NVIDIA CUDA Toolkit 12.1.72

Release Notes for CUDA 12.0

RN-06722-001 _v12.1 | February 2023

Table of Contents

List of Tables

Chapter 1. CUDA 12.0 Release Notes

The release notes for the NVIDIA® CUDA® Toolkit can be found online at https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html.

Note: The release notes have been reorganized into two major sections: the general CUDA release notes, and the CUDA libraries release notes.

1.1. CUDA Toolkit Major Component Versions

CUDA Components

Starting with CUDA 11, the various components in the toolkit are versioned independently.

For CUDA 12.1, the following table indicates the versions:

Table 1. CUDA 12.0 Component Versions

Component Name	Version Information	Supported Architectures
CUDA C++ Core Compute Libraries	12.0.90	x86_64, ppc64le, AArch64, Windows
CUDA Compatibility	12.0.31752801	AArch64
CUDA Runtime (cudart)	12.0.76	x86_64, ppc64le, AArch64, Windows
cuobjdump	12.0.76	x86_64, ppc64le, AArch64, Windows
CUPTI	12.0.90	x86_64, ppc64le, AArch64, Windows
CUDA cuxxfilt (demangler)	12.0.76	x86_64, ppc64le, AArch64, Windows
CUDA Demo Suite	12.0.76	x86_64
CUDA GDB	12.0.90	x86_64, ppc64le, AArch64
CUDA Nsight	12.0.78	x86_64, ppc64le

Component Name	Version Information	Supported Architectures
CUDA NVCC	12.0.76	x86_64, ppc64le, AArch64, Windows
CUDA nvdisasm	12.0.76	x86_64, ppc64le, AArch64, Windows
CUDA NVML Headers	12.0.76	x86_64, ppc64le, AArch64, Windows
CUDA nvprof	12.0.90	x86_64, ppc64le, Windows
CUDA nvprune	12.0.76	x86_64, ppc64le, AArch64, Windows
CUDA NVRTC	12.0.76	x86_64, ppc64le, AArch64, Windows
CUDA NVTX	12.0.76	x86_64, ppc64le, AArch64, Windows
CUDA NVVP	12.0.90	x86_64, ppc64le, Windows
CUDA openCL	12.0.76	x86_64, , Windows
CUDA Profiler API	12.0.76	x86_64, ppc64le, AArch64, Windows
CUDA Compute Sanitizer API	12.0.90	x86_64, ppc64le, AArch64, Windows
CUDA cuBLAS	12.0.0.153	x86_64, ppc64le, AArch64, Windows
CUDA cuDLA	12.0.76	AArch64
CUDA cuFFT	11.0.0.21	x86_64, ppc64le, AArch64, Windows
CUDA cuFile	1.5.0.59	x86_64
CUDA cuRAND	10.3.1.50	x86_64, ppc64le, AArch64, Windows
CUDA cuSOLVER	11.4.2.57	x86_64, ppc64le, AArch64, Windows
CUDA cuSPARSE	12.0.0.76	x86_64, ppc64le, AArch64, Windows
CUDA NPP	12.0.0.30	x86_64, ppc64le, AArch64, Windows
CUDA nvJitLink	12.0.76	x86_64, ppc64le, AArch64, Windows
CUDA nvJPEG	12.0.0.28	x86_64, ppc64le, AArch64, Windows
CUDA NVVM Samples	12.0.76	x86_64, ppc64le, AArch64, Windows
Nsight Compute	2022.4.0.15	x86_64, ppc64le, AArch64 (CLI only), Windows
Nsight Systems	2022.4.2.18	x86_64, ppc64le, AArch64 (CLI only), Windows

Component Name	Version Information	Supported Architectures
Nsight Visual Studio Edition (VSE)	2022.3.0.22319	Windows
nvidia_fs[1]	2.14.12	x86_64, AArch64
Visual Studio Integration	12.0.76	Windows
NVIDIA Linux Driver	525.60.03	x86_64, ppc64le, AArch64
NVIDIA Windows Driver	526.98	x86_64 (Windows)

CUDA Driver

Running a CUDA application requires the system with at least one CUDA capable GPU and a driver that is compatible with the CUDA Toolkit. See Table 3. For more information various GPU products that are CUDA capable, visit https://developer.nvidia.com/cuda-gpus.

Each release of the CUDA Toolkit requires a minimum version of the CUDA driver. The CUDA driver is backward compatible, meaning that applications compiled against a particular version of the CUDA will continue to work on subsequent (later) driver releases.

More information on compatibility can be found at https://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html#cuda-compatibility-and-upgrades.

Note: Starting with CUDA 11.0, the toolkit components are individually versioned, and the toolkit itself is versioned as shown in the table below.

The minimum required driver version for CUDA minor version compatibility is shown below. CUDA minor version compatibility is described in detail in https://docs.nvidia.com/deploy/cuda-compatibility/index.html

Table 2. CUDA Toolkit and Minimum Required Driver Version for CUDA Minor Version Compatibility

CUDA Toolkit	Minimum Required Driver Version for CUDA Minor Version Compatibility*		
	Linux x86_64 Driver Version	Linux AArch64 Driver Version	Windows x86_64 Driver Version
CUDA 12.0.x	>=450.80.02		>=452.39
CUDA 11.8.x			
CUDA 11.7.x			
CUDA 11.6.x			
CUDA 11.5.x			
CUDA 11.4.x			
CUDA 11.3.x			
CUDA 11.2.x			
CUDA 11.1 (11.1.0)			

[1] Only available on select Linux distros

CUDA Toolkit	Minimum Required Driver Version for CUDA Minor Version Compatibility*		
	Linux x86_64 Driver Version	Linux AArch64 Driver Version	Windows x86_64 Driver Version
CUDA 11.0 (11.0.3)	>=450.36.06**	>=450.28.01**	>=451.22**

* Using a Minimum Required Version that is **different** from Toolkit Driver Version could be allowed in compatibility mode -- please read the CUDA Compatibility Guide for details.

** CUDA 11.0 was released with an earlier driver version, but by upgrading to Tesla Recommended Drivers 450.80.02 (Linux) / 452.39 (Windows), minor version compatibility is possible across the CUDA 11.x family of toolkits.

The version of the development NVIDIA GPU Driver packaged in each CUDA Toolkit release is shown below.

Table 3. CUDA Toolkit and Corresponding Driver Versions

CUDA Toolkit	Toolkit Driver Version	
	Linux x86_64 Driver Version	Windows x86_64 Driver Version
CUDA 12.0 GA	>=525.60.03	>=526.98
CUDA 11.8 GA	>=520.61.05	>=522.06
CUDA 11.7 Update 1	>=515.48.07	>=516.31
CUDA 11.7 GA	>=515.43.04	>=516.01
CUDA 11.6 Update 2	>=510.47.03	>=511.65
CUDA 11.6 Update 1	>=510.47.03	>=511.65
CUDA 11.6 GA	>=510.39.01	>=511.23
CUDA 11.5 Update 2	>=495.29.05	>=496.13
CUDA 11.5 Update 1	>=495.29.05	>=496.13
CUDA 11.5 GA	>=495.29.05	>=496.04
CUDA 11.4 Update 4	>=470.82.01	>=472.50
CUDA 11.4 Update 3	>=470.82.01	>=472.50
CUDA 11.4 Update 2	>=470.57.02	>=471.41
CUDA 11.4 Update 1	>=470.57.02	>=471.41
CUDA 11.4.0 GA	>=470.42.01	>=471.11
CUDA 11.3.1 Update 1	>=465.19.01	>=465.89
CUDA 11.3.0 GA	>=465.19.01	>=465.89
CUDA 11.2.2 Update 2	>=460.32.03	>=461.33
CUDA 11.2.1 Update 1	>=460.32.03	>=461.09
CUDA 11.2.0 GA	>=460.27.03	>=460.82
CUDA 11.1.1 Update 1	>=455.32	>=456.81

CUDA Toolkit	Toolkit Driver Version	
	Linux x86_64 Driver Version	Windows x86_64 Driver Version
CUDA 11.1 GA	>=455.23	>=456.38
CUDA 11.0.3 Update 1	>= 450.51.06	>= 451.82
CUDA 11.0.2 GA	>= 450.51.05	>= 451.48
CUDA 11.0.1 RC	>= 450.36.06	>= 451.22
CUDA 10.2.89	>= 440.33	>= 441.22
CUDA 10.1 (10.1.105 general release, and updates)	>= 418.39	>= 418.96
CUDA 10.0.130	>= 410.48	>= 411.31
CUDA 9.2 (9.2.148 Update 1)	>= 396.37	>= 398.26
CUDA 9.2 (9.2.88)	>= 396.26	>= 397.44
CUDA 9.1 (9.1.85)	>= 390.46	>= 391.29
CUDA 9.0 (9.0.76)	>= 384.81	>= 385.54
CUDA 8.0 (8.0.61 GA2)	>= 375.26	>= 376.51
CUDA 8.0 (8.0.44)	>= 367.48	>= 369.30
CUDA 7.5 (7.5.16)	>= 352.31	>= 353.66
CUDA 7.0 (7.0.28)	>= 346.46	>= 347.62

For convenience, the NVIDIA driver is installed as part of the CUDA Toolkit installation. Note that this driver is for development purposes and is not recommended for use in production with Tesla GPUs.

For running CUDA applications in production with Tesla GPUs, it is recommended to download the latest driver for Tesla GPUs from the NVIDIA driver downloads site at https://www.nvidia.com/drivers.

During the installation of the CUDA Toolkit, the installation of the NVIDIA driver may be skipped on Windows (when using the interactive or silent installation) or on Linux (by using meta packages).

For more information on customizing the install process on Windows, see https://docs.nvidia.com/cuda/cuda-installation-guide-microsoft-windows/index.html#install-cuda-software.

For meta packages on Linux, see https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#package-manager-metas

1.2. CUDA Compilers

12.0

- Host Compiler : GCC 12
- JIT LTO

- C++20 support

1.3. CUDA Developer Tools

- For changes to nvprof and Visual Profiler, see the changelog.
- For new features, improvements, and bug fixes in CUPTI, see the changelog.
- For new features, improvements, and bug fixes in Nsight Compute, see the changelog.
- For new features, improvements, and bug fixes in Compute Sanitizer, see the changelog.
- For new features, improvements, and bug fixes in CUDA-GDB, see the changelog.

1.4. Resolved Issues

1.5. Deprecated Features

The following features are deprecated in the current release of the CUDA software. The features still work in the current release, but their documentation may have been removed, and they will become officially unsupported in a future release. We recommend that developers employ alternative solutions to these features in their software.

General CUDA

- CentOS Linux 8 has reached End-of-Life on December 31, 2021. Support for this OS is now removed from the CUDA Toolkit and is replaced by Rocky Linux 8.
- Server 2016 support has been deprecated and shall be removed in a future release.
- Kepler architecture support is removed from CUDA 12.0.
- JIT LTO support in CUDA Driver APIs which was previewed in CUDA 11.4 stand removed as the feature is productized as an independent library part of the CUDA Toolkit - nvJitLinker. The following enums supported by the cuLink Driver APIs are being removed:
  - `CU_JIT_INPUT_NVVM`
  - `CU_JIT_LTO`
  - `CU_JIT_FTZ`
  - `CU_JIT_PREC_DIV`
  - `CU_JIT_PREC_SQRT`
  - `CU_JIT_FMA`

- `CU_JIT_REFERENCED_KERNEL_NAMES`
- `CU_JIT_REFERENCED_KERNEL_COUNT`
- `CU_JIT_REFERENCED_VARIABLE_NAME`
- `CU_JIT_REFERENCED_VARIABLE_COUNT`
- `CU_JIT_OPTIMIZE_UNUSED_DEVICE_VARIABLES`

If your application used the above enums and was compiled with 11.x driver then it will not work with 12.0 driver. The 12.0 libcuda.so will still link but passing the above enums to the cuLinkAPIs will return `CUDA_ERROR_INVALID_VALUE` error at runtime. The same application source code that compiled with an 11.x toolkit will not compile with the 12.0 toolkit if it has references to the above enum. If your application is using cuSparse or JITIFY please refer to your library documentation for details of impact.

CUDA Tools

- CUDA-MEMCHECK is removed from CUDA 12.0.

CUDA Compiler

- NVCC and NVRTC now support the c++20 dialect. Most of the language features are available in host and device code; some, like coroutines, are not supported in device code. Modules are not supported for both host and device code.
- NVCC has removed support for 32-bit native and cross-compilation support on all platforms for ALL GPUs. Older CUDA toolkits will continue to support it. CUDA Driver will retain the 32-bit support functionality on all GPUs.
- NVRTC default C++ dialect changed from C++14 to C++17. Please refer to the ISO C++ standard for reference on the feature set and compatibility between the dialects.
- NVVM IR Update: with CUDA 12.0 we are releasing NVVM IR 2.0 which is incompatible with NVVM IR 1.x accepted by the libNVVM compiler in prior CUDA release toolkits. Users of the libNVVM compiler in CUDA 12.0 toolkit must generate NVVM IR 2.0.

1.6. Known Issues

1.6.1. CUDA Tools

- NVIDIA Visual Profiler can't remote into a target machine running Ubuntu 20.04.

Chapter 2. CUDA Libraries

This section covers CUDA Libraries release notes for 12.x releases.

- Support for the following compute capabilities is removed for all libraries:
  - sm_35 (Kepler)
  - sm_37 (Kepler)

2.1. cuBLAS Library

2.1.1. cuBLAS: Release 12.0

- **New Features**
  - `cublasLtMatmul` now supports FP8 with a non-zero beta.
  - Add int64 APIs to enable larger problem sizes; refer to 64-bit integer interface.
  - Add more Hopper specific kernels for `cublasLtMatmul` with epilogues:
    - `CUBLASLT_EPILOGUE_BGRAD{A,B}`
    - `CUBLASLT_EPILOGUE_{RELU,GELU}_AUX`
    - `CUBLASLT_EPILOGUE_D{RELU,GELU}`
  - Improve Hopper performance on arm64-sbsa by adding Hopper kernels that were previously supported only on the x86_64 architecture for Windows and Linux.
- **Known Issues**
  - There are no forward compatible kernels for single precision complex gemms that do not require workspace. Support will be added in a later release.
- **Resolved Issues**
  - Fixed an issue on NVIDIA Ampere architecture and newer GPUs where `cublasLtMatmul` with epilogue `CUBLASLT_EPILOGUE_BGRAD{A,B}` and a nontrivial reduction scheme (that is, not `CUBLASLT_REDUCTION_SCHEME_NONE`) could return incorrect results for the bias gradient.

- `cublasLtMatmul` for gemv-like cases (that is, m or n equals 1) might ignore bias with the `CUBLASLT_EPILOGUE_RELU_BIAS` and `CUBLASLT_EPILOGUE_BIAS` epilogues.

- **Deprecations**
  - Disallow including `cublas.h` and `cublas_v2.h` in the same translation unit.
  - Removed:
    - `CUBLAS_MATMUL_STAGES_16x80` and `CUBLAS_MATMUL_STAGES_64x80` from `cublasLtMatmulStages_t`. No kernels utilize these stages anymore.
    - `cublasLt3mMode_t`, `CUBLASLT_MATMUL_PREF_MATH_MODE_MASK`, and `CUBLASLT_MATMUL_PREF_GAUSSIAN_MODE_MASK` from `cublasLtMatmulPreferenceAttributes_t`. Instead, use the corresponding flags from `cublasLtNumericalImplFlags_t`.
    - `CUBLASLT_MATMUL_PREF_POINTER_MODE_MASK`, `CUBLASLT_MATMUL_PREF_EPILOGUE_MASK`, and `CUBLASLT_MATMUL_PREF_SM_COUNT_TARGET` from `cublasLtMatmulPreferenceAttributes_t`. The corresponding parameters are taken directly from `cublasLtMatmulDesc_t`.
    - `CUBLASLT_POINTER_MODE_MASK_NO_FILTERING` from `cublasLtPointerModeMask_t`. This mask was only applicable to `CUBLASLT_MATMUL_PREF_MATH_MODE_MASK` which was removed.

2.2. cuSPARSE Library

2.2.1. cuSPARSE: Release 12.0

- **New Features**
  - JIT LTO functionalities (`cusparseSpMMOp()`) switched from driver to nvJitLto library. Starting from CUDA 12.0 the user needs to link to `libnvJitLto.so`, see cuSPARSE documentation. JIT LTO performance has also been improved for `cusparseSpMMOpPlan()`.
  - Introduced const descriptors for the Generic APIs, e.g. `cusparseConstSpVecGet()`. Now, the Generic APIs interface clearly declares when a descriptor and its data are modified by the cuSPARSE functions.
  - Added two new algorithms to `cusparseSpGEMM()` with lower memory utilization. The first algorithm computes a strict bound on the number of intermediate product, while the second one allows partitioning the computation in chunks.
  - Added `int8_t` support to `cusparseGather()`, `cusparseScatter()`, and `cusparseCsr2cscEx2()`.
  - Improved `cusparseSpSV()` performance for both the analysis and the solving phases.
  - Improved `cusparseSpSM()` performance for both the analysis and the solving phases.

- Improved `cusparseSDDMM()` performance and added support for batch computation.
- Improved `cusparseCsr2cscEx2()` performance.

- **Resolved Issues**
  - `cusparseSpSV()`, `cusparseSpSM()` could produce wrong results.
  - `cusparseDnMatGetStridedBatch()` did not accept batchStride == 0.
- **Deprecations**
  - Removed deprecated CUDA 11.x APIs, enumerators, and descriptors.

2.3. Math Library

2.3.1. CUDA Math: Release 12.0

- **New Features**
  - Introduced new integer/fp16/bf16 CUDA Math APIs to help expose performance benefits of new DPX instructions. Refer to https://docs.nvidia.com/cuda/cuda-math-api/index.html.
- **Deprecations**
  - All previously deprecated undocumented APIs are removed from CUDA 12.0.

2.4. NVIDIA Performance Primitives (NPP)

2.4.1. NPP: Release 12.0

- **Deprecations**
  - Deprecating non-CTX API support from next release.
- **Resolved Issues**
  - A performance issue with NPP `ResizeSqrPixel` API is now fixed and shows improved performance.

2.5. nvJPEG Library

2.5.1. nvJPEG: Release 12.0

- **New Features**
  - Added the improvement in the GPU Memory optimisation for nvJPEG codec.
- **Resolved Issues**
  - An issue that causes runtime failures when nvJPEGDecMultipleInstances was tested with a large number of threads is resolved.
  - An issue with CMYK four component color conversion is now resolved.
- **Known Issues**
  - Backend `NVJPEG_BACKEND_GPU_HYBRID` - Unable to handle bistreams with extra scans lengths.
- **Deprecated Features**
  - The reuse of Huffman table in Encoder (`nvjpegEncoderParamsCopyHuffmanTables`) .

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

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

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
www.nvidia.com