NVIDIA DriveOS 7.x SDK Migration Guide

Release 7.0.3

NVIDIA Corporation

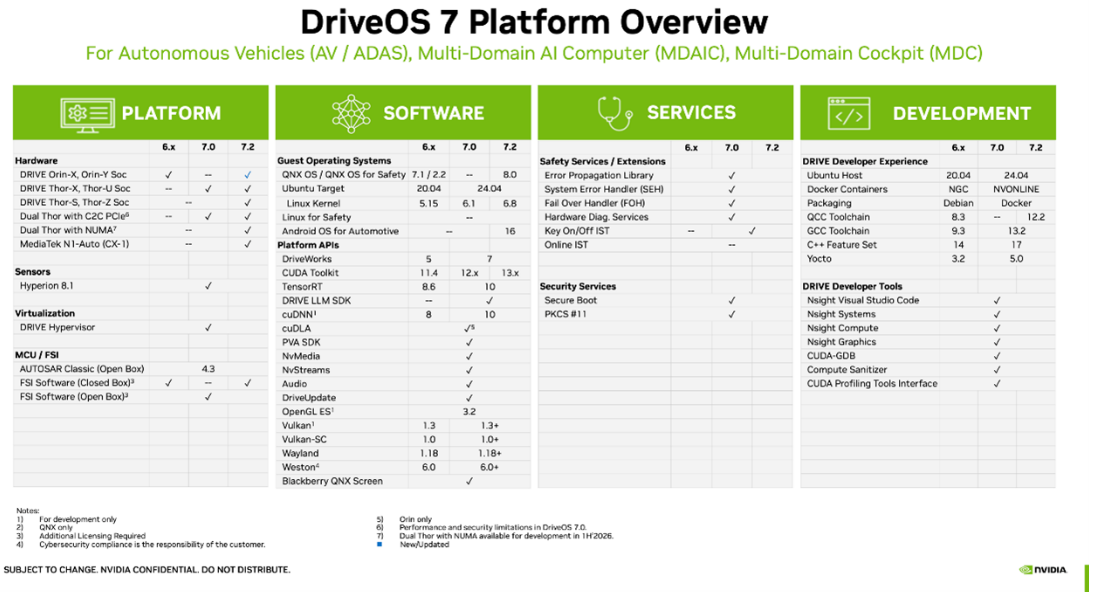
Jul 21, 2025

Contents

Chapter 1. Overview

This document summarizes expected changes moving from NVIDIA DRIVE® OS 6.0 to NVIDIA DriveOS™ 7.x. While NVIDIA DRIVE Orin™ supports DRIVE OS 6.0, the DriveOS 7.0 is the next major version update, supporting next generation SoC NVIDIA DRIVE Thor, which delivers new features and version updates across all the components of DriveOS.

The following figure identifies component changes between 6.0 and 7.0.



DriveOS 7 Platform Overview

For Autonomous Vehicles (AV / ADAS), Multi-Domain AI Computer (MDAIC), Multi-Domain Cockpit (MDC)

PLATFORM

	6.x	7.0	7.2
Hardware			
DRIVE Orin-X, Orin-Y Soc	✓	--	✓
DRIVE Thor-X, Thor-U Soc	--	✓	✓
DRIVE Thor-S, Thor-Z Soc	--		✓
Dual Thor with C2C PCIe[6]	--	✓	✓
Dual Thor with NUMA[7]	--		✓
MediaTek N1-Auto (CX-1)	--		✓
Sensors			
Hyperion 8.1	✓		
Virtualization			
DRIVE Hypervisor	✓		
MCU / FSI			
AUTOSAR Classic (Open Box)	4.3		
FSI Software (Closed Box)[3]	✓	--	✓
FSI Software (Open Box)[3]	✓		

SOFTWARE

	6.x	7.0	7.2
Guest Operating Systems			
QNX OS / QNX OS for Safety	7.1 / 2.2	--	8.0
Ubuntu Target	20.04	24.04	
Linux Kernel	5.15	6.1	6.8
Linux for Safety	--		
Android OS for Automotive	--		16
Platform APIs			
DriveWorks	5	7	
CUDA Toolkit	11.4	12.x	13.x
TensorRT	8.6	10	
DRIVE LLM SDK	--	✓	
cuDNN[1]	8	10	
cuDLA	✓[5]		
PVA SDK	✓		
NvMedia	✓		
NvStreams	✓		
Audio	✓		
DriveUpdate	✓		
OpenGL ES[1]	3.2		
Vulkan[1]	1.3	1.3+	
Vulkan-SC	1.0	1.0+	
Wayland	1.18	1.18+	
Weston[4]	6.0	6.0+	
Blackberry QNX Screen	✓		

SERVICES

	6.x	7.0	7.2
Safety Services / Extensions			
Error Propagation Library	✓		
System Error Handler (SEH)	✓		
Fail Over Handler (FOH)	✓		
Hardware Diag. Services	✓		
Key On/Off IST	--	✓	
Online IST	--		
Security Services			
Secure Boot	✓		
PKCS #11	✓		

DEVELOPMENT

	6.x	7.0	7.2
DRIVE Developer Experience			
Ubuntu Host	20.04	24.04	
Docker Containers	NGC	NVONLINE	
Packaging	Debian	Docker	
QCC Toolchain	8.3	--	12.2
GCC Toolchain	9.3	13.2	
C++ Feature Set	14	17	
Yocto	3.2	5.0	
DRIVE Developer Tools			
Nsight Visual Studio Code	✓		
Nsight Systems	✓		
Nsight Compute	✓		
Nsight Graphics	✓		
CUDA-GDB	✓		
Compute Sanitizer	✓		
CUDA Profiling Tools Interface	✓		

Notes:
1) For development only
2) QNX only
3) Additional Licensing Required
4) Cybersecurity compliance is the responsibility of the customer.
5) Orin only
6) Performance and security limitations in DriveOS 7.0.
7) Dual Thor with NUMA available for development in 1H'2026.
■ New/Updated

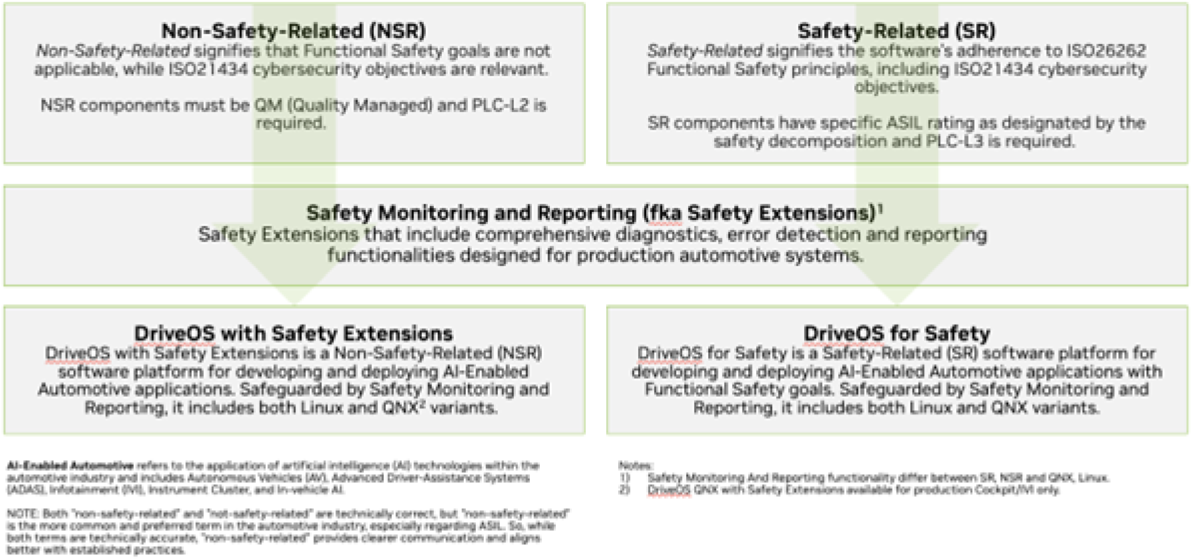
SUBJECT TO CHANGE. NVIDIA CONFIDENTIAL. DO NOT DISTRIBUTE.

The following table describes variables used in this document:

Field	Mandatory	DRIVE OS Values
DrivePlatform	Yes	driveos
OS	Yes	common
DriveType	Yes	dev_nsr prod_nsr test_nsr prod_sr test_sr debug_sr
Module	Yes, only for tar-based artifacts	Component name
SDK		sdk
DriveVersion	Yes	{RELEASE}-{GCID} For example, 7.0.x.0-GCID, 7.x.0.0-GCID

Chapter 2. Product Changes

2.1. New Terminology for Non-Safety-Related and Safety-Related



Non-Safety-Related (NSR)

Non-Safety-Related signifies that Functional Safety goals are not applicable, while ISO21434 cybersecurity objectives are relevant.

NSR components must be QM (Quality Managed) and PLC-L2 is required.

Safety-Related (SR)

Safety-Related signifies the software's adherence to ISO26262 Functional Safety principles, including ISO21434 cybersecurity objectives.

SR components have specific ASIL rating as designated by the safety decomposition and PLC-L3 is required.

Safety Monitoring and Reporting (fka Safety Extensions)[1]

Safety Extensions that include comprehensive diagnostics, error detection and reporting functionalities designed for production automotive systems.

DriveOS with Safety Extensions

DriveOS with Safety Extensions is a Non-Safety-Related (NSR) software platform for developing and deploying AI-Enabled Automotive applications. Safeguarded by Safety Monitoring and Reporting, it includes both Linux and QNX[2] variants.

DriveOS for Safety

DriveOS for Safety is a Safety-Related (SR) software platform for developing and deploying AI-Enabled Automotive applications with Functional Safety goals. Safeguarded by Safety Monitoring and Reporting, it includes both Linux and QNX variants.

AI-Enabled Automotive refers to the application of artificial intelligence (AI) technologies within the automotive industry and includes Autonomous Vehicles (AV), Advanced Driver-Assistance Systems (ADAS), Infotainment (IVI), Instrument Cluster, and In-vehicle AI.

NOTE: Both "non-safety-related" and "not-safety-related" are technically correct, but "non-safety-related" is the more common and preferred term in the automotive industry, especially regarding ASIL. So, while both terms are technically accurate, "non-safety-related" provides clearer communication and aligns better with established practices.

Notes:

1) Safety Monitoring And Reporting functionality differ between SR, NSR and QNX, Linux.
2) DriveOS QNX with Safety Extensions available for production Cockpit/IVI only.

The following table highlights product overview changes for DriveOS 6.x to 7.2 migration:

Use Case	DRIVE OS 6.x	DriveOS 7.2
Linux for Production (Non-Safety-Related) (NSR)	DRIVE OS Linux with Safety Extensions (aka DRIVE OS Linux "Production")	DriveOS Linux with Safety Extensions
Linux for Development	DRIVE OS Linux "Standard"	DriveOS Linux with Safety Extensions + Overlay (NSR)

2.2. SDK and PDK Merge in 7.0

In DRIVE OS 6.x, an SDK facilitated application development and the PDK was for platform customization. In DriveOS 7:

- SDK and PDK builds merge into a single package, which is available to all customers and named NVIDIA DriveOS 7 SDK.
- SDK and PDK developer guides merge into a single package, which is available to all customers and named *NVIDIA DriveOS 7 SDK Developer Guide*.
- DriveOS is migrating to a single-package comprehensive EULA.

2.3. System on a Chip (SoC) in DriveOS 7.0

The following table highlights the SoC changes for DriveOS 6.x to 7.2 migration:

SoC	6.x	7.0	7.2
Orin	Yes	No	No
Thor		Yes	Yes

Note

"Yes" indicates it is/was for production.

Chapter 3. Platform Changes

3.1. Boards

The following table lists the boards, along with SKUs, which remain supported.

Board	SKU	SoC	Release
P3710	SKU 10/12	Orin	6.x to 7.2
P3663	SKU A03	Orin	6.x to 7.2
P3898	SKU 0	Orin	6.0 Only
P3960	SKU 10/12	Thor	7.x Only

3.2. Platform Topologies and Node Configurations

The following table lists node configurations for platform configuration, which remain supported along with new Platform Config support.

Platform Config ID (PCID)	Configuration Description	DriveOS Linux	DriveOS QNX	x86
ASO	(Automotive) Single Orin (P3710) or (P3663)	6.x to 7.0	6.x to 7.0	
ADO	(Automotive) Dual Orin (P3710) over PCIe	6.x to 7.0	6.x to 7.0	
AQO	(Automotive) Quad Orin (P3710) over PCIe, nearest neighbor	6.x to 7.0	n/s	
ASO-SXSA	(Automotive) Single Orin (P3701) as EP to Single x86 with Single Ampere (PG199)	6.x to 7.0		6.x to 7.0
AST	(Automotive) Single Thor (P3960)	7.0	7.0	
ADT	(Automotive) Dual Thor (P3960) over PCIe	7.0	7.0	

Note

DriveOS 7 has no support for NvSciC2C to connect Thor to x86

3.3. Partition Configuration Table (PCT) Changes

PCT	Platform flavors	Release
AV + L	Linux standard▯Linux NSR + Development Build Linux with safety extensions▯Linux NSR	6.x to 7.0
AV + Q	QNX standard to QNX NSR + Development Build QNX Safety to QNX SR	6.x to 7.2

Table for PCT bind option changes used to build images:

AV PCT DRIVE OS 6.0	AV PCT DriveOS 7.x	Comments
Only SoC-based board support (AV+L, AV+Q)	Orin + Thor SoC-based board support (AV+L, AV+Q)	Orin carryforward, Thor is a new addition
PCTs bind options supported: **QNX** ▶ standard ▶ prod ▶ prod_debug ▶ prod_debug_extra **Linux** ▶ standard ▶ prod ▶ prod_debug	**PCTs bind options supported:** **QNX** ▶ dev_nsr ▶ prod_sr ▶ test_sr ▶ debug_sr **Linux** ▶ dev_nsr ▶ prod_nsr ▶ test_nsr	

Chapter 4. Packaging and Distribution Changes

DriveOS 7.0 includes packaging and distribution changes from 6.0.

4.1. Artifacts Distribution and Delivery Changes

- Docker-only Distribution.
- DriveOS 7.0 uses tar archive packages, replacing some Debian packages.
- Docker container distribution occurs via the Artifactory repository with NVIDIA NVONLINE authentication enabled. Users are not required to sign up for NGC. Refer to the *NVIDIA DriveOS 7.0 SDK Developer Guide* for Docker container download instructions.

4.2. Developer Tools

Note

The NVIDIA Developer Tools package is preinstalled in Docker and deb packages are available under /drive/extra folder.

4.2.1. Installing Nsight Systems

```
sudo dpkg -i /drive/extra/public/NsightSystems-linux-drive-7-nda-2025.1.2.19-3545308.
↪deb
```

4.2.2. Installing Nsight Graphics Systems

```
sudo dpkg -i /drive/extra/public/NVIDIA_Nsight_Graphics_DRIVE_NDA_2024.3.25036.deb
```

4.3. CUDA

Note

The NVIDIA CUDA package is preinstalled inside Docker and files are available under /drive/extra folder in docker. CUDA version will be updated from 11.4 to 12.8

Install the CUDA 12.8 Debian packages for Linux under the same folder:

```
sudo dpkg -i \
/drive/extra/public/driveos-cuda-repo-ubuntu2404-12-8-local_12.8.10-570.86.10-1_amd64.
↪deb \
/drive/extra/public/driveos-cuda-thor-nsr-repo-cross-aarch64-ubuntu2404-12-8-local_12.
↪8.10-1_all.deb

sudo cp /var/*cuda-*/*-keyring.gpg /usr/share/keyrings/

sudo apt update

sudo apt -y install cuda-toolkit-12-8
```

For issues with the installation, remove old packages and reinstall:

```
sudo rm /var/lib/apt/lists/_var_driveos-cuda*

sudo apt --fix-broken install -y

sudo apt autoremove -y

sudo apt remove --purge -y "cuda*" "driveos-cuda*"

sudo apt remove --purge -y "*cublas*"
```

4.4. cuDNN

Note

The NVIDIA cuDNN package is preinstalled inside Docker, and files are available under the */drive/extra* folder in Docker. The cuDNN versions will be updated from 8.x to 10.x.

```
sudo dpkg -i \
/drive/extra/public/cudnn-local-repo-cross-aarch64-ubuntu2404-9.99.3_1.0-1_all.deb \
/drive/extra/public/cudnn-local-repo-ubuntu2404-9.99.3_1.0-1_amd64.deb

sudo cp /var/cudnn*/cudnn*keyring.gpg /usr/share/keyrings/

sudo apt install -y cudnn libcudnn9-samples cudnn-cross-aarch64
```

4.5. TensorRT

Note

The NVIDIA TensorRT package are preinstalled inside Docker and deb files are available under /drive/extra folder in docker. The TensorRT versions will be updated from 8.x to 10.x.

```
sudo dpkg -i /drive/extra/public/nv-tensorrt-repo-ubuntu2404-cuda12.8-trt10.8.10.5-
↪d7l-cross-ga-20250201_1-1_amd64.deb

sudo apt-key add /var/nv-tensorrt*/*.pub

sudo apt update

sudo apt-get install tensorrt-cross-aarch64 libnvinfer-dev-cross-aarch64 -y
```

Note

TensorRT removed the dependency on cuDNN in DriveOS 7.0

4.6. Linux (NSR) Docker

4.6.1. Docker Container Names (Linux NSR)

Docker Container File Names
edge.urm.nvidia.com/sw-driveos-linux-docker-local/release/drive-agx-linux-nsr-aarch64-sdk-build-x86:{RELEASE}-{GCID}

Chapter 5. Installation Changes

Installation changes will be documented in an upcoming version of the Migration Guide.

Chapter 6. SDK Environment (For Structural Changes)

This section covers SDK environment structural changes. DriveOS 7.0 presents a single source of truth by defining envfile at the top of the SDK install directory, which assists in understanding DriveOS variables used throughout the SDK.

The envfile is a text file for development, used for sourcing SDK environment variables.

Security Services SDK Structure Changes		**Backward Compatible**	**Migration Path**	**Platform**	**OS**
PKCS#11 Sample Application is intended for usage on Thor in 7.0.2. README describes how to set a Makefile flag to build for Orin if needed.		Yes	Orin to Orin Orin to Thor	NSR SR	Linux and QNX
Migration Rationale	Channel Id(s) differ between Thor and Orin. Support for NVIDIA channel extension APIs relies on supplying the correct Channel Id according to the SOC in use. The sample application is built for Thor to use the Channel Id(s)				

continues on next page

Table 1 – continued from previous page

Security Services SDK Structure Changes		Backward Compatible	Migration Path	Platform	OS
Steps to Migrate	Refer to Sample Application README file for the following note: This reference application is intended for customers using Thor. If you are using Orin, then please set the TEGRA_CHIPID_TEGRA23 flag in Makefile.tmk to use the correct channel IDs.				

Multimedia SDK Structure Changes		Backward Compatible	Migration Path	Platform	OS
nvm_ide_sci app samples prior to the 7.0.1.0 release included logic to insert fences associated with reference frames to ensure ordered execution. Changes implemented in 7.0.3 invalidated the need to insert fences in this manner. We recommend removing the logic in existing applications (if implemented).		Yes	Orin to Thor	NSR SR	Linux and QNX
Migration Rationale	There is no run time switching allowed in AV use cases. A single Nvdec instance is used throughout. There is no need to insert pre-fence for every reference frame.				
Steps to Migrate	Remove the fences logic in existing applications (if implemented). Refer to the nvm_ide_sci test application for additional information				

NvDisplay Serializer SDK Structure Changes		Backward Compatible	Migration Path	Platform	OS
For the NvDisplay element, one of its sub-elements, serializer, has changed its source files, dts file, and migrated to a new driver architecture.		No	Orin to Thor Orin to Orin	NSR SR	Linux
Migration Rationale	Architecture display driver changes				
Steps to Migrate	Refer to the SDK documentation for nvdisplay serdes. For additional information, refer to the following section.				

Additional Migration Information (for the previous table)

Source File Changes

Platform AV + L

The legacy display serializer driver source files are as follows:

maxim_gmsl_dp_serializer.c", "maxim_gmsl_hdmi_serializer.c

ti_fpdlink_dp_serializer.c

The legacy display serializer driver device tree nodes are as follows:

max_gmsl_dp_ser@i2caddress

max_gmsl_hdmi_ser@i2caddress

ti_fpdlink_dp_ser@i2caddress

Platform AV + Q

Only use legacy. The new nvdisp serdes driver is not yet available for AV + Q.

Device Tree Node Changes

The previous driver device tree nodes are migrated as follows:

The new opcode nvdisp serdes driver source file is `nvdisp_serdes.c`

The new opcode nvdisp serdes driver device tree node is as follows: `nvdisp_serdes@i2caddress`

Additional Information

Supported Hardware

The legacy driver has a version for each supported serializer. It has different drivers for hardware, such as MAXIM DP, MAXIM HDMI, and TI DP serializers.

The new opcode-based nvdisp serdes driver is common for all types of serializers.

Supported Features

For legacy driver, it is important to set fields in device tree, as per SDK documentation, for driver to program the hardware for feature enablement. For example, to enable MST, enable that property in device tree and then driver enables MST functionality during initialization.

The new nvdisp serdes driver does not provide such specific property exposure in device tree. It offers a change in “display-serdes-config” so that you can add OPCODE, and MSTfunctionality is enabled using i2c register writes.

Limitations:

The legacy driver only allows certain functionality to be configurable via device tree. For new functionality support, driver source code must be updated, which requires updates to the SDK documentation.

The new nvdisp serdes driver does not need its source code to be updated. If new functionality must be supported, it can occur by adding another OPCODE in “init-seq”.

SDK Documentation:

The legacy driver exposes the required features.

In new serdes drivers, users working with serializer hardware must be aware of the hardware data sheet and be able to update the OPCODE sequence; it contains details of the register address and values.

PDC SDK Structure Changes		Backward Compatible	Migration Path	Platform	OS
The Thor board config JSON files moved from hardware/nvidia/platform/t264/ configs to a common automotive folder: /hardware/nvidia/platform/ platform-configs. These board configuration files were sorted based on their platform SKU information.		No	Orin to Thor	NSR SR	Linux and QNX
Migration Rationale	The update simplifies cloning an existing board. All files are in a single location sorted by the platform SKU to identify what configuration files are required for the given platform.				
Steps to Migrate	The files that were moved are NVIDIA-specific. Review them as reference for your own custom configurations.				

Chapter 7. Target Changes

This section covers target changes such as Kernel, IFS, Rootfs Structure, and tool changes.

Security Services Target Changes		**Backward Compatible**	**Migration Path**	**Platform**	**OS**
In 7.0.3, Thor does not use TSEC hardware to provide cryptographic services. For backwards compatibility ,the TSEC token and TSECRADAR token are emulated and available in Thor		Yes	Orin to Thor	NSR SR	Linux and QNX
Migration Rationale	To ensure backwards compatibility where possible, it is preferential to emulate these legacy tokens on Thor rather than force a name change that affects customer code				
Steps to Migrate	N/A				

Target Changes		Backward Compatible	Migration Path	Platform	OS
Filesystems renamed from D6.0->D7. GUI Oem-Config added; Ubuntu moved to the 24.04 version.		No (Primarily because FS renaming and U24.04 are not backward compatible, but GUI OEM config is because the Serial UART mode (part of D6.0) still works if display is not available.	Orin to Thor Orin to Orin	NSR	Linux
Migration Rationale	FS renamed to align with PCT for better alignment with use cases				
Steps to Migrate	Install the D7.0 SDK and flash the target. Note: The Docker launch a new NV_WORKSPACE, so no clear is required				

Target Changes		Backward Compatible	Migration Path	Platform	OS
Drive Update VM IFS is removed in 7.0		No	Orin to Thor Orin to Orin	NSR SR	Linux and QNX
Steps to Migrate	The migration does not require user action: DUVM IFS is not customizable				

Target Changes		Backward Compatible	Migration Path	Platform	OS
Changes to key meta data in secure storage		No	Orin to Thor	NSR SR	Linux and QNX
Migration Rationale	To support crypto and larger key sizes				
Steps to Migrate	HSM Tools will be updated to handle the meta data change				

Chapter 8. Tools and Utilities Changes

This section covers target changes such as DevTools and Flashing. The basic tools and utilities changes are documented in the following sections.

- SDK and PDK merge is described in the “Product Changes” section.
- Docker distribution via Artifactory is captured in the “Packaging and Distribution” section
- Tar archive packages are signed by NVIDIA, and public key is included in container
- NSR/SR product naming changes are documented in the “Product Changes” section.

NvDisplay		Backward Compatible	Migration Path	Platform	OS
New tool to calculate static IMP settings - LAPTSA IMP (laptsa-imp)		No	Orin to Thor	NSR	Linux
Migration Rationale	In 7.0.1.0, NvDisplay supported dynamic IMP and users did not have to feed IMP values explicitly to the driver. From 7.0.2.0, NvDisplay will switch to static IMP and users are required to calculate the static-imp values, according to their use case, using the LAPTSA IMP tool and populate “static-imp-data” Device Tree Node. (For additional information, see changes to NvDisplay Tool changes to use LAPTSA IMP tool)				

continues on next page

Table 1 – continued from previous page

NvDisplay		Backward Compatible	Migration Path	Platform	OS
Steps to Migrate	1. Use laptsa-imp host side tool to generate static IMP DT fragment. 2. Add this fragment to the NvDisplay DCE device tree.				

HDE		Backward Compatible	Migration Path	Platform	OS
GCC toolchain upgrade from 9.3 to 13.2.		Yes	Orin to Thor	NSR	Linux
Migration Rationale	Host system Ubuntu 24.04 has 13.2 as the default toolchain version.				
Steps to Migrate	Toolchains are installed in host container under $NW_WORKSPACE/toolchain. The following are available: aarch64–glibc–bleeding-edge-2024.02-1 armv5-eabi–glibc–bleeding-edge-2024.02-1 armv7-eabihf–glibc–bleeding-edge-2024.02-1				

HDE		Backward Compatible	Migration Path	Platform	OS
Toolchain 13.2 migration upgrade from C++ to version 17		Yes	Orin to Thor Orin to Orin	NSR	Linux

continues on next page

Table 3 – continued from previous page

HDE		Backward Compatible	Migration Path	Platform	OS
Migration Rationale	For alignment with QNX 8, which is also moving to C++17.				
Steps to Migrate	Note that embedded Linux component built in GVS migrated to C++17 and samples are updated from to C++ to version 17 as well.				

Flashing Tools Changes		Backward Compatible	Migration Path	Platform	OS
1. Fuse blob generation 2. Command line chipID update. 3. Public key hash change from 1 key to 16keys max, the OEM can choose.		Yes	Orin to Thor	NSR SR	Linux and QNX
Migration Rationale	Updated to support Thor.				
Steps to Migrate	1. Overview: Same command with applicable chip Id (Orin or Thor) 2. No change in fuse burn steps for Orin. For Thor, select up to 16 keys.				

Flashing Tools Changes		**Backward Compatible**	**Migration Path**	**Platform**	**OS**
Customer data format in BCT		No	Orin to Thor	NSR SR	Linux and QNX
Migration Rationale	In 7.2.x.x moving data to the signed section of the customer data for greater security.				
Steps to Migrate	Customer data must be ported from 6.x signed_customer_data and unsigned_customer_data JSON files must be ported to the updated format JSON files in 7.0				

HDE Changes		**Backward Compatible**	**Migration Path**	**Platform**	**OS**
Starting with 7.0.2.1 in DriveOS 7.x, SDK and PDK merged as SDK and we are publishing SDK Docker only. Any additional restricted artifacts are provided via NVIDIA NVONLINE. DriveOS packages used in Dockers are now built as a tar bundle archive instead of a Debian package. Compute stack remains to Debian packages.		No. Debians are deprecated. Distribution mechanism of Dockers changed from NGC to Artifactory.	Orin to Thor Orin to Orin	NSR SR	Linux and QNX

continues on next page

Table 6 – continued from previous page

HDE Changes		Backward Compatible	Migration Path	Platform	OS
Migration Rationale	For the advantages of using Artifactory over NGC: ▶ No requirement to sign up for a separate NGC account. ▶ Publishers no longer must create and manage NGC orgs and publish content separately. ▶ Customers use the API key directly from partners.nvidia.com ▶ Simplified customer workflow				
Steps to Migrate	Refer to the NVIDIA DriveOS 7.0 Installation Guide.				

Chapter 9. Customer-facing Configuration File Changes

This section covers target changes to customer-facing files such as PCT, DT, .JSON, BR-BCT, and startup.

Security Services		Backward Compatible	Migration Path	Platform	OS
OESP PCT is new for Thor in 7.0.3 and later releases. Its configuration requires synchronization with regular PCT (guest configuration)		Yes	Orin to Thor	NSR SR	Linux and QNX
Migration Rationale	OESP TCM is under size restriction. The regular PCT cannot fit into OESP TCM so OESP PCT is used to provide necessary configurations to OESP				
Steps to Migrate	OESP PCT (oesp_platform_config.h) needs synchronization with regular PCT (guest_config.h) When updating tos_keystore_conf, ethernet_cfg, or ethernet_mcfg in guest_config.h, need to update tos_keystore_conf or ethernet_en_vm in oesp_platform_config.h accordingly				

Security Services		Backward Compatible	Migration Path	Platform	OS
In 7.0.3, the number of ephemeral (PKCS11 session) asymmetric keys supported on Thor across all sessions of a single application per token will change due to memory constraints of Thor. The default Token, CCPLEX2, retains support for the same numbers as has been the case since 6.0.6.0, i.e. 8 RSA public, 8 EC public and 8 EC private. Non-default CCPLEX tokens shall each support: 4 RSA public, 4 EC public and 4 EC private. In a future 7.2.x release, there will be a smaller allocation of keys by default. This will be configurable per token so that if a particular token needs more keys, then another token can be adjusted lower.		Yes, for the default token. No otherwise	Orin to Thor	NSR SR	Linux and QNX
Migration Rationale	The change is due to Thor memory constraints				
Steps to Migrate	N/A				

Security Services		Backward Compatible	Migration Path	Platform	OS
Thor will use KDS hardware in the 7.0.3.0 release, which allows for faster key loading and more efficient use of keyslots. To support this hardware there is a change in the number of AES symmetric keys that can be supported, for Thor only: The TSEC token will now support 500 AES symmetric keys. The TSECRADAR token will now support 20 AES symmetric keys. In total, across all CCPLEX tokens, 500 AES symmetric keys. Public and private keys do not use KDS; those allocations are unchanged. Each application that uses a token receives an allocation of ephemeral AES symmetric keys. Tokens will be configured as follows: CCPLEX_SAFETY_2: 8 applications, 64 keys each CCPLEX_DYNAMIC_2: 8 applications, 64 keys each All other CCPLEX tokens: 8 applications, 16 keys each In a future release, each of those 8 applications will receive a smaller allocation of keys by default. This will be configurable per token. If a particular token needs more keys, then another token can be adjusted lower.		No	Orin to Thor	NSR SR	Linux and QNX

continues on next page

Table 3 – continued from previous page

Security Services		Backward Compatible	Migration Path	Platform	OS
Migration Rationale	In Thor, there is a limit imposed by the KDS hardware that dictates the number of keys to support				
Steps to Migrate	N/A				

NvDisplay		Backward Compatible	Migration Path	Platform	OS
Device Tree: New config: "static-imp-data" changes		No	Orin to Thor	NSR	Linux
Migration Rationale	In 7.0.1.0, NvDisplay supported dynamic IMP, and users did not have to feed IMP values explicitly to the driver. From 7.0.2.0, NvDisplay will switch to static IMP, and users are required to calculate the static-imp values, according to their use case, using the LAPTSA IMP tool and populate the "static-imp-data" Device Tree Node. (For additional information, see changes to NvDisplay Tool changes to use LAPTSA IMP tool)				
Steps to Migrate	Use laptsa-imp host-side tool to generate static IMP DT fragment. Add this fragment to the NvDisplay DCE device tree.				

NvDisplay		**Backward Compatible**	**Migration Path**	**Platform**	**OS**
Device Tree: Changes to specify display-timings.		No	Orin to Thor	NSR	Linux
Migration Rationale	In 6.x/7.0.1.0, users specified connector configuration and mode timings using "display-timings" device tree entry. (Refer to 6.x DriveOS SDK section for NvDisplay/Display Serializer.) From 7.0.2.0 "display-timings" entry are deprecated and will be replaced with multiple entries as described below. For actual changes to structure, please refer to NvDisplay section in DRIVE OS SDK guide.				

continues on next page

Table 5 – continued from previous page

NvDisplay		Backward Compatible	Migration Path	Platform	OS

Steps to Migrate	`display-modes/mode`
	`display-modes` specifies a list of modes along with a unique identifier (timing-id). The rest of the schema of the timings is similar to 6.x. The timing-id is then used to look up the full specification of the mode.
	`heads/head`
	Specifies the list of heads and timing-id (unique id as per display-modes/mode) for each head - head timings are programmed by the server.
	`dpys/dpy`
	A dpy specifies the logical display used by the display driver. A dpy is associated with a connector-id and a hw-head-id. The connector-id is the same as the dcbIndex. This dpy has a timing-id that is an identifier to a unique timing configuration as per "display-modes/mode".
	Steps to Migrate
	Refer to the following code.				

Steps to Migrate Example Per the Previous Table

6.x/7.0.1.0 Config:

```
{
    display@13800000 {
        display-timings {
            display-connector-0 {
                dcb-index = <0>;

                stream-0 {
                    timings-phandle = <&mode_ss_1080p>;

                    superframe-info {
                        pipe_x_view: view-0 {
```

(continues on next page)

(continued from previous page)

```
                            x = <0>;
                            y = <0>;
                            width = <1920>;
                            height = <1080>;
                            hfront-porch = <24>;
                            hback-porch = <40>;
                            hsync-len = <16>;
                            vfront-porch = <3>;
                            vback-porch = <18>;
                            vsync-len = <10>;
                        };

                        pipe_y_view: view-1 {
                            x = <1920>;
                            y = <0>;
                            width = <1920>;
                            height = <1080>;
                            hfront-porch = <24>;
                            hback-porch = <40>;
                            hsync-len = <16>;
                            vfront-porch = <3>;
                            vback-porch = <18>;
                            vsync-len = <10>;
                        };
                    };
                };

                stream-1 {
                    timings-phandle = <&mode_ss_1080p>;

                    superframe-info {
                        pipe_z_view: view-0 {
                            x = <0>;
                            y = <0>;
                            width = <1920>;
                            height = <1080>;
                            hfront-porch = <24>;
                            hback-porch = <40>;
                            hsync-len = <16>;
                            vfront-porch = <3>;
                            vback-porch = <18>;
                            vsync-len = <10>;
                        };

                        pipe_u_view: view-1 {
                            x = <1920>;
                            y = <0>;
                            width = <1920>;
                            height = <1080>;
                            hfront-porch = <24>;
                            hback-porch = <40>;
                            hsync-len = <16>;
                            vfront-porch = <3>;
                            vback-porch = <18>;
                            vsync-len = <10>;
                        };
```

(continues on next page)

(continued from previous page)

```
                    };
                };
            };
        };

        mode_ss_1080p: 3840-1080-60Hz {
            clock-frequency-khz = <266640>;
            hactive = <3840>;
            vactive = <1080>;
            hfront-porch = <48>;
            hback-porch = <80>;
            hsync-len = <32>;
            vfront-porch = <3>;
            vback-porch = <18>;
            vsync-len = <10>;
            rrx1k = <60000>;
        };

        mode0: 1920-1080-60Hz {
            clock-frequency-khz = <148500>;
            hactive = <1920>;
            vactive = <1080>;
            hfront-porch = <88>;
            hback-porch = <148>;
            hsync-len = <44>;
            vfront-porch = <4>;
            vback-porch = <36>;
            vsync-len = <5>;
            rrx1k = <60000>;
        };

        mode1: 1280-720-60Hz {
            clock-frequency-khz = <74250>;
            hactive = <1280>;
            vactive = <720>;
            hfront-porch = <110>;
            hback-porch = <220>;
            hsync-len = <40>;
            vfront-porch = <5>;
            vback-porch = <20>;
            vsync-len = <5>;
            rrx1k = <60000>;
        };
    };
};
```

7.0.2.0 Config:

```
/ {
    display@8808c00000 {
        display-modes {
            mode-0 {
                timing-id = <0>;
                clock-frequency-khz = <148500>;
                hactive = <1920>;
                vactive = <1080>;
                hfront-porch = <88>;
```

(continues on next page)

(continued from previous page)

```
                hback-porch = <148>;
                hsync-len = <44>;
                vfront-porch = <4>;
                vback-porch = <36>;
                vsync-len = <5>;
                rrx1k = <60000>;
                pps-data = [
                    11 00 00 89 30 80 04 38
                    07 80 04 38 03 c0 03 c0
                    02 00 03 58 00 20 73 3e
                    00 0d 00 0f 00 1d 00 0e
                    18 00 10 f0 03 0c 20 00
                    06 0b 0b 33 0e 1c 2a 38
                    46 54 62 69 70 77 79 7b
                    7d 7e 01 02 01 00 09 40
                    09 be 19 fc 19 fa 19 f8
                    1a 38 1a 78 22 b6 2a b6
                    2a f6 2a f4 43 34 63 74
                    00 00 00 00 00 00 00 00
                    00 00 00 00 00 00 00 00
                    00 00 00 00 00 00 00 00
                    00 00 00 00 00 00 00 00
                    00 00 00 00 00 00 00 00 ];
            };
        };
    };
};
```

```
/ {
    display@8808c00000 {
        dpys {
            dpy-0 {
                hw-head-id = <0>;
                timing-id = <0>;
                connector-id = <0>;
            };
            dpy-1 {
                hw-head-id = <1>;
                timing-id = <0>;
                connector-id = <1>;
            };
            dpy-2 {
                hw-head-id = <2>;
                timing-id = <0>;
                connector-id = <2>;
            };
        };
        heads {
            head-0 {
                hw-head-id = <0>;
                timing-id = <0>;
            };
            head-1 {
                hw-head-id = <1>;
                timing-id = <0>;
            };
```

(continues on next page)

(continued from previous page)

```
            head-2 {
                hw-head-id = <2>;
                timing-id = <0>;
            };
        };
    };
};
```

NVIPC		Backward Compatible	Migration Path	Platform	OS
Changes to the per endpoint streaming mode buffer limit. Added per endpoint streaming mode buffer limit, which effects per endpoint overall buffer size usage. Example: Before the migration (in 6.0) the following command works without errors (with -s 64000000 & -p 2 –> -s (size of the packet), -p (number of packets)) ./test_nvscic2c_stream -t cons -c nvscic2c_pcie_s0_c6_1 -v 1 -p 2 -s 64000000 -l -o /tmp/output_perf.csv -i 10000 & After the migration, the same command fails due to the per endpoint buffer limit of 67MB implemented in 7.0.		No	Orin to Thor Orin to Orin	NSR SR	Linux and QNX
Migration Rationale	To align with security ARR configuration of NVIPC Buffer Limits.				

continues on next page

Table 6 – continued from previous page

NVIPC		Backward Compatible	Migration Path	Platform	OS
Steps to Migrate	Adjust commands to align with the buffer limits.				

PCD		Backward Compatible	Migration Path	Platform	OS
1. Moving to Generic DT format from legacy format to align with upstream Linux expectations. 2. The mechanism of storage configuration is changing to a simplified and automated interface. 3. Board configuration files changed from JSON to YAML		No	Orin to Thor Orin to Orin	NSR SR	Linux and QNX
Migration Rationale	Generic DT: To align with upstream Linux expectation Storage Config and Board Config Update: Enhancements for ease of configuration for customer and tooling support.				
Steps to Migrate	Configuration update steps with reference examples will be provided in the upcoming revisions of the migration guide and release documentation.				

TCF		Backward Compatible	Migration Path	Platform	OS
Of the many changes related to Boot Flow in 7.0, a few are noted below: MCE component is no longer available. MB1 boots the boot core. TSEC is loaded in MB2. iGPU firmware is loaded in MB1. NVDEC FW is not loaded by MB1. New Security enclaves: OESP and SB. MB1 loads them. New AUXP, which MB2 loads. MB1BCT and MB2 BCT flags changed.		No	Orin to Thor	NSR SR	Linux and QNX
Migration Rationale	Changes are required due to Orin to Thor hardware platform changes and boot flow architecture updates				
Steps to Migrate	In-depth details will be provided in the upcoming revisions of the migration guide and developer guide documentation.				

Security Services		Backward Compatible	Migration Path	Platform	OS
No longer need to limit keyslots if the operation is not multipart. Enforcement of keyslots is no longer on a per token basis.		No	Orin to Thor	NSR SR	Linux and QNX

continues on next page

Table 9 – continued from previous page

Security Services		Backward Compatible	Migration Path	Platform	OS
Migration Rationale	Hardware changes in Thor increases flexibility in using keyslots. The change also improves performance and latency of key load operations.				
Steps to Migrate	PCT requires updating to replace the old configuration (which limits the number of keyslots an app can use) to the new configuration. Applications should update which SE Engine needs to be used for crypto operations.				

Chapter 10. Security Services Changes

API Changes		Backward Compatible	Migration Path	Platform	OS
Pre 7.0.3.0: Encryption: No parameter is required for CKM_AES_CTR. Implicit counter width is 32. Use C_NVIDIA_EncryptGetIV to retrieve the IV. Decryption and encryption with CKA_NVIDIA_CALLER_NONCE: Requires CK_AES_CTR_PARAMS with ulCounterBits set to 32 **Post 7.0.3.0:** Encryption (with and without CKA_NVIDIA_CALLER_NONCE) and Decryption: CK_AES_CTR_PARAMS parameters are required for CKM_AES_CTR. Supports variable counter width (such as ulCounterBits) from 1 to 32		Yes, If no parameter is supplied for CKM_AES_CTR encryption, the implicit value of 32 is used for the counter width	Orin to Orin Orin to Thor	NSR SR	Linux and QNX
Migration Rationale	The change improves the CKM_AES_CTR mechanism implementation to allow for variable counter widths.				

continues on next page

Table 1 – continued from previous page

API Changes		Backward Compatible	Migration Path	Platform	OS
Steps to Migrate	Encryption (with and without CKA_NVIDIA_CALLER_NONCE) and Decryption: CK_AES_CTR_PARAMS parameter is required for CKM_AES_CTR. Supports variable counter width (such as ulCounterBits) from 1 to 32. If no parameter is supplied for CKM_AES_CTR encryption, the implicit value of 32 is used for the counter width				

API Changes		Backward Compatible	Migration Path	Platform	OS
With 7.2.1.0, in the event of a failure in the PKCS#11 Library Known Answer Test, DriveOS shall not transition to INIT_DONE state		Yes	Orin to Orin Orin to Thor	NSR SR	Linux and QNX
Migration Rationale	If initialization fails, the system must not transition to INIT_DONE. Added the requirement to explicitly state that "DriveOS shall not transition to INIT_DONE state on any failure in Known Answer Test (KAT)				
Steps to Migrate	Not applicable				

API Changes		Backward Compatible	Migration Path	Platform	OS
With 7.0.3, any function call to PKCS#11 Library C_GenerateRandom with 0 data length is classed as a valid call returning CKR_OK, whereas it used to return CKR_ARGUMENTS_BAD		No	Orin to Thor Thor to Thor	NSR SR	Linux and QNX
Migration Rationale	C_GenerateRandom function returned CKR_ARGUMENTS_BAD for 0 length data, but according to specification it is allowed. The change is to return CKR_OK, but not modify data				
Steps to Migrate	For C_GenerateRandom, if 0 data length is used for error checks, check for 0 data directly instead of calling C_GenerateRandom				

API Changes		Backward Compatible	Migration Path	Platform	OS
With 7.0.3, PKCS#11 Library C_GetMechanismInfo, RSA key size is now returned with modulus size in bits rather than a fixed value of 4 bytes		No	Orin to Thor Thor to Thor	NSR SR	Linux and QNX
Migration Rationale	C_GetMechanismInfo, function returned the allowed exponent size for RSA keys, but since this is a fixed value for most keys it is of no use to users. Instead, RSA keys are more commonly measured by the modulus size, 2K, 3K, 4K keys				

continues on next page

Table 4 – continued from previous page

API Changes		Backward Compatible	Migration Path	Platform	OS
Steps to Migrate	If allowed RSA keys are checked with C_GetMechanismInfo, either update to check modulus size or if exponent size is required, use NVPKCS11_MAX_KEY_EXPONENT and NVPKCS11_MIN_KEY_EXPONENT defines from nvpkcs11_public_defs.h				

API Changes		Backward Compatible	Migration Path	Platform	OS
With 7.0.3, PKCS#11 Library C_SignInit, the return value for Private keys is updated from CKR_DEVICE_ERROR to CKR_OPERATION_ACTIVE when all key slots are currently in use		No	Orin to Thor Thor to Thor	NSR SR	Linux and QNX
Migration Rationale	Changed for consistency in error handling. RSA and EC public keys return CKR_OPERATION_ACTIVE if keyslot is in use, whereas EC Private keys returned CKR_DEVICE_ERROR				
Steps to Migrate	Update accordingly if relying on this error to trigger a retry of the call to C_SignInit				

Chapter 11. Safety Services Changes

Target Changes		Backward Compatible	Migration Path	Platform	OS
An API is added for FSI-CCPLEX-COM. This will not affect backward compatibility of existing APIs		Yes	Orin to Thor Orin to Orin	NSR SR	Linux and QNX
Migration Rationale	The total number processes are limited, and due to limitation of NvFsiCom support to manage multiple channels within a process				
Steps to Migrate	FsiCom APIs allow the current usage with multiple threads within a single process; that performance can be offloaded to individual threads as well instead of managing everything within the same thread of a process.				

Chapter 12. MCU Changes

Target Changes		Backward Compatible	Migration Path	Platform	OS
sMCU is changing from AURIX to Renesas. There will be major changes in flashing and other hardware-related topics		No	Orin to Thor	NSR SR	Linux and QNX
Migration Rationale	Hardware platform changes from 6.0				
Steps to Migrate	Integration guide for MCU will be provided in the upcoming revisions.				

Chapter 13. FSI Changes

Note

This section will be covered in the upcoming revisions of the Migration Guide, as required.

Chapter 14. x86 Changes

DriveOS 7.0 supports all previous 6.0 GPUs plus newer ones, including RTX Ada generation workstations that use AD102 and AD104. x86 EMU (emulation of Tegra imaging stack) remains supported for DriveWorks on these GPUs but is not offered by NVIDIA DRIVE OS x86.

Chapter 15. API/Interface Changes

The following sections cover API and interface changes such as DriveWorks, Camera, NvStreams, NvMedia, cuPVA. The sections also cover dynamic changes, such as the programming sequence and static changes to header files.

Note

The cuPVA SDK API is backwards compatible when migrating to version 7.0, and the PVA SDK maintains source compatibility between major releases. Download the version of the PVA SDK that corresponds to your NVIDIA DriveOS™ version and rebuild your application.

15.1. Support Matrix for Multimedia Entities through 7.0 Releases

15.1.1. Configuration

- SoC: Thor
- OS: Linux NSR, QNX NSR

Legend:

SP - Supported on production builds

S - Supported only through debug/development overlay (Not compliant for production)

Feature	7.0.0.0 – 7.0.0.2	7.0.1.0 and Above
Image Encode (IEP)		
H.264	SP	SP
H.265 (HEVC)	SP	SP
Image Decode (IDE)		
H.264	SP	SP
H.265 (HEVC)	SP	SP
VP8	SP	S
VP9	SP	S
MPEG-2	SP	S
MPEG-4	SP	S
VC-1/WMV9	SP	S
AV1	SP	S
Image Optical Flow Accelerator (IOFA)		
Optical Flow/Stereo Processing	SP	SP
Image JPEG Encode (IJPE)		
JPEG Encoding	SP	SP
Image JPEG Decode (IJPD)		
JPEG Decoding	SP	S

15.2. NvStreams Changes

NvStreams		Backward Compatible	Migration Path	Platform	OS

The following APIs, which return `NvSciError_BadParameter` for invalid block in 6.0, will return `NvSciError_StreamBadBlock` instead:
`NvSciStreamBlockConnect`
`NvSciStreamProducerCreate`
`NvSciStreamProducerCreate2`
`NvSciStreamConsumerCreate`
`NvSciStreamConsumerCreate2`
`NvSciStreamBlockEventQuery`
`NvSciStreamBlockDelete`
`NvSciStreamBlockEventServiceSetup`
`NvSciStreamBlockInternalEventServiceSetup`
		No	Orin to Thor
			Orin to Orin
				NSR
				SR
					Linux and QNX

Migration Rationale	In 5.2, NvSciStream used NvSciError_BadParameter for all invalid input, like null pointer, invalid block handle, invalid packet handle, etc. The user could not tell the exact issue from the error code. In 6.0, specific error code: NvSciError_StreamBadBlock, for invalid block handle was introduced, but not all APIs migrated to this new error code. 7.0 moves all the APIs to this new error code (NvSciError_StreamBadBlock) consistently.				

continues on next page

Table 1 – continued from previous page

NvStreams		Backward Compatible	Migration Path	Platform	OS

Steps to Migrate	Sample old application implementation:
	NvSciError err;
	err = NvSciStreamProducerCreate(pool, &producer);
	if (err != NvSciError_Success) {
	if (err == NvSciError_BadParameter) {
	printf("NvSciError_BadParameter received from NvSciStreamProducerCreate, pool block could be invalid\n");
	}
	exit();
	}				

NvStreams		Backward Compatible	Migration Path	Platform	OS
NvSciSync timestamp behavior changes for CUDA NvSciSync on Linux. CUDA-provided timestamps will be in microseconds instead of nanoseconds. NvSciSync on DRIVE Linux will obtain time via a CCPLEX register tracking with ARM TSC (previously from CLOCK_MONOTONIC). This change exists from 6.0.9.3` prior releases.		No	Orin to Thor Orin to Orin	NSR SR	Linux and QNX
Migration Rationale	It was confusing to have a timestamp from a different unit or source. This update makes timestamps across engines and the CCPLEX consistent.				

continues on next page

Table 2 – continued from previous page

NvStreams		Backward Compatible	Migration Path	Platform	OS
Steps to Migrate	Application code that assumes nanoseconds for NvSciSync timestamps from CUDA requires a change to the scaling factor. Application code that assumes CLOCK_MONOTONIC and adjusts it to be consistent with ARM TSC requires removal; use the raw value instead.				

15.3. NvDisplay Changes

NvDisplay API Functional Behavioral Changes (Dynamic Changes: Programming Sequence)		Backward Compatible	Migration Path	Platform	OS
For 7.0.2.0 Production profiles, the display mode timings will be set when the driver loads based on the timings in the Device Tree (see the "Display Serializer" page in the NvDisplay SDK docs). Changing the mode timings dynamically will not be supported. Client-initiated modeset requests will return success only if the requested mode timings match the mode that was already set by the driver, and will return failure otherwise.		No. Note: Prior code that relies on dynamic modeset will not work, unless the mode being set happens to match the mode that is specified in the device tree and set by the driver during init.	Orin to Thor Orin to Orin	NSR SR	Linux and QNX

continues on next page

Table 3 – continued from previous page

NvDisplay API Functional Behavioral Changes (Dynamic Changes: Programming Sequence)		Backward Compatible	Migration Path	Platform	OS
Migration Rationale	The new display driver implementation is designed to accommodate multiple guest virtual machines. The setting of the display mode is an operation that would affect other virtual machines, so no guest VM should be able to control it. Instead, the client will specify the mode in the device tree and the driver will set the mode during initialization.				

continues on next page

Table 3 – continued from previous page

NvDisplay API Functional Behavioral Changes (Dynamic Changes: Programming Sequence)		Backward Compatible	Migration Path	Platform	OS
Steps to Migrate	Change: For 7.0.2.0 Production profiles, the display mode timings will be set when the driver loads based on the timings in the Device Tree (see the "Display Serializer" topic in the NvDisplay SDK documentation). Changing the mode timings dynamically will not be supported. Client-initiated modeset requests will return success only if the requested mode timings match the mode that was already set by the driver, and will return failure otherwise. Relevant WFD API: Clients dynamically set the display mode timings via `wfdSetPortMode` (section 4.4.4 of the OpenWFD specification). Since dynamic mode changes will not be supported, this function will fail unless the mode timings requested match the mode timings that were already set by the driver during driver initialization. Clients should know what their desired mode is and set it in the Device Tree instead of dynamically in their client code.				

15.4. Multimedia Changes

15.4.1. Multimedia (Such as NvMedia, NvMMM-Legacy, and IDE NvJPG)

API Functional Behavioral Changes	Backward Compatible	Migration Path	Platform	OS
IDE for VP8, VP9, MPEG-2, MPEG-4, VC-1, AV1 and IJPD will be available only for development overlay and not in production filesystem	Yes	Orin to Thor Orin to Orin	NSR	Linux only
Migration Rationale	The update streamlines cybersecurity efforts for all codecs.			
Steps to Migrate	IDE (VP8, VP9, MPEG-2, MPEG-4, VC-1, AV1) and IJPD will be supported in the debug/development profile. nvm_ide_sci test app will only accept h264/h265 codecs in production mode. Other codecs are not supported.			

15.4.2. IOFA, IEP, IDE, IJPE, IJPD Changes

API Functional Behavioral Changes		Backward Compatible	Migration Path	Platform	OS
Changes to the following APIs: IOFA, IEP, IDE, IJPE, IJPD A new API queries the device properties. This data determines if a specific engine is available and the number of instances available. Additionally, the API is used to select a specific engine instance for processing. The use of this API is optional. The API is primarily intended to ensure that the same applications can execute seamlessly on different SKUs/platform configurations. In such cases, the applications can choose the NVENC/NVDEC/OFA/NVJPG hardware instance to be used (based on the list of available instances) at runtime based on the SKU/platform on which the application is running.		Yes	Orin to Thor Only	NSR SR	Linux and QNX
Migration Rationale	Hardware architecture changes in Thor				

There is a major internal hardware architectural change for the multimedia hardware accelerator engines (NVENC, NVDEC, NVJPG and OFA) introduced from the Thor chip family onward, resulting in these engines being integrated into the iGPU chiplet.

Specific SKUs in the Thor chip family could house multiple instances of these engines. Additionally, Drive Thor platforms include support for multi-socket configurations (Thor with NUMA configuration), where the multimedia hardware accelerator on each of the sockets needs to be addressable.

The NvMedia Multimedia APIs (NvMediaIEP, NvMediaIOFA, NvMediaIDE, NvMediaIJPE, NvMediaIJPD) are extended to support addressing combinations of hardware engines for single- and multi-socket configurations. As a result of this change, a new API is introduced in each of NvMediaIEP, NvMediaIOFA, NvMediaIDE, NvMediaIJPE, NvMediaIJPD APIs to enable querying the list of available hardware engine instances and its properties. Applications need to select the required hardware instance from this list and subsequent operations (initialization, setup, processing, de-initialization) performed on this instance.

Note

1. The number of engines available could vary based on the chip SKU, and multiple instances of a given engine may not be available for all engines.
2. NvMediaDeviceInfo.numaDomainId is an important attribute for the multi-socket Drive Thor configuration. Each die in this case has an associated CPU, iGPU and local physical memory. Specific instances can be addressed using the numaDomainId. Applications should use this information to select a local/preferred GPU device from the list to ensure acceptable performance.

15.4.2.1 APIs

15.4.2.1.1 NvMediaIEP

```
/**
 * @brief Queries the IEP device list
 *
 * @pre NvMediaIEPGetVersion()
 * @post NvMediaIEPCreate()
 *
 * @usage
 * - Allowed context for the API call
 * - Interrupt handler: No
 * - Signal handler: No
 * - Thread-safe: Yes
 * - Re-entrant: No
 * - Async/Sync: Sync
 * - Required privileges: None
 *
 * - API group
 *   - Init: Yes
 *   - Runtime: No
 *   - De-Init: No
 *
 * @param[in] deviceList A pointer to a @ref NvMediaDeviceList structure
 *                        of the client.
 * @return ::NvMediaStatus The status of the operation.
 *         Possible values are:
 *         - ::NVMEDIA_STATUS_OK
 *         - ::NVMEDIA_STATUS_BAD_PARAMETER if the pointer is invalid.
 */

NvMediaStatus NvMediaIEPQueryDevices(
    NvMediaDeviceList *deviceList
);
```

15.4.2.1.2 NvMediaIOFA

```
/**
 * @brief Query OFA devices present
 *
 * @pre NvMediaIOFAGetVersion()
 * @post NvMediaIOFACreate()
 *
 * @usage
 * - Allowed context for the API call
 * - Interrupt handler: No
 * - Signal handler: No
 * - Thread-safe: Yes
 * - Re-entrant: No
 * - Async/Sync: Sync
 * - Required privileges: None
 *
 * - API group
 *   - Init: Yes
 *   - Runtime: No
 *   - De-Init: No
 *
 * @param[in] deviceList
 * A pointer to the @ref NvMediaIofaDeviceList to populate device size
 * and info
 * Non-NULL - valid pointer address
 *
 * @return The completion status of the operation:
 * - ::NVMEDIA_STATUS_OK if the call is successful.
 * - ::NVMEDIA_STATUS_ERROR if there is an internal error in processing.
 */

NvMediaStatus
NvMediaIOFAQueryDevices(
    NvMediaIofaDeviceList *deviceList
);
```

15.4.2.1.3 NvMediaIDE

```
/**
 * @brief Queries the HW device list for IDE
 *
 * @pre NvMediaIDEGetVersion()
 * @post NvMediaIDECreate()
 *
 * @usage
 * - Allowed context for the API call
 * - Interrupt handler: No
 * - Signal handler: No
 * - Thread-safe: Yes
 * - Re-entrant: No
 * - Async/Sync: Sync
 * - Required privileges: None
 *
 * - API group
```

(continues on next page)

(continued from previous page)

```
 *   - Init: Yes
 *   - Runtime: No
 *   - De-Init: No
 *
 * @param[in] deviceList A pointer to a @ref NvMediaDeviceList structure
 *                       of the client.
 *
 * @return ::NvMediaStatus The status of the operation.
 *         Possible values are:
 *         - ::NVMEDIA_STATUS_OK
 *         - ::NVMEDIA_STATUS_BAD_PARAMETER if the pointer is invalid.
 */

NvMediaStatus
NvMediaIDEQueryDevices(
    NvMediaDeviceList *deviceList
);
```

15.4.2.1.4 NvMediaIJPE

```
/**
 * @brief Queries the IJPE device list
 *
 * @pre NvMediaIJPEGetVersion()
 * @post NvMediaIJPECreate()
 *
 * @usage
 * - Allowed context for the API call
 * - Interrupt handler: No
 * - Signal handler: No
 * - Thread-safe: Yes
 * - Re-entrant: No
 * - Async/Sync: Sync
 * - Required privileges: None
 *
 * - API group
 *   - Init: Yes
 *   - Runtime: No
 *   - De-Init: No
 *
 * @param[in] deviceList A pointer to a @ref NvMediaDeviceList structure
 *                       of the client.
 *
 * @return ::NvMediaStatus The status of the operation.
 *         Possible values are:
 *         - ::NVMEDIA_STATUS_OK
 *         - ::NVMEDIA_STATUS_BAD_PARAMETER if the pointer is invalid.
 */

NvMediaStatus
NvMediaIJPEQueryDevices(
    NvMediaDeviceList *deviceList
);
```

15.4.2.1.5 NvMediaIJPD

```
/**
 * @brief Queries the IJPD device list
 *
 * @pre NvMediaIJPDGetVersion()
 * @post NvMediaIJPDCreate()
 *
 * @usage
 * - Allowed context for the API call
 * - Interrupt handler: No
 * - Signal handler: No
 * - Thread-safe: Yes
 * - Re-entrant: No
 * - Async/Sync: Sync
 * - Required privileges: None
 *
 * - API group
 *   - Init: Yes
 *   - Runtime: No
 *   - De-Init: No
 *
 * @param[in] deviceList A pointer to a @ref NvMediaDeviceList structure
 *                       of the client.
 *
 * @return ::NvMediaStatus The status of the operation.
 *         Possible values are:
 *         - ::NVMEDIA_STATUS_OK
 *         - ::NVMEDIA_STATUS_BAD_PARAMETER if the pointer is invalid.
 */

NvMediaStatus
NvMediaIJPDQueryDevices(
    NvMediaDeviceList *deviceList
);
```

15.4.2.1.6 Data Types

```
/**
 * @brief The 128-bit GPU device id
 */
typedef struct {
    /** array of id */
    uint8_t id[NVMEDIA_HW_DEVICE_ID_LEN];
} NvMediaDeviceGID;

/**
 * \brief Hold information about a MM device
 */
typedef struct {
    /**
     * @brief The GPU UUID – the primary device identifier
     * \n The GPU UUID is the primary identifier of the GPU device. The
     * main purpose of the UUID is to reliably identify a device in various
     * inter-operability scenarios.
```

(continues on next page)

(continued from previous page)

```
     */
    NvMediaDeviceGID uuid;

    /**
     * @brief The GPU UUID for the hardware device
     * \n This is the identifier for the hardware device. The hardware
     * device identifier may be different to the primary device identifier
     * in case the GPU is a multi-instance GPU (MIG) device.
     * \n The primary use case for the hardware identifier is cross-driver
     * interoperability, where hardware identifier is needed rather than the
     * logical identifier.
     */
    NvMediaDeviceGID hwGid;

    /**
     * @brief The NUMA domain id in which the GPU device is present.
     * \n A Non-Uniform Memory Access (NUMA) architecture partitions a
     * system into multiple domains, typically each associated with its own
     * local CPU and physical memory. Performance of memory accesses to a
     * remote domain is significantly lower than that of accesses to the
     * local domain's physical memory.
     * \n Applications need to be NUMA-aware for good performance
     * under a NUMA architecture. Application threads should use
     * domain-local hardware resources for higher throughput and lower latency.
     */
    uint32_t numaDomainId;

} NvMediaDeviceInfo;

/**
 * \brief Hold information about list of a specific MM device
 */
typedef struct {
    // @brief The size of the Device list.
    uint32_t deviceListSize;

    /* @brief MM Engine Information.
     * This struct is the main source for device info uuid, hwid, numaDomainId
     */
    NvMediaDeviceInfo deviceList[NVMEDIA_HW_DEVICE_MAX_COUNT];
} NvMediaDeviceList;
```

15.4.2.1.7 API Call Sequence

Following is the adapted API call sequence for NvMediaIEP post the introduction of `NvMediaIEPQueryDevices()`:

```
/* Query the list of NVENC HW Instances on the chip */

NvMediaDeviceList deviceList;

NvMediaIEPQueryDevices(&deviceList);

/* deviceList.deviceListSize provides the number of NVENC instances.
```

(continues on next page)

(continued from previous page)

```
 * The following members provide info on the device properties:
 *   deviceList.deviceList[index].uuid.id,
 *   deviceList.deviceList[index].hwGid.id,
 *   deviceList.deviceList[index].numaDomainId
 */

/* Select the NvMediaEncoderInstanceId enumeration based on the “index”
 * selected in deviceList. For instance, if the target UUID and HW GID
 * match with the deviceList at ‘index’ 0, then the instanceId to be passed
 * to all subsequent NvMediaIEP APIs is NVMEDIA_ENCODER_INSTANCE_0.
 * Similarly, it would be NVMEDIA_ENCODER_INSTANCE_1 if ‘index’ 1 is
 * selected.
 */

NvMediaIEP *iepCtxHandle = NvMediaIEPCreate(
    encodeType,
    encoderInitParams,
    bufReconciledList,
    outBuffering,
    instanceId
);

<No change to sequence of calls/arguments for subsequent APIs>
```

15.4.3. Video Interlace Compositor (VIC) Changes

API Functional Behavioral Changes		Backward Compatible	Migration Path	Platform	OS
Dynamic changes, such as to the programming sequence		No	Orin to Thor Only	NSR SR	Linux and QNX
Migration Rationale	VIC hardware change				
Steps to Migrate	The LDC region parameters (NvMedia2DLdcRegionParameters type) need to be adjusted to work with the strict height alignment requirement on Thor. With warp maps, this will affect the warp map contents – the position and potentially the total count of the control points changes.				

API Functional Behavioral Changes		Backward Compatible	Migration Path	Platform	OS
LDC and TNR API		No, however Orin will use the old API on 7.0.1 (deprecated), which will be deleted in 7.0.2	Orin to Thor Orin to Orin	NSR SR	Linux and QNX
Migration Rationale	Hardware architecture change in Thor				
Steps to Migrate	Use 2D API for Init, Deinit, and task submission calls. Modify LDC calls to use the new API. Refer to the following details.				

A major internal hardware architectural change for the 2D hardware allows you to perform per-layer compose and TNR operations in a single pass, without the need to program these operations as separate actions through different APIs (Compose, TNR), and consequently resort to storing an intermediate result to memory.`

- NvMedia2D API is updated to support LDC and TNR operations in a combination with compose operation`
  - Same programming model as `NvMediaLDC`
  - Similar API types to `NvMediaLDC`
    - `NvMediaLdcWarpMapParameters` will become `NvMedia2DLdcWarpMapParameters`
  - Similar API functions to NvMediaLDC
    - `NvMediaLdcCreateParameters()` will become `NvMedia2DCreateLdcWarpMap()`
    - `NvMediaLdcSetWarpMapParameters()` will become `NvMedia2DSetSrcLdcWarpMap()`
  - Separation of LDC and TNR state
    - `NvMediaLDC` API uses a single parameter object to hold both LDC and TNR state (`NvMediaLdcParameters`)
    - `NvMedia2D` API will use separate objects instead (`NvMedia2DLdcWarpMap`, `NvMedia2DLdcMaskMap`, and `NvMedia2DTnrState`).
- `NvMedia2D` API is updated with new functions for querying chip capabilities
  - `NvMedia2DGetCapabilities()`

and a new associated type:

- `NvMedia2DCapabilities`

through which a user can query, among others, whether a current implementation supports Compose operation, LDC operation, and/or Compose and LDC operations in a single pass.

- NvMedia2D API is updated with new functions for reading CRCs
  - `NvMedia2DSetChecksumMode()`
  - `NvMedia2DGetChecksum()`

and new associated types:

- `NvMedia2DChecksum`
- `NvMedia2DChecksumMode`
- NvMediaLDC API will not work with VIC products
  - API Functions will return `NVMEDIA_STATUS_NOT_SUPPORTED`
  - Client code must be ported to use `NvMedia2D` API instead

Key points of migrating from the old to the new API:

- Client code needs to port from using `NvMediaLDC` API to using `NvMedia2D` API.

Example of LDC only operation (applicable for both Orin and Thor):

```
NvMedia2DCreate(...);

NvMedia2DCreateLdcWarpMap(...);

NvMedia2DSetSrcLdcWarpMap(...);

NvMedia2DCompose(params);
```

- Existing code can be optimized to use the combined Compose+LDC path.

Example of Compose + LDC only operation (applicable for Thor only):

```
NvMedia2DCreate(...);

NvMedia2DCreateLdcWarpMap(...);

NvMedia2DSetSrcBlendMode(...);

NvMedia2DSetSrcLdcWarpMap(...);

NvMedia2DCompose(params);
```

Attempts to use combined Compose+LDC path on Orin will fail.

Example of invalid usage attempt of Compose + LDC (Orin):

```
NvMedia2DCreate(...);

NvMedia2DCreateLdcWarpMap(...);

NvMedia2DSetSrcBlendMode(...);

NvMedia2DSetSrcLdcWarpMap(...);

NvMedia2DCompose(params); // The compose call will fail!
```

- Client code that targets both Orin-based and Thor-based products may need to maintain a conditional path to apply a combined Compose+LDC path on Thor only.

Clients can use a new capabilities API to check whether a combined Compose+LDC is available.

Example of conditionally applying Compose + LDC operation (either Orin or Thor):

```
NvMedia2DCapabilities const *caps;
caps = NvMedia2DGetCapabilities(NvMedia2D const * const handle);

if (caps->supportsComposeLdc)
{
    NvMedia2DCreate(...);
    NvMedia2DCreateLdcWarpMap(...);
    NvMedia2DSetSrcBlendMode(...);
    NvMedia2DSetSrcLdcWarpMap(...);
    NvMedia2DCompose(params);
}
else
{
    // Context for LDC operation
    NvMedia2DCreate(...);
    // Separate LDC operation
    NvMedia2DCreateLdcWarpMap(...);
    NvMedia2DSetSrcLdcWarpMap(...);
    NvMedia2DCompose(params);

    // Context for Compose operation
    NvMedia2DCreate(...);
    // Separate Compose operation
    NvMedia2DSetSrcBlendMode(...);
    NvMedia2DCompose(params);
}
```

15.5. DRIVE Update Changes

API Functional Behavioral Changes (Dynamic Changes: Programming Sequence)		Backward Compatible	Migration Path	Platform	OS
The BRBCT marker-based boot chain selection is removed in Thor		No	Orin to Thor	NSR	Linux and QNX
Migration Rationale	Due to changes to Thor BootROM, we only have sequential boot chain selection or GPIO-based boot chain selection now.				
Steps to Migrate	Migrate to GPIO-based boot chain selection. This has been the default option for Orin				

DRIVE Update Changes		Backward Compatible	Migration Path	Platform	OS
The DRIVE Update router parent path will change from "/gos-a/ to "/"		No	Orin to Thor Orin to Orin	NSR SR	Linux and QNX
Migration Rationale	In DRIVE Update 7.0, Update VM was removed, resulting in only having "/gos-a" for Guest OS Tegra A. The DRIVE Update Router, however, will look for upper level "/". This may create issues for the router. To reduce this potential risk, in DRIVE Update 7.0, for all units in Guest OS Tegra A, the router parent path will change from "/gos-a" to "/".				
Steps to Migrate	Refer to the NVIDIA DriveOS 7.0 Installation Guide.				

DRIVE Update Changes		Backward Compatible	Migration Path	Platform	OS
Changes to the following dupkg templates: ▶ dupkg_dudelta_pull_template ▶ dupkg_dudiff_template		Yes	Orin to Thor Orin to Orin	NSR SR	Linux and QNX
Migration Rationale	Implemented a better solution for the delta update in 6.5.2.0, which changes the delta generation flow.				

continues on next page

Table 10 – continued from previous page

DRIVE Update Changes		Backward Compatible	Migration Path	Platform	OS
Steps to Migrate	The new delta solution template generates all necessary packages/scripts to complete a delta update to a new version. There is no need to manually handle special partitions. The previous delta generation dupkg command line is: python3 dupkg.py gen –template dupkg_dudelta_pull_template –in VALIDATE_OPTION=True TEGRA_A_SRC_OLD=<old bsp> TEGRA_A_SRC_NEW=<new bsp> –out outdir The new delta generation dupkg command line is: python3 dupkg.py gen –in BSDIFF_PATH=./ TEGRA_A_SRC_OLD=<old bsp> TEGRA_A_SRC_NEW=<new bsp> VALIDATE_OPTION=True –template dupkg_dudiff_template –out outdir.				

Drive Update		Backward Compatible	Migration Path	Platform	OS
The Drive Update workflow with DUCC is deprecated in 7.0		No	Orin to Thor Orin to Orin	NSR SR	Linux and QNX
Migration Rationale	A new API is being introduced to provide better control over the update sequence				

continues on next page

Table 11 – continued from previous page

Drive Update		Backward Compatible	Migration Path	Platform	OS
Steps to Migrate	There is no functionality and no need to migrate. Drive OS 6.0 existing DUCC interface are supported through the Drive OS 7.x release. Customers do not need to change if they still use the current DUCC interface. In Drive OS 7.x, NVIDIA Drive® Update will provide the Update Server API (based on HVRTOS) to customers, who should use the Update Server API directly instead of the DUCC interface if it improves software performance. The new API will be provided in the SDK once it is ready and available.				

15.6. Camera Software Changes

Camera Software Changes		Backward Compatible	Migration Path	Platform	OS
The precision of TSC_EDGE_OUT signal changed from 32ns in Orin to 1n in Thor.		Yes because the software uses the different precision internally based on the chip information	Orin to Thor	NSR SR	Linux and QNX
Migration Rationale	Hardware architecture changes in Thor				

continues on next page

Table 12 – continued from previous page

Camera Software Changes		Backward Compatible	Migration Path	Platform	OS
Steps to Migrate	1. Understand the precision of TSC_EDGE_OUT tick to convert TSC tick to the absolute time value (such as nanosecond, seconds, and so on). 2. The change does not result in API changes; no extra steps are required.				

API Functional Behavioral Changes		Backward Compatible	Migration Path	Platform	OS
Struct ImageMetaData added fields: NvSiplISPDeadPixelCorrectionStatsData deadPixelStats, and bool deadPixelStatsValid Struct ImageMetaData updated fields: bool histogramStatsValid, NvSiplISPHistogramStatsData histogramStats, NvSiplISPHistogramStats histogramSettings. All going from 2 -> 3 array elements."		Yes	Orin to Thor	NSR SR	Linux and QNX
Migration Rationale	Hardware architecture changes in Thor				
Steps to Migrate	A new token is defined for Thor instead of the Orin TSEC token				

15.7. CUDA Changes

CUDA Math		Backward Compatible	Migration Path	Platform	OS
No breaking changes to include fp8 Math APIs vs 6.0. Safety scope are planned. Non-Safety scope will also include fp8.		Yes	Orin to Thor Orin to Orin	NSR SR	Linux and QNX
Migration Rationale	6.0 was based on CUDA 11.4, while 7.0 is forked off CUDA 12.x+. Both non-safety and safety CUDA Math receive more functionality (primarily, the cuda_fp8.h header-based types and APIs). APIs remain backward compatible. Safety scope increase is based on internal requests for fp8 support.				
Steps to Migrate	Recompile using a newer toolkit for the applicable target operating system and hardware. In the safety scope, consult with the updated CUDA Math Safety Manual (once available).				

CUDA		Backward Compatible	Migration Path	Platform	OS
The Debian installer name will change to reflect the change from a single unified installer to an installer per target SoC. The installation location for target specific binaries will change. With the move to target specific installers, the location/path of the target binaries will change to include the target name. Current Installer Name cuda-repo-cross-aarch64-ubuntu2004-11-4-local_11.4.28-1_all.deb Future Installer Name for Orin driveos-cuda-orin-nsr-repo-cross-aarch64-ubuntu2404-12-8-local_12.8.99-1_all.deb Future Installer Name for Thor driveos-cuda-thor-nsr-repo-cross-aarch64-ubuntu2404-12-8-local_12.8.99-1_all.deb		No	Orin to Thor Orin to Orin	NSR SR	Linux and QNX
Migration Rationale	6.0 was based on CUDA 11.4, while 7.0 is forked off CUDA 12.x+. Both non-safety and safety CUDA Math receive more functionality (namely, the cuda_fp8.h header-based types and APIs). APIs remain backward compatible. Safety scope increase is based on internal requests for fp8 support.				

continues on next page

Table 15 – continued from previous page

CUDA		Backward Compatible	Migration Path	Platform	OS
Steps to Migrate	Develop CI/CD scripts or other installation scripts keeping this future change in mind. The name and install paths should not be configurable, not hardcoded, to minimize changes to the configuration.				

15.8. Deep Learning Changes

Note

Refer to NVIDIA TENSORRT DOCUMENTATION for in-depth details.

API Changes (Static Changes: Header File)		Backward Compatible	Migration Path	Platform	OS
TensorRT automotive safety public API changed		No	Orin to Thor Orin to Orin	SR	Linux and QNX
Migration Rationale	To provide a better automotive-oriented API				

continues on next page

Table 16 – continued from previous page

API Changes (Static Changes: Header File)		Backward Compatible	Migration Path	Platform	OS
Steps to Migrate	DOS 6 has InferRuntime: an object that is served as an entry point to the TRT library CudaEngine: an that object logically represents the immutable part of a neural network graph(Weights and structure of the network) ExecutionContext: an object that logically represents the mutable part of a neural network inference (Input / output tensor) In DOS 7 we simplify it down to a single object TRTGraph: an object that represents a self-contained network ready for inference. (i.e. it includes the weights, network structure and also the input/output) ErrorHandling: In DOS6 we rely on errorRecorder to communicate API errors. Thor introduces CUDA style errorHandling. In addition to error-Recorder, we provide a return code that indicates the potential issue with the API call.				

15.9. QNX BSP and IO API Changes

15.9.1. Software Development Platform 8.0 Changes

- **Deprecations:** Utilities and APIs are deprecated. For the complete list, refer to the QNX Software Development Platform 8.0: Discontinuation and Deprecation Notice

- **Interrupt Handling:**

  - The flags _NTO_INTR_FLAGS_TRK_MSK, _NTO_INTR_FLAGS_PROCESS are no longer supported for InterruptAttachEvent(). _NTO_INTR_FLAGS_TRK_MSK is the default behavior in SDP 8.0 (for example, kernel always tracks the number of times an interrupt is masked or unmasked)
  - InterruptAttachThread() is more efficient than InterruptAttachEvent() for waiting for interrupts.
  - Replace interruptevent ability with interrupt. interrupt ability is now used for both InterruptAttachEvent() and InterruptAttachThread().
  - InterruptUnmask() and InterruptMask() do not allow interupt ID to be passed as -1.

- **Thread Priorities:** The maximum user thread priority is 253. Priority 255 is used for per core IPI thread, and priority 254 is used for per core timer IST.
- **Signal Handling:** Signal 32 (SIGDOOM) is not allowed to have a signal handler.
- **Custom Callouts:** Custom callouts are implemented like a regular kernel call (unlike with SDP7.1). They are surrounded by InterruptEnable()/Disable(). The call is made in EL1T, not EL1H, and there is no guarantee that the call is made from CPU0.
- **Channel Limits:** There is a new limit for the number of actively open channels. For RLIMIT_CHANNELS_NP, the limit is 100. For applications with over 100 open channels, increase the limit with setrlimit() or with iolauncher while starting the process.

Note

A process linking to a library can create channels. Run pidin -p <pid> channelds` for the number of open channels.

- **Directory File Descriptors:** To open a file descriptor for a directory, use O_DIRECTORY to call open(); otherwise fchdir() will fail like fdopendir().
- **Command-line Utilities:** The *toybox* package combines command-line utilities into a single executable. Utilities available in previous QNX SDP versions are replaced or discontinued with Toybox tools. For information about specific utilities, refer to the *Utilities Reference* in the SDP documentation and the QNX Software Development Platform 8.0: Discontinuation and Deprecation Notice.
- **io-char:** io-char now has a dependency on libsecpol.so and libfsnotify.so.
- **DHCP client:** dhcpcd replaces dhclient.
- **Inline Functions:** Use __attribute__((always_inline)) to ensure inline functions are always inlined.
- **GDB Requirements:** SDP 8.0 requires GLIBC 2.29 or 2.30 (available on Ubuntu 20.04).
- **Tickless Kernel:** SDP 8 remains tickless, but it uses 4 ms periodic ticks per core, since each core has its own timer and no IPIs are generally involved in scheduling, for tasks like round-robin scheduling.
- **Tracelogger:**

  - `_NTO_TRACE_INTENTER` is replaced with `_TRACE_INT_DELIVER`.
  - Use the traceprinter tool for SDP 8 to analyze logs.
- **Thread Abilities:** `xthread`, `threadctl`, and `keydata` abilities are deprecated; the functionality moved under different abilities. For details, refer to Abilities.
- **Semaphore Behavior:** `sem_wait()` is not unblocked by `sem_close()`. Unblocking `sem_wait()` requires an explicit `sem_post()`. The change is still POSIX compliant.
- **MAP_BELOW Semantics:** `MAP_BELOW` starts from the bottom of the address space and goes up, finding the lowest address that matches the requested size.

API changes		Backward Compatible	Migration Path	Platform	OS
QNX SDP8 kernel introduces Clusters A thread can only have a runmask that matches the runmask of a Cluster. The QNX kernel defines `C_all` and `C_cpu-<num>` Clusters by default per the QNX documentation. Any other Cluster needs to be defined prior to booting the QNX kernel. If an application thread needs to have a runmask that does not match the default Clusters, they need to define new Clusters with the desired runmask.		No.	Orin to Thor Orin to Orin	NSR SR	QNX
Migration Rationale	Changes in QNX SDP8 from 6.x to 7.x				
Steps to Migrate	For the steps to migrate, refer to the next section				

Steps to Migrate Example Per the Previous Table

To define a custom Cluster, modify the Device Tree. For example,
to add one Cluster with Core0 and Core1 and another Cluster with Core 2 and Core 3, add the following DT nodes:

```
cpus {
    os-cpu-clusters {
        cpus {
            cpu-map {
                os-cluster_0_1 {
                    cluster-type = "OS-CLUSTER";
                    cluster-name = "cluster_0_1";
```

(continues on next page)

(continued from previous page)

```
                    cpumask = <0x3>;
                    status = "okay";
                };
                os-cluster_2_3 {
                    cluster-type = "OS-CLUSTER";
                    cluster-name = "cluster_2_3";
                    cpumask = <0xc>;
                    status = "okay";
                };
        };
    };
 };
```

IOLaucher has a new option to set thread affinity using a Cluster name: “–cluster <cluster_name>”. The previous approach of setting affinity by specifying the runmask will continue to be supported, but the runmask needs to match an existing cluster.

nvdt_set_thr_attr() added support to set affinity using Cluster name. To do this, in the Device tree, replace “thr-runmask” with “thr-cluster” and specify the cluster name string as the value. The previous approach of using “thr-runmask” will continue to be supported. Only one of “thr-runmask” or “thr-cluster” should be specified for a given thread.

```
nvi2c-bpmp {
    dvms_events_listener {
        thr-name="dvms_events_listener";
        thr-prio=<0x17>;
        thr-prio-range=<&asil_prio_range_other>;
        thr-priv=<0x0>;
        thr-cluster="cluster_name;
    };
};
```

QNX BSP and IO API Changes (Static Changes: Header File)		Backward Compatible	Migration Path	Platform	OS
Adding new enum(s)		Yes	Orin to Thor Orin to Orin	NSR SR	Linux and QNX
Migration Rationale	Due to new feature support, additional enums are defined and an external API is updated to accept the new enum. This feature supports Runtime SC7/ReInit functionality starting with the 6.0.9.3 release. Customers moving from releases prior to 6.0.9.3 to 7.x are affected.				

continues on next page

Table 18 – continued from previous page

QNX BSP and IO API Changes (Static Changes: Header File)		Backward Compatible	Migration Path	Platform	OS
Steps to Migrate	Only the required new functionality can be called. For Init API for NvDVMS Client Library - `nvdvms_init()` The constructor is deprecated in 7.2. Ensure the `nvdvms_init()` API is called for client library init. For De-Init API for NvDVMS Client Library - `nvdvms_deinit()` The destructor is deprecated in 7.2. Ensure `nvdvms_deinit` is called during deinit for client library deinit. For additional information, refer to the nvdvms_ sample application in the DriveOS SDK.				

Chapter 16. Additional Changes

16.1. QNX OS Changes

For QNX OS changes, refer to the following websites:

- QNX Software Development Platform 8.0 Release Notes
- QNX SDP 8.0 Discontinuation and Deprecation Notice

Chapter 17. Deprecations

NvDisplay		Backward Compatible	Migration Path	Platform	OS
Client modeset feature deprecation.		No	Orin to Thor	NSR	Linux
Migration Rationale	Client modeset is not required for automotive use cases. Display is switching to a different architecture and codebase where the client modeset feature is not supported.				
Steps to Migrate	Provide fixed-mode timings in device tree.				

NvDisplay		Backward Compatible	Migration Path	Platform	OS
Dynamic IMP feature deprecation		No	Orin to Thor	NSR	Linux
Migration Rationale	From 7.0.2.0, NvDisplay only supports driver modeset and mode changes are not allowed. NvDisplay will port over to a new codebase where dynamic IMP support is not available. Since mode changes are not required, NvDisplay will not port over unnecessary dynamic IMP changes to the new codebase.				

continues on next page

Table 2 – continued from previous page

NvDisplay		Backward Compatible	Migration Path	Platform	OS
Steps to Migrate	Configure IMP settings statically. For additional information, refer to the NvDisplay tables in *Tools and Utilities Changes* and *Customer-facing Configuration File Changes*.				

NvDisplay		Backward Compatible	Migration Path	Platform	OS
OpenWFD is deprecating Xavier-style display post-flip fence behavior. `WFD_PIPELINE_POSTFENCE_SCANOUT_BEGIN_NVX` attribute macro will be removed from the OpenWFD extensions header wfdext.h.		No	Orin to Thor Orin to Orin	NSR	Linux

continues on next page

Table 3 – continued from previous page

NvDisplay		Backward Compatible	Migration Path	Platform	OS
Migration Rationale	OpenWFD driver introduced this attribute in DRIVE OS 6.0 to emulate Xavier style display post-flip fences to avoid application changes and to give time for applications to change. The attribute is being deprecated now because: Client applications should use the Orin-style post-flip fences to provide a faster and efficient path for tracking flips. Xavier is not supported on DRIVE OS 6.0 or 7.0 and, therefore, the start of DriveOS 7.0 is the appropriate time to deprecate legacy behavior support. This simplifies the OpenWFD driver code.				

continues on next page

Table 3 – continued from previous page

NvDisplay		Backward Compatible	Migration Path	Platform	OS
Steps to Migrate	OpenWFD driver introduced this attribute in DRIVE OS 6.0 to provide backward compatibility to DRIVE OS 5.2 applications that used OpenWFD non-blocking flip extension APIs. In Xavier, the display post-flip fences signal at the beginning of scan out, whereas in Orin the display post-flip fences signal at end of scan out. To avoid application changes and to give time for applications to change, OpenWFD added `WFD_PIPELINE_POSTFENCE_SCANOUT_BEGIN_NVX` attribute to enable applications to emulate the Xavier style display post-fence behavior on Orin. If a client application is using Xavier style post-flip fence behavior, then it should switch to Orin style post-flip fences. If a client application is using Orin style post-flip fence behavior but using `WFD_PIPELINE_POSTFENCE_SCANOUT_BEGIN_NVX` attribute in the code then remove this macro from the code. If a client application is using Orin style post-flip fence behavior and not using `WFD_PIPELINE_POSTFENCE_SCANOUT_BEGIN_NVX` attribute in the code, then there is no impact. `OpenWFD` updates `openwfd_nvsci_sample` app to demonstrate how to use Orin-style post-flip fence behavior.				

continues on next page

Table 3 – continued from previous page

NvDisplay		Backward Compatible	Migration Path	Platform	OS

Graphics		Backward Compatible	Migration Path	Platform	OS
X11/Weston/Wayland/XWa OpenGLES and Vulkan have been moved to the development profile.		Yes	Orin to Thor Orin to Orin	NSR	Linux and QNX
Migration Rationale	This packaging change improves clarity regarding the intended use of these components. These components, previously included in the QNX Standard and Linux Standard releases, are included with DriveOS solely for use in the development of graphics applications and are not intended for production deployment.				
Steps to Migrate	To continue using these components for development, ensure the corresponding packages are installed prior to use. Users are encouraged to migrate to Vulkan SC and OpenWFD; these components are intended for use in both production and development deployments				

NvStreams		Backward Compatible	Migration Path	Platform	OS
The following APIs are deprecated. They are supported in 7.0 for backward compatibility but will be completely removed: `NvSciStreamProducerCreate` `NvSciStreamConsumerCreate` `NvSciStreamIpcSrcCreate` `NvSciStreamIpcDstCreate`		Yes	Orin to Thor Orin to Orin	NSR SR	Linux & QNX
Migration Rationale	The newer version of these APIs are already available from 6.0, such as NvSciStreamProducerCreate2, NvSciStreamConsumerCreate2, NvSciStreamIpcSrcCreate2, and NvSciStreamIpcDstCreate2, which can provide the same functionality as the deprecated APIs and much more. The customers should to move to these newer versions of APIs before the end of support.				
Steps to Migrate	Refer to the following section.				

Steps to Migrate for the Previous Table

Note

The function signatures of newer API versions are in the API Reference; they were released in 6.0.

```
  NvSciStreamProducerCreate:

NvSciError NvSciStreamProducerCreate(

NvSciStreamBlock const pool,

NvSciStreamBlock \*const producer

);

NvSciError NvSciStreamProducerCreate2(
```

(continues on next page)

(continued from previous page)

```
NvSciStreamBlock const pool,

bool const crcValidate,

NvSciStreamBlock \*const producer

);

Passing param crcValidate as false in
NvSciStreamProducerCreate2 is equivalent to
NvSciStreamProducerCreate.

NvSciStreamConsumerCreate:

NvSciError NvSciStreamConsumerCreate(

NvSciStreamBlock const queue,

NvSciStreamBlock \*const consumer

);

NvSciError NvSciStreamConsumerCreate2(

NvSciStreamBlock const queue,

bool const crcValidate,

:NvSciStreamBlock \*const consumer

:);

Passing param crcValidate as false in
NvSciStreamConsumerCreate2 is equivalent to
NvSciStreamConsumerCreate.

NvSciStreamIpcSrcCreate:

NvSciError NvSciStreamIpcSrcCreate(

NvSciIpcEndpoint const ipcEndpoint,

NvSciSyncModule const syncModule,

NvSciBufModule const bufModule,

NvSciStreamBlock \*const ipc

);

NvSciError NvSciStreamIpcSrcCreate2(

NvSciIpcEndpoint const ipcEndpoint,

NvSciSyncModule const syncModule,
```

(continues on next page)

(continued from previous page)

```
NvSciBufModule const bufModule,

NvSciStreamBlock const queue,

NvSciStreamBlock \*const ipc

);

Passing the param queue as 0 (invalid) in
NvSciStreamIpcSrcCreate2 is equivalent to
NvSciStreamIpcSrcCreate.

NvSciStreamIpcDstCreate:

NvSciError NvSciStreamIpcDstCreate(

NvSciIpcEndpoint const ipcEndpoint,

NvSciSyncModule const syncModule,

NvSciBufModule const bufModule,

NvSciStreamBlock \*const ipc

);

NvSciError NvSciStreamIpcDstCreate2(

NvSciIpcEndpoint const ipcEndpoint,

NvSciSyncModule const syncModule,

NvSciBufModule const bufModule,

:NvSciStreamBlock const pool,

NvSciStreamBlock \*const ipc

:);
```

Passing the param pool as `0` (invalid) in `NvSciStreamIpcDstCreate2` is equivalent to `NvSciStreamIpcDstCreate`.

Video Interlace Compositor (VIC) Changes		**Backward Compatible**	**Migration Path**	**Platform**	**OS**

The following APIs will be removed:
`NvMedia2DGetVersion()`
`NvMediaLdcGetVersion()`
`VicDiagnosticsGetVersion()`
		Yes	Orin to Thor
			Orin to Orin
				NSR
				SR
					Linux and QNX

Migration Rationale	There is no use case for these functions.				
Steps to Migrate	Remove these functions from your code.				

NvMediaLDC

`NvMediaLDC` is deprecated. The LDC methods are migrated to NvMedia2D. NvMedia2D can drive all the backends from the one API.

DLA

DriveOS 7.0 will continue to support the NVIDIA Deep Learning Accelerator (DLA) cores on Orin. These DLA cores were removed from Thor to optimize GPU scheduling and increase flexibility.

Chapter 18. Appendix: Additional Resources

For information about NVIDIA products and resources, refer to the NVIDIA Documentation Hub and NVIDIA Developer.

Chapter 19. Document Version History

Document control number: SWE-SWE-004-BSTP

Version	Date	Description of Change
01	July, 2024	Initial release
02	August, 2024	▶ Section 4: ▶ Deleted text: “NsightSystems-linux-nda-2021.2.2.3-9b295cc.deb” and “NVIDIA_Nsight_Graphics_D5Q_NDA_2020.5.20339.deb” ▶ Removed a note regarding NVIDIA CUDA ▶ Section 12. NvDisplay: Replaced all instances of WFD_PIPELINE_COMMIT_NON_BLOCKING_NVX with WFD_PIPELINE_POSTFENCE_SCANOUT_BEGIN_NVX ▶ Added a new section: 16. Support Matrix for Multimedia Entities through 7.0 Releases
3	September, 2024	▶ Added an NvDisplay Serializer SDK Structure Changes table in Section 6 ▶ Added Section 16: NvStreams Changes ▶ Created a CUDA Changes section and added a CUDA table in Section 20 ▶ Added QNX OS Changes information in Section 21

continues on next page

Table 1 – continued from previous page

Version	Date	Description of Change
4	November, 2024	▶ Added PCD SDK Structure Changes in Section 6, SDK Environment ▶ Added Flashing Tools tables in Section 8, Tools and Utilities Changes ▶ Added HDE tables in Section 8, Tools and Utilities Changes ▶ Added an NvDisplay section: Section 19 ▶ Added Drive Update changes in Section 19 ▶ Added Deep Learning API changes in Section 22 ▶ Added VIC Deprecations table in Section 24, Deprecations ▶ Added NvStreams API changes to Section 16, NvStreams Changes ▶ Removed the PVA table in Section 24, Deprecations ▶ Removed a graphics table and updated another in Section 24, Deprecations

continues on next page

Table 1 – continued from previous page

Version	Date	Description of Change
5	February, 2025	▶ In the Packaging and Distribution Changes section, updated installation and package instructions for the following: ▶ Nsight Systems ▶ Nsight Graphics Systems ▶ CUDA ▶ cuDNN ▶ TensorRT ▶ In the Tools and Utilities Changes section, added a table for a new tool that calculates static IMP settings - LAPTSA IMP (laptsa-imp) ▶ In Customer-facing Configuration File Changes section, added the following tables: ▶ NvIpc for endpoint streaming mode buffer limit changes ▶ Device Tree: New config: "static-imp-data" changes ▶ Device Tree: Changes to specify display-timings changes ▶ In the Multimedia Changes section, added a table for changes to the IDE for VP8, VP9, MPEG-2, MPEG-4, VC-1, AV1 and IJPD ▶ In the Multimedia Changes section, added a VIC table for LDC region height alignment changes ▶ Added a table for TSC_EDGE_OUT signal changes in the Camera Software Changes section ▶ In the Deprecations section, added two tables for NvDisplay: ▶ Client modeset feature deprecations ▶ Dynamic IMP feature deprecation

continues on next page

Table 1 – continued from previous page

Version	Date	Description of Change
6	May 15, 2025	▶ In the Target Changes section, added the following tables: ▶ TSEC hardware cryptographic services ▶ Use of KDS hardware ▶ In the Tools and Utilities section, added a table for Botlin GCC 13.2.0 changes ▶ In the Customer-facing Configuration File Changes section, added the following tables: ▶ OESP PCT configuration ▶ Changes to ephemeral (PKCS11 session) asymmetric keys supported ▶ In the Security Services section, added the following tables for API changes: ▶ Encryption parameters ▶ Use of TSEC hardware ▶ PKCS#11Library Known Answer Test ▶ Function call to PKCS#11 Library C_GenerateRandom ▶ PKCS#11 Library C_GetMechanismInfo, ▶ PKCS#11 Library C_SignInit return value for Private keys ▶ In the Video Interlace Compositor (VIC) Changes section, added the following tables: ▶ NvMedia2DAttributes object initialization ▶ for NvMedia 2D default filtering changes
7	July, 2025	▶ Added a section,Software Development Platform 8.0 Changes, in the QNX BSP and IO API Changes topic ▶ Added a new table in the QNX BSP and IO API Changes section for new enum(s) ▶ Updated the table in the QNX BSP and IO API Changes section for nvdvms

Chapter 20. Legal

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.

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.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer’s own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND 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. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

Arm

Arm, AMBA and Arm Powered are registered trademarks of Arm Limited. Cortex, MPCore and Mali are trademarks of Arm Limited. All other brands or product names are the property of their respective holders. □Arm□ is used to represent Arm Holdings plc; its operating company Arm Limited; and the regional subsidiaries Arm Inc.; Arm KK; Arm Korea Limited.; Arm Taiwan Limited; Arm France SAS; Arm Consulting (Shanghai) Co. Ltd.; Arm Germany GmbH; Arm Embedded Technologies Pvt. Ltd.; Arm Norway, AS and Arm Sweden AB.

OpenCL

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

Trademarks

NVIDIA, the NVIDIA logo, NVIDIA DRIVE, CUDA, NVIDIA DRIVE Xavier, NVIDIA DRIVE AGX Orin, NVIDIA DRIVE AGX Pegasus, and TensorRT 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

Copyright