



NVIDIA CUDA TEGRA TOOLKIT 10.2.89

RN-06722-001 _v10.2 | October 2019

Release Notes for Development Auto 5.1.9

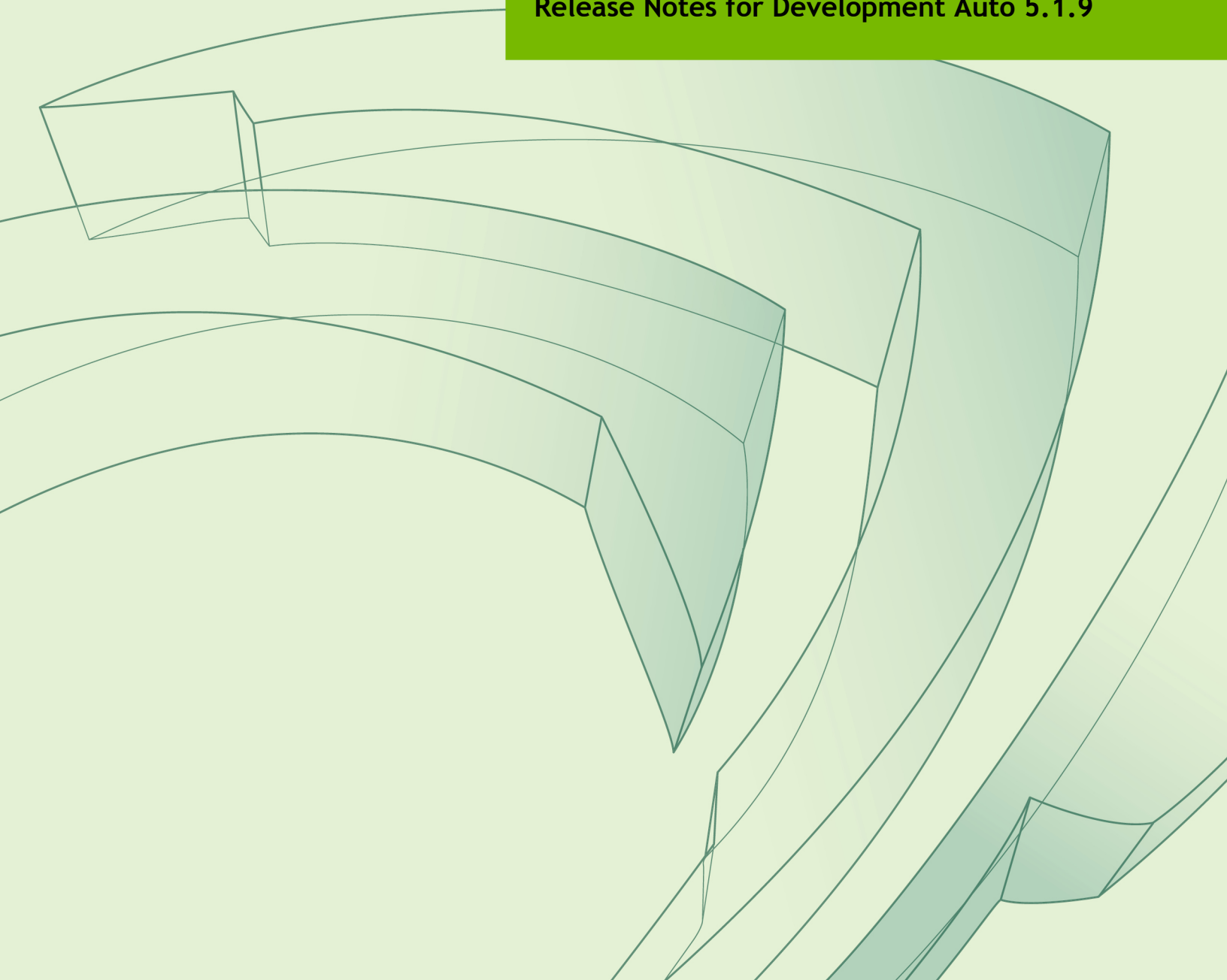


TABLE OF CONTENTS

Chapter 1. CUDA Tegra Release Notes for Development..... 1

 1.1. New Features..... 1

 1.2. Known Issues and Limitations..... 2

 1.3. Resolved Issues..... 2

 1.4. Support Matrix..... 3

LIST OF TABLES

Table 1 OS Support Matrix for CUDA Tegra for Development	3
--	---

Chapter 1.

CUDA TEGRA RELEASE NOTES FOR DEVELOPMENT

These are the release notes for the early access (EA) version of the CUDA Tegra Toolkit for Development. The release notes for the desktop version of CUDA also apply to CUDA Tegra. On Tegra, the CUDA Toolkit version is 10.2. The latest release notes for the desktop CUDA Toolkit are [posted here](#).

1.1. New Features

General

- ▶ CUDA now supports interoperability with cross-system EGLStreams between Tegra and x86 (pitch linear).

CUDA Installer

- ▶ Support is added for simultaneous installation of multiple versions of CUDA on the same host machine.
- ▶ CUDA Installer now provides a Debian file for QNX target systems with **qnx-aarch64** binaries for both safe & non-safe versions.

CUDA Compiler

- ▶ Optimizations made to improve the performance of **CUTLASS**.

CUDA Tegra Driver

- ▶ Support added for Ubuntu 18.04 on host for the AGX Drive platform.
- ▶ CUDA compatible with QNX SDP 7.0.4.
- ▶ Support added for NvSciSync-based inter-op (supported only on Tegra).
- ▶ Support added for NvSciBuffer-based inter-op (supported only on Tegra).

CUDA Libraries

- ▶ nvJPEG Library is now supported on QNX.

CUDA Developer Tools

- ▶ `nvcc` now supports the q++ QNX compiler as host compiler. A new `nvcc` flag `--qpp-config` has been added to specify the host compiler configuration (`[[compiler/]version,] [target])` when using q++. The arguments to this flag will be forwarded to q++ with its `-v` flag.
- ▶ CUPTI extends Profiling API data collection to Linux **aarch64** and QNX **aarch64** platforms.
- ▶ **Nsight Compute** - The following feature is now added to QNX: The ability to profile an application that spawns child processes, and have the profiler collect the data for those child processes. This feature was available previously on other platforms.

1.2. Known Issues and Limitations

- ▶ `7_CUDALibraries/nvJPEG` and `7_CUDALibraries/nvJPEG_encoder` samples fail to build for QNX as `TARGET_OS` due to incompatible QNX toolchain code used in it. This will be fixed in a future release.
- ▶ In the Early Access (EA) version of the CUDA Toolkit 10.2 for NVIDIA DRIVE OS 5.1.9, the compiler will report a version number with "10.1" in the version output string, like this: "release 10.1, V<10.1.xxx>". The `nvcc` predefined macro `__CUDACC_VER_MINOR__` will report "1" instead of "2" as expected for 10.2 toolkit.
- ▶ Objects generated with CUDA 10.2 compiler in NVIDIA DRIVE OS 5.1.6 Toolkit cannot be linked with the linker from the NVIDIA DRIVE OS 5.1.9 Toolkit. This is the expected behavior.
- ▶ CUPTI and Nsight Compute kernel profiling on TU104 dGPU may not work reliably with a background dGPU workload.
- ▶ The CUPTI documentation and samples are bundled only with the host (x86_64) CUDA Toolkit packages, and are located under ".../extras/CUPTI" directory. These samples and documentation are not available in the native arm64 packages.

1.3. Resolved Issues

General CUDA

- ▶ In earlier releases, the `cudaDeviceGetAttribute` methods returned **false** for the attribute `cudaDevAttrHostNativeAtomicSupported` for T194, because system-wide atomics were not supported (for NVIDIA Drive and Jetson platforms). This is fixed in CUDA 10.2, where system-wide atomics are now supported.
- ▶ In earlier Auto 5.1.x versions, the max clock rate obtained through `cudaGetDeviceProperties()` may be inaccurate for TU104. This is fixed in CUDA 10.2 for Auto 5.1.9.

1.4. Support Matrix

The table below shows the supported OS versions for CUDA Tegra for Development.

Table 1 OS Support Matrix for CUDA Tegra for Development

Host OS	Host OS Version	Target OS	Target OS Version	Compiler Support
Ubuntu	18.04 LTS	Ubuntu	Ubuntu 18.04	GCC 7.3
		QNX	QNX (7.0.4 SDP)	GCC 5.4
		Yocto	Yocto 2.5	GCC 7.3

Acknowledgments

NVIDIA extends thanks to Professor Mike Giles of Oxford University for providing the initial code for the optimized version of the device implementation of the double-precision `exp()` function found in this release of the CUDA toolkit.

NVIDIA acknowledges Scott Gray for his work on small-tile GEMM kernels for Pascal. These kernels were originally developed for OpenAI and included since cuBLAS 8.0.61.2.

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

© 2007-2019 NVIDIA Corporation. All rights reserved.
www.nvidia.com

