



NVIDIA HPC SDK RELEASE NOTES

RN-09976-001-V23.5 | May 2023



TABLE OF CONTENTS

- Chapter 1. What's New..... 1
- Chapter 2. Release Component Versions..... 2
- Chapter 3. Supported Platforms..... 4
 - 3.1. Platform Requirements for the HPC SDK..... 4
 - 3.2. Supported CUDA Toolchain Versions..... 5
- Chapter 4. Known Limitations..... 6
- Chapter 5. Deprecations and Changes..... 7

LIST OF TABLES

Table 1 HPC SDK Release Components	2
Table 2 HPC SDK Platform Requirements	4

Chapter 1.

WHAT'S NEW

Welcome to the 23.5 version of the NVIDIA HPC SDK, a comprehensive suite of compilers and libraries enabling developers to program the entire HPC platform, from the GPU foundation to the CPU and out through the interconnect. The 23.5 release of the HPC SDK includes new features as well as important functionality and performance improvements.

- ▶ Environment variables for controlling how the OpenACC runtime controls memory allocations are available. Please refer to the [Using OpenACC](#) section of the the HPC Compilers Users Guide for more details.
- ▶ HPC-X and OpenMPI 4 have been updated to work with CUDA 12 and CUDA 11. HPX-X 2.15 is included to work with CUDA 12 and HPC-X 3.14 works with CUDA 11. New modulefiles, "nvhpc-hpcx-cuda12" and "nvhpc-hpcx-cuda11" are included. Updated OpenMPI 4.1.5 libraries are included with CUDA 12 and CUDA 11.
- ▶ The HPC Compilers now provide the `-gpu=sm_XY` option to include SASS following the behavior of `nvcc's --gpu-architecture=sm_80`.
- ▶ Using the `-MD` and `-MMD` options will now cause the HPC Compilers to output a `.d` file if the `-o` is specified.
- ▶ The CUDA compatibility files for CUDA 12 are now being shipped with the HPC SDK. Please refer to the [CUDA Documentation](#) for usage instructions.

Chapter 2.

RELEASE COMPONENT VERSIONS

The NVIDIA HPC SDK 23.5 release contains the following versions of each component:

Table 1 HPC SDK Release Components

	Linux_x86_64			Linux_ppc64le			Linux_aarch64		
	CUDA 11.0	CUDA 11.8	CUDA 12.1	CUDA 11.0	CUDA 11.8	CUDA 12.1	CUDA 11.0	CUDA 11.8	CUDA 12.1
nvc++	23.5			23.5			23.5		
nvc	23.5			23.5			23.5		
nvfortran	23.5			23.5			23.5		
nvcc	11.0.221	11.8.89	12.1.105	11.8.89	11.0.221	12.1.105	11.0.221	11.8.89	12.1.105
NCCL	2.18.1	2.18.1	2.18.1	2.18.1	2.18.1	2.18.1	2.18.1	2.18.1	2.18.1
NVSHMEM	2.9.0	2.9.0	2.9.0	2.9.0	2.9.0	2.9.0	N/A	N/A	N/A
cuBLAS	11.2.0.252	11.11.4.17	12.1.3.1	11.2.0.252	11.11.3.6	12.1.3.1	11.2.0.252	11.11.3.6	12.1.3.1
cuFFT	10.2.1.245	10.9.0.58	11.0.2.54	10.2.1.245	10.9.0.58	11.0.2.54	10.2.1.245	10.9.0.58	11.0.2.54
cuFFTMp	N/A	11.0.5	11.0.5	N/A	11.0.5	N/A	N/A	N/A	N/A
cuRAND	10.2.1.245	10.3.0.86	10.3.2.106	10.2.1.245	10.3.0.86	10.3.2.106	10.2.1.245	10.3.0.86	10.3.2.106
cuSOLVER	10.6.0.245	11.4.1.48	11.4.5.107	10.6.0.245	11.4.1.48	11.4.5.107	10.6.0.245	11.4.1.48	11.4.5.107
cuSOLVER	N/A	0.4.0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
cuSPARSE	11.1.1.245	11.7.5.86	12.1.0.106	11.1.1.245	11.7.5.86	12.1.0.106	11.1.1.245	11.7.5.86	12.1.0.106
cuTENSORFLOW	1.7.0	1.7.0	1.7.0	1.7.0	1.7.0	1.7.0	1.7.0	1.7.0	1.7.0
Nsight Compute	2023.1.1			2023.1.1			2023.1.1		
Nsight Systems	2023.2.1.122			2023.2.1.122			2023.2.1.122		
OpenMPI	3.1.5			3.1.5			3.1.5		
HPC-X	N/A	2.14	2.15	N/A	N/A	N/A	N/A	2.14	2.15

	Linux_x86_64			Linux_ppc64le			Linux_aarch64		
	CUDA 11.0	CUDA 11.8	CUDA 12.1	CUDA 11.0	CUDA 11.8	CUDA 12.1	CUDA 11.0	CUDA 11.8	CUDA 12.1
OpenBLAS	0.3.20			0.3.20			0.3.20		
Scalapack	2.2.0			2.2.0			2.2.0		
Thrust	1.9.9	1.15.1	2.0.1	1.9.9	1.15.1	2.0.1	1.9.9	1.15.1	2.0.1
CUB	1.9.9	1.15.1	2.0.1	1.9.9	1.15.1	2.0.1	1.9.9	1.15.1	2.0.1
libcud++	1.0.0	1.8.1	1.9.0	1.0.0	1.8.1	1.9.0	1.0.0	1.8.1	1.9.0

Chapter 3. SUPPORTED PLATFORMS

3.1. Platform Requirements for the HPC SDK

Table 2 HPC SDK Platform Requirements

Architecture	Linux Distributions	Minimum gcc/ glibc Toolchain	Minimum CUDA Driver
x86_64	RHEL/CentOS 7.3 - 7.9 RHEL/CentOS/Rocky 8.0 - 8.7 Fedora 33, 34 OpenSUSE Leap 15.2 - 15.4 SLES 15SP2, 15SP3, 15SP4 Ubuntu 18.04, 20.04, 22.04 Debian 10	C99: 4.8 C11: 4.9 C++03: 4.8 C++11: 4.9 C++14: 5.1 C++17: 7.1 C++20: 10.1	450.36.06
ppc64le	RHEL 7.3 - 7.7 RHEL 8.0 - 8.7	C99: 4.8 C11: 4.9 C++03: 4.8 C++11: 4.9 C++14: 5.1 C++17: 7.1 C++20: 10.1	450.36.06
aarch64	RHEL/CentOS/Rocky 8.0 - 8.7 Ubuntu 20.04, 22.04	C99: 4.8 C11: 4.9 C++03: 4.8 C++11: 4.9	450.36.06

Architecture	Linux Distributions	Minimum gcc/ glibc Toolchain	Minimum CUDA Driver
	SLES 15SP2, 15SP3, 15SP4 Amazon Linux 2	C++14: 5.1 C++17: 7.1 C++20: 10.1	

Programs generated by the HPC Compilers for x86_64 processors require a minimum of AVX instructions, which includes Sandy Bridge and newer CPUs from Intel, as well as Bulldozer and newer CPUs from AMD. POWER 8 and POWER 9 CPUs from the POWER architecture are supported. The HPC SDK includes support for v8.1+ Server Class Arm CPUs that meet the requirements appendix E specified in the SBSA 7.1 specification.

The HPC Compilers are compatible with gcc and g++ and use the GCC C and C++ libraries; the minimum compatible versions of GCC are listed in Table 2. The minimum system requirements for CUDA and NVIDIA Math Library requirements are available in the [NVIDIA CUDA Toolkit documentation](#).

3.2. Supported CUDA Toolchain Versions

The NVIDIA HPC SDK uses elements of the CUDA toolchain when building programs for execution with NVIDIA GPUs. Every HPC SDK installation package puts the required CUDA components into an installation directory called `[install-prefix]/[arch]/[nvhpc-version]/cuda`.

An NVIDIA CUDA GPU device driver must be installed on a system with a GPU before you can run a program compiled for the GPU on that system. The NVIDIA HPC SDK does not contain CUDA Drivers. You must download and install the appropriate [CUDA Driver from NVIDIA](#), including the [CUDA Compatibility Platform](#) if that is required.

The `nvaccelinfo` tool prints the CUDA Driver version in its output. You can use it to find out which version of the CUDA Driver is installed on your system.

The NVIDIA HPC SDK 23.5 includes the following CUDA toolchain versions:

- ▶ CUDA 11.0
- ▶ CUDA 11.8
- ▶ CUDA 12.1

The minimum required CUDA driver versions are listed in the table in Section 3.1.

Chapter 4.

KNOWN LIMITATIONS

- ▶ The `-Mipa` option has been disabled starting with the 23.3 version of the HPC Compilers.
- ▶ The latest version of `cuSolverMp` bundled with this release has two new dependencies on UCC and UCX libraries. To execute a program linked against `cuSolverMP`, please use the “`nvhpc-hpcx-cuda12`” environment module for the HPC-X library, or set the environment variable `LD_LIBRARY_PATH` as follows:
`LD_LIBRARY_PATH=${NVHPCSDK_HOME}/comm_libs/hpcx/latest/ucc/lib:
${NVHPCSDK_HOME}/comm_libs/11.8/hpcx/latest/ucx/lib:$LD_LIBRARY_PATH`
- ▶ If not using the provided modulefiles, prior to using HPC-X, users should take care to source the `hpcx-init.sh` script: `./[install-path]/Linux_x86_64/dev/comm_libs/hpcx/hpcx-2.11/hpcx-init.sh` Then, run the `hpcx_load` function defined by this script: `$ hpcx_load` These actions will set important environment variables that are needed when running HPC-X. The following warning from HPC-X while running an MPI job – “WARNING: Open MPI tried to bind a process but failed. This is a warning only; your job will continue, though performance may be degraded” – is a known issue, and may be worked around as follows: `export OMPI_MCA_hwloc_base_binding_policy=""`
- ▶ Fortran derived type objects with zero-size derived type allocatable components that are used in sourced allocation or allocatable assignment may result in a runtime segmentation violation.
- ▶ When using `-stdpar` to accelerate C++ parallel algorithms, the algorithm calls cannot include virtual function calls or function calls through a function pointer, cannot use C++ exceptions, can only dereference pointers that point to the heap, and must use random access iterators (raw pointers as iterators work best).

Chapter 5.

DEPRECATIONS AND CHANGES

- ▶ Beginning with the HPC SDK 23.7, the deprecated `CUDA_HOME` environment variable will not affect the HPC Compilers.
- ▶ The `-ta=tesla`, `-Mcuda`, `-Mcudalib` options for the HPC Compilers have been deprecated.
- ▶ Support for the RHEL 7-based operating systems will be removed in the HPC SDK version 23.7, corresponding with the upstream end-of-life (EOL).
- ▶ In an upcoming release the HPC SDK will bundle only CUDA 11.8 and the latest version of the CUDA 12.x series. Codepaths in the HPC Compilers that support CUDA versions older than 11.0 will no longer be tested or maintained.
- ▶ Support for the Ubuntu 18.04 operating system will be removed in the HPC SDK version 23.5, corresponding with the upstream end-of-life (EOL).
- ▶ Support for CUDA Fortran textures is deprecated in CUDA 11.0 and 11.8, and has been removed from CUDA 12.
- ▶ `cudaDeviceSynchronize()` in CUDA Fortran has been deprecated, and support has been removed from device code. It is still supported in host code.
- ▶ Starting with the 21.11 version of the NVIDIA HPC SDK, the HPC-X package is no longer shipped as part of the packages made available for the POWER architecture.
- ▶ Starting with the 21.5 version of the NVIDIA HPC SDK, the `-cuda` option for `NVC++` and `NVFORTTRAN` no longer automatically links the NVIDIA GPU math libraries. Please refer to the `-cudalib` option.
- ▶ HPC Compiler support for the Kepler architecture of NVIDIA GPUs was deprecated starting with the 21.3 version of the NVIDIA HPC SDK.
- ▶ Support for the KNL architecture of multicore CPUs in the NVIDIA HPC SDK was removed in the HPC SDK version 21.3.

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, the NVIDIA logo, CUDA, CUDA-X, GPUDirect, HPC SDK, NGC, NVIDIA Volta, NVIDIA DGX, NVIDIA Nsight, NVLink, NVSwitch, and Tesla are trademarks and/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

© 2013-2023 NVIDIA Corporation. All rights reserved.