respective host. Remote launch means that the application can be launched on the target system from the host UI. Instead, the application must be launched from the target system.

Profiling of 32-bit processes is not supported.

## 3.2. GPU Support

Table 2 GPU architectures supported by NVIDIA Nsight Compute

Architecture	Support
Kepler	No
Maxwell	No
Pascal	No
Volta GV100	Yes
Volta GV11b	Yes
Turing TU1xx	Yes
NVIDIA GA100	Yes
NVIDIA GA10x	Yes
NVIDIA GA10b	Yes
NVIDIA AD10x	Yes
NVIDIA GH100	Yes

Most metrics used in NVIDIA Nsight Compute are identical to those of the PerfWorks Metrics API and follow the documented [Metrics Structure](#). A comparison between the metrics used in nvprof and their equivalent in NVIDIA Nsight Compute can be found in the [NVIDIA Nsight Compute CLI User Manual](#).

## 3.3. Library Support

NVIDIA Nsight Compute can be used to profile CUDA applications, as well as applications that use CUDA via NVIDIA or third-party libraries. For most such libraries, the behavior is expected to be identical to applications using CUDA directly. However, for certain libraries, NVIDIA Nsight Compute has certain restrictions, alternate behavior, or requires non-default setup steps prior to profiling.

### 3.3.1. OptiX

NVIDIA Nsight Compute supports profiling of OptiX applications, but with certain restrictions.

#### ► Internal Kernels

Kernels launched by OptiX that contain no user-defined code are given the generic name *NVIDIA internal*. These kernels show up on the API Stream in the NVIDIA Nsight Compute UI, and can be profiled in both the UI as well as the NVIDIA Nsight Compute CLI. However, no CUDA-C source, PTX or SASS is available for them.

### ► User Kernels

Kernels launched by OptiX can contain user-defined code. OptiX identifies these kernels in the API Stream with a custom name. This name starts with *raygen\_\_* (for "ray generation"). These kernels show up on the API Stream and can be profiled in the UI as well as the NVIDIA Nsight Compute CLI. The Source page displays CUDA-C source, PTX and SASS defined by the user. Certain parts of the kernel, including device functions that contain OptiX-internal code, will not be available in the Source page.

### ► SASS

When SASS information is available in the profile report, certain instructions might not be available in the Source page and shown as *N/A*.

The [Acceleration Structure Viewer](#) for OptiX traversable handles currently has the following limitations:

- The [Acceleration Structure Viewer](#) is not supported on MacOSX.
- Viewing instance acceleration structures using multi-level instancing is not supported.
- Applying motion traversables to acceleration structures is not supported.

The following feature set is supported per OptiX API version:

OptiX API Version	Kernel Profiling	API Interception	Resource Tracking
6.x	Yes	No	No
7.0 - 7.7	Yes	Yes	Yes

## 3.4. System Requirements

### Linux and WSL

On all Linux platforms, NVIDIA Nsight Compute CLI requires GLIBC version 2.15 or higher. On x86\_64, the NVIDIA Nsight Compute UI requires GLIBC version 2.17 or higher. On Linux (aarch64 sbsa), the NVIDIA Nsight Compute UI requires GLIBC version 2.26 or higher.

The NVIDIA Nsight Compute UI requires several packages to be installed to enable Qt. Please refer to the [Qt for X11 Requirements](#). When executing `ncu-ui` with missing dependencies, an error message with information on the missing packages is shown. Note that only one package will be shown at a time, even though multiple may be missing from your system. For selected operating systems, the following commands install needed packages for NVIDIA Nsight Compute on X11:

### ► Ubuntu 18.04

```
apt install libopengl0 libxcb-icccm4 libxcb-image0 libxcb-
keysyms1 libxcb-render-util0 libxcb-xkb1 libxkbcommon-x11-0
```

► **Ubuntu 20.04**

```
apt install libopengl0 libxcb-icccm4 libxcb-image0 libxcb-  
keysyms1 libxcb-render-util0 libxcb-xinerama0 libxcb-xkb1  
libxkbcommon-x11-0
```

► **RHEL 8.6**

```
yum install libglvnd-opengl libxcb libxkbcommon-x11
```

Profiling on Windows Subsystem for Linux (WSL) is only supported with WSL version 2. Profiling is supported on Windows 10 WSL with OS build version 19044 and greater, and NVIDIA display driver version 545 or higher. It is not supported on Windows 10 WSL for systems that exceed 1 TB of system memory. Profiling is supported on Windows 11 WSL with NVIDIA display driver version 525 or higher. The Linux (x86\_64) NVIDIA Nsight Compute package can be used and should be installed directly within WSL2. Remote profiling to and from WSL2 works equivalently to regular Linux (x86\_64) hosts and targets, as long as it's accessible via SSH. Access to NVIDIA GPU Performance Counters must be enabled in the NVIDIA Control Panel of the Windows host. See also the [CUDA on WSL User Guide](#).

## Windows

Only Windows 10 and 11 are supported as host and target.

The Visual Studio 2017 redistributable is not automatically installed by the NVIDIA Nsight Compute installer. The workaround is to install the x64 version of the 'Microsoft Visual C++ Redistributable for Visual Studio 2017' manually. The installer is linked on the main download page for Visual Studio at <https://www.visualstudio.com/downloads/> or download directly from <https://go.microsoft.com/fwlink/?LinkId=746572>.



## **Notice**

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication of otherwise under any patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all other information previously supplied. NVIDIA Corporation products are not authorized as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

## **Trademarks**

NVIDIA and the NVIDIA logo are trademarks or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

## **Copyright**

© 2018-2023 NVIDIA Corporation and affiliates. All rights reserved.

This product includes software developed by the Syncro Soft SRL (<http://www.sync.ro/>).