



# NVIDIA DATA LOADING LIBRARY (DALI)

RN-09096-001 \_v01 | May 2019

## Release Notes



# TABLE OF CONTENTS

Chapter 1. DALI Overview.....	1
Chapter 2. DALI Release 0.9.1 Beta.....	2
Chapter 3. DALI Release 0.8.1 Beta.....	4
Chapter 4. DALI Release 0.8 Beta.....	6
Chapter 5. DALI Release 0.7 Beta.....	8
Chapter 6. DALI Release 0.6.1 Beta.....	10
Chapter 7. DALI Release 0.6 Beta.....	12
Chapter 8. DALI Release 0.5 Beta.....	14
Chapter 9. DALI Release 0.4.1 Beta.....	15
Chapter 10. DALI Release 0.4 Beta.....	16
Chapter 11. DALI Release 0.3 Beta.....	18
Chapter 12. DALI Release 0.2 Beta.....	20
Chapter 13. DALI Release 0.1.2 Beta.....	22
Chapter 14. DALI Release 0.1.1 Beta.....	23

# Chapter 1.

## DALI OVERVIEW

NVIDIA® Data Loading Library™ (DALI) accelerates the preprocessing of input data for deep learning applications. By offloading augmentations onto GPUs, DALI addresses performance bottlenecks in today's computer vision deep learning applications that include complex, multi-stage data augmentation steps. With DALI beta release, deep learning researchers can scale training performance on image classification models such as ResNet-50 with MXNet, TensorFlow, and PyTorch across Amazon Web Services P3 8 GPU instances or DGX-1 systems with Volta architecture. Framework developers will have less duplication due to better code reuse and maintainability.

DALI offers both performance and flexibility of accelerating different data pipelines (graphs that can have multiple outputs and inputs), as a single library, that can be easily integrated into different deep learning training and inference applications.

# Chapter 2.

## DALI RELEASE 0.9.1 BETA

The DALI 0.9.1 is a beta release. Hence, for all the features, the functionality and performance will likely be limited.

### Using DALI 0.9.1 Beta

The DALI 0.9.1 can be used with the 19.05 NVIDIA GPU Cloud (NGC) optimized container for MXNet, PyTorch, and TensorFlow. Also, the 19.05 container will be shipped with DALI 0.9.1.

To upgrade to DALI 0.9.1 beta from an older version of DALI, follow the installation instructions in the [DALI Quick Start Guide](#).

Refer to the [DALI Developer Guide](#) for usage details.



The internal DALI C++ API used for operators implementation, and the C++ API that enables using DALI as a library from native code, are not yet officially supported. Hence these APIs may change in the next release without advance notice.

### Key Features and Enhancements

This DALI release includes the following key features and enhancements.

- ▶ Added Optical Flow example.
- ▶ Added full support for .pnm (.ppm/.pgm/.pnm) (in case of nvJPEG, fallback to CPU is done).
- ▶ Added the ability to do the lazy initialization of DALI. It is now possible to separate touching the data during the actual run, and at the time of all the necessary allocations during the build of the pipeline.
- ▶ Reduced the resize operator scratch buffer size by processing the data in mini batches.
- ▶ Added the ability to set the DALI CUDA streams priority.
- ▶ Fixed the sync issue in the DALI Python iterators.

- ▶ Fixed the initialization of CUDA context on the default device during pipeline creation.

### Breaking API Changes

- ▶ Internal Python pipeline API has changed. If any function `_*` was used, then that function should be updated to reflect the new semantic.

### Deprecated Features

- ▶ None.

### Known Issues

- ▶ The new video reader operator requires NVIDIA VIDEO CODEC SDK support in the platform. Prior to 19.01, the NVIDIA GPU Cloud (NGC) optimized containers lack this functionality in the default configuration. To enable the functionality, run the container with the "video" capability enabled, as shown below:

```
-e "NVIDIA_DRIVER_CAPABILITIES=compute,utility,video"
```

- ▶ The video loader operator requires that the key frames occur at a minimum every 10 to 15 frames of the video stream. If the key frames occur at a lesser frequency, then the returned frames may be out of sync.

# Chapter 3.

## DALI RELEASE 0.8.1 BETA

The DALI 0.8.1 is a beta release. Hence, for all the features, the functionality and performance will likely be limited.

### Using DALI 0.8.1 Beta

The DALI 0.8.1 can be used with the 19.04 NVIDIA GPU Cloud (NGC) optimized container for MXNet, PyTorch, and TensorFlow. Also, the 19.05 container will be shipped with DALI 0.9.

To upgrade to DALI 0.8.1 beta from an older version of DALI, follow the installation instructions in the [DALI Quick Start Guide](#).

Refer to the [DALI Developer Guide](#) for usage details.



The internal DALI C++ API used for operators implementation, and the C++ API that enables using DALI as a library from native code, are not yet officially supported. Hence these APIs may change in the next release without advance notice.

### Key Features and Enhancements

This DALI release includes the following key features and enhancements.

- ▶ Fixed nvJPEGDecoder cache when using the new nvJPEG decoupled API.

### Breaking API Changes

- ▶ None.

### Deprecated Features

- ▶ None.

## Known Issues

- ▶ The new video reader operator requires NVIDIA VIDEO CODEC SDK support in the platform. Prior to 19.01, the NVIDIA GPU Cloud (NGC) optimized containers lack this functionality in the default configuration. To enable the functionality, run the container with the "video" capability enabled, as below:

```
-e "NVIDIA_DRIVER_CAPABILITIES=compute,utility,video"
```

- ▶ The video loader operator requires that the key frames occur at a minimum every 10 to 15 frames of the video stream. If the key frames occur at a lesser frequency, then the returned frames may be out of sync.

# Chapter 4.

## DALI RELEASE 0.8 BETA

The DALI 0.8 is a beta release. Hence, for all the features, the functionality and performance will likely be limited.

### Using DALI 0.8 Beta

The DALI 0.8 can be used with the 19.03 NVIDIA GPU Cloud (NGC) optimized container for MXNet, PyTorch, and TensorFlow. Also, the 19.04 container will be shipped with DALI 0.8.

To upgrade to DALI 0.8 beta from an older version of DALI, follow the installation instructions in the [DALI Quick Start Guide](#).

Refer to the [DALI Developer Guide](#) for usage details.



The internal DALI C++ API used for operators implementation, and the C++ API that enables using DALI as a library from native code, are not yet officially supported. Hence these APIs may change in the next release without advance notice.

### Key Features and Enhancements

This DALI release includes the following key features and enhancements.

- ▶ Provides the CUDA 10-based DALI prebuilt binary.
- ▶ Added the customizable and decoupled executor stages. The CPU and GPU prefetch queues can now have different depths.
- ▶ The nvJPEG decoder now supports ROI-based decoding.
- ▶ The nvJPEG decoder now has split variants. The CPU and GPU part can be concurrently executed and independently prefetched.
- ▶ Improved the resize operator to support bilinear and triangular resampling. This improves the quality during aggressive down-scaling.
- ▶ Added the ability to return sparse tensor on CPU for TF DALI op.



- ▶ Fixed bugs in the detection pipeline operators. These bugs might have lead to inferior training accuracy.

### Breaking API Changes

- ▶ None.

### Known Issues

- ▶ The new video reader operator requires NVIDIA VIDEO CODEC SDK support in the platform. Prior to 19.01, the NVIDIA GPU Cloud (NGC) optimized containers lack this functionality in the default configuration. To enable the functionality, run the container with the "video" capability enabled, as below:

```
-e "NVIDIA_DRIVER_CAPABILITIES=compute,utility,video"
```

- ▶ The video loader operator requires that the key frames occur at a minimum every 10 to 15 frames of the video stream. If the key frames occur at a lesser frequency, then the returned frames may be out of sync.

# Chapter 5.

## DALI RELEASE 0.7 BETA

The DALI 0.7 is a beta release. Hence, for all the features, the functionality and performance will likely be limited.

### Using DALI 0.7 Beta

The DALI 0.7 can be used with the 19.03 NVIDIA GPU Cloud (NGC) optimized container for MXNet, PyTorch, and TensorFlow.

To upgrade to DALI 0.7 beta from an older version of DALI, follow the installation instructions in the [DALI Quick Start Guide](#).

Refer to the [DALI Developer Guide](#) for usage details.



The internal DALI C++ API used for operators implementation, and the C++ API that enables using DALI as a library from native code, are not yet officially supported. Hence these APIs may change in the next release without advance notice.

### Key Features and Enhancements

This DALI release includes the following key features and enhancements.

- ▶ Added new operators:
  - ▶ HostDecoderRandomCrop
  - ▶ Element Extract for sequences
- ▶ Added the API stub for incoming Optical Flow video operator.
- ▶ Introduced Python 3.7 based **whl** build.
- ▶ Added Docker-based "one click" build script.
- ▶ Added the ability to fine-tune the internal DALI buffers using per-operator presize hints to stage the output queues.
- ▶ Added **step**, **stride** and **shuffling** in **SequenceReader**.
- ▶ Added a new, internal test API.

- ▶ Added the ability to custom handle CPU affinity using the DALI\_AFFINITY\_MASK environment variable.

### Breaking API Changes

- ▶ None.

### Deprecated Features

- ▶ Dropped the Python 3.4 based **wh1** build.

### Known Issues

- ▶ The new video reader operator requires NVIDIA VIDEO CODEC SDK support in the platform. The NVIDIA GPU Cloud (NGC) optimized containers lack this functionality in the default configuration prior to the 19.01 version. To enable the functionality, run the container with the "video" capability enabled, as below:

```
-e "NVIDIA_DRIVER_CAPABILITIES=compute,utility,video"
```

- ▶ There is no clear distinction in the documentation between the operators supporting video sequences and the operators supporting images.

# Chapter 6.

## DALI RELEASE 0.6.1 BETA

The DALI 0.6.1 is a beta release. Hence, for all the features, the functionality and performance will likely be limited.

### Key Features and Enhancements

This DALI release includes the following key features and enhancements.

- ▶ Added the DALI TensorFlow plugin to improve out-of-the-box forward compatibility.
- ▶ Added a new package **nvidia-dali-tf-plugin** that allows user to compile DALI TensorFlow plugin using the locally installed version.
- ▶ Added the below examples:
  - ▶ VideoReader
  - ▶ PyTorch and MXNet examples with various readers
- ▶ Added a new operator variant **gpu box encoder**.
- ▶ Added a new JSON parser—now COCO reader start up time is significantly faster (~10x).
- ▶ Added the ability to automatically reset MXNet and PyTorch Python iterators.
- ▶ Enhanced the Video reader to link dynamically with the CUDA Driver API and **nvcuvid**.
- ▶ Added the ability to deliver exactly 1 epoch from **DALIGenericIterator** (PyTorch and MXNet).

### Breaking API Changes

- ▶ PyTorch iterator returns the exact number of samples per epoch, so the final batch could be smaller if epoch size is not divisible by the batch size. To keep the old behaviour when data is wrapped up, use **stop\_at\_epoch** argument.

## Using DALI 0.6.1 Beta

The 19.01 NVIDIA GPU Cloud (NGC) optimized container for MXNet, PyTorch, and TensorFlow includes an older version of DALI.

To upgrade to DALI 0.6.1 beta, follow the installation instructions in the [DALI Quick Start Guide](#).

Refer to the [DALI Developer Guide](#) for usage details.

## Known Issues

- ▶ The New Video reader operator requires NVIDIA VIDEO CODEC SDK support in the platform. The NVIDIA GPU Cloud (NGC) optimized containers lack this functionality in the default configuration prior to the 19.01 version, resulting in problem with loading `libnvcuvid.so`. To enable it, run the container with the 'video' capability enabled, as below:

```
-e "NVIDIA_DRIVER_CAPABILITIES=compute,utility,video"
```

# Chapter 7.

## DALI RELEASE 0.6 BETA

The DALI 0.6 is a beta release. Hence, for all the features, the functionality and performance will likely be limited.

### Key Features and Enhancements

This DALI release includes the following key features and enhancements.

- ▶ Added the CPU variant of DALI TensorFlow operator. Now the DALI pipeline can be scheduled on the CPU also.
- ▶ Added the TensorFlow operator API to allow defining an arbitrary number of outputs and their types.
- ▶ Added full DALI detection pipeline sample for SSD.
- ▶ Added clear mapping between operators and the supported device types in the API documentation.
- ▶ Created initial support for the video-based pipeline - [NVVL](#) functionality is available as `VideoReader`.
- ▶ Added the below new operators:
  - ▶ `BoxEncoder`
  - ▶ `VideoReader`
  - ▶ `SequenceCrop`
  - ▶ `SequenceReader`
- ▶ Added new variants of the below existing operators:
  - ▶ For the GPU:
    - ▶ `RandomBBoxCrop`
    - ▶ `BbFlip`
    - ▶ `Slice`
  - ▶ For the CPU:
    - ▶ `CropMirrorNormalize`, and

▶ **BBoxPaste**

### Breaking API Changes

- ▶ DALI TensorFlow operator has a new API - check the examples for the reference.
- ▶ PyTorch and MXNet python iterators API have changed - check the examples for the reference.

### Using DALI 0.6 Beta

The 19.01 NVIDIA GPU Cloud (NGC) optimized container for MXNet, PyTorch, and TensorFlow includes an older version of DALI. To upgrade to DALI 0.6 beta, follow the installation instructions in the [DALI Quick Start Guide](#).

Refer to the [DALI Developer Guide](#) for usage details.

### Known Issues

- ▶ The New Video reader operator requires NVIDIA VIDEO CODEC SDK support in the platform. The NVIDIA GPU Cloud (NGC) optimized containers lack this functionality in the default configuration prior to the 19.01 version, resulting in problem with loading `libnvcuvid.so`. To enable it, run the container with the 'video' capability enabled, as below:

```
-e "NVIDIA_DRIVER_CAPABILITIES=compute,utility,video"
```

# Chapter 8.

## DALI RELEASE 0.5 BETA

This DALI 0.5 release is a beta release.

### Key Features and Enhancements

This DALI release includes the following key features and enhancements.

- ▶ Fixed rare hangs during DALI pipeline destruction.
- ▶ Added ResNet50 example with TensorFlow.
- ▶ Added more verbose error messages to HostDecoder and nvJpegDecoder.
- ▶ Added pipelined execution completion callback setter.
- ▶ Added support for Tiff images.

### Breaking API Changes

- ▶ Random seed type changed from INT to INT64, therefore, serialized pipelines from versions prior to 0.5 are not compatible with the current DALI version.

### Using DALI 0.5 Beta

The 18.11 NVIDIA GPU Cloud (NGC) optimized container for MXNet, PyTorch, and TensorFlow, includes an older version of DALI. To upgrade to DALI 0.5 beta, follow the installation instructions in the [DALI Quick Start Guide](#).

Refer to the [DALI Developer Guide](#) for usage details.

### Known Issues

- ▶ This is a beta release. Hence, for all the features, the functionality and performance will likely be limited.



# Chapter 9.

## DALI RELEASE 0.4.1 BETA

This DALI 0.4.1 release is a beta release.

### Key Features and Enhancements

This DALI release includes the following key features and enhancements.

- ▶ Added TensorFlow 1.11 and 1.12 compatibility.
- ▶ Adjusted PyTorch example to use new nvJPEG API.
- ▶ Reduced DALI memory footprint.
- ▶ Fixed lack of ability to customize pipeline processing length.

### Using DALI 0.4.1 Beta

The 18.10 NVIDIA GPU Cloud (NGC) optimized container for MXNet, PyTorch, and TensorFlow, includes an older version of DALI. To upgrade to DALI 0.4.1 beta, follow the installation instructions in the [DALI Quick Start Guide](#).

Refer to the [DALI Developer Guide](#) for usage details.

### Known Issues

- ▶ The DALI integrated ResNet-50 samples in the 18.10 NGC TensorFlow and PyTorch containers may result in lower than expected performance results. We are working to address the issue in the next release.
- ▶ This is a beta release. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.

# Chapter 10.

## DALI RELEASE 0.4 BETA

This DALI 0.4 release is a beta release.

### Key Features and Enhancements

This DALI release includes the following key features and enhancements.

- ▶ Added basic operators for detection (SSD), such as COCO dataset reader (**COCOReader**), random crop operator with bounding boxes (**SSDRandomCrop**), and flip operator for bounding boxes (**BbFlip**).
- ▶ Added CPU versions of **Crop/CropCastPermute** operators.
- ▶ Added a random **Paste** operator.
- ▶ Upgraded OpenCV to 3.4.3.
- ▶ Upgraded MXNet to 1.3.0.
- ▶ Fixed parsing of JPEG headers by the Host Decoder known issue. The Host Decoder now handles all images from the ImageNet dataset.
- ▶ Added fine grained control over output buffers in the pipeline
- ▶ Updated to nvJPEG 0.2.0

### Breaking API Changes

The pipeline constructor signature has changed; an additional argument (**prefetch\_queue\_depth**) was added to allow defining depth of the prefetch queue at runtime.

## Using DALI 0.4 Beta

The 18.09 NVIDIA GPU Cloud (NGC) optimized container for MXNet, PyTorch, and TensorFlow, includes an older version of DALI. To upgrade to DALI 0.4 beta, follow the installation instructions in the [DALI Quick Start Guide](#).

Refer to the [DALI Developer Guide](#) for usage details.

## Deprecated Features

- ▶ DALI 0.4 is not compatible with TensorFlow 1.11. This will be addressed in the next release.

## Known Issues

- ▶ The DALI integrated ResNet-50 samples in the 18.10 NGC TensorFlow and PyTorch containers may result in lower than expected performance results. We are working to address the issue in the next release.
- ▶ This is a beta release. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.

# Chapter 11.

## DALI RELEASE 0.3 BETA

This DALI 0.3 release is a beta release.

### Key Features and Enhancements

This DALI release includes the following key features and enhancements.

- ▶ Updated PyTorch ResNet-50 example to obtain expected accuracy (Top1 76%).
- ▶ Introduced CPU variant of resize operator and added stand-alone flip operator.
- ▶ Added support for DALI to work with float16 data passed from Python.
- ▶ Added fallback to CPU for BMP images.
- ▶ Fixed training accuracy with TFRecord reader

### Using DALI 0.3 Beta

The 18.08 NVIDIA GPU Cloud (NGC) optimized container for MXNet, PyTorch, and TensorFlow, includes an older version of DALI. To upgrade to DALI 0.2 beta, follow the installation instructions in the [DALI Quick Start Guide](#).

Refer to the [DALI Developer Guide](#) for usage details.

### Known Issues

- ▶ HosDecoder cannot handle all jpeg files from ImageNet dataset. This will be fixed in the next release.
- ▶ On file systems where the directory entries are not stored in any order, the File Reader may assign different labels to the training and validation folders with the same name. This will be fixed in the next release.

- ▶ The DALI integrated ResNet-50 samples in the 18.09 NGC TensorFlow and PyTorch containers may result in lower than expected performance results. We are working to address the issue in the next release.
- ▶ This is a beta release. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.

# Chapter 12.

## DALI RELEASE 0.2 BETA

This DALI 0.2 release is a beta release.

### Key Features and Enhancements

This DALI release includes the following key features and enhancements.

- ▶ Added Sphinx based documentation that is in sync with the code on GitHub. For more information, see [DALI Master Branch User Guide](#).
- ▶ Build system has been refined, common errors have meaningful messages, improved localization and version detection of key dependency packages, such as, nvJPEG, libturbo-jpeg, and LMDB.
- ▶ Added **Unfused Crop** and **CropCastPermute** operators.
- ▶ Added improvements for TensorFlow plugin (polymorphism and shape argument for the output).
- ▶ Expanded examples of TensorFlow working with different readers, such as, MXNetReader, FileReader, and TFRecordReader.
- ▶ Updated nvJPEG to 0.1.4
- ▶ Added fallback to host decoder when image is not JPEG but PNG instead. For example, `n02105855_2933.JPEG` from ImageNet.

### Breaking API Changes

- ▶ The API for the **Resize** operator changed to match other similar operators like **ResizeCropMirror**.
- ▶ The API for the TensorFlow plugin changed to allow specifying the whole shape of the tensor instead of **N**, **H**, and **W** separately; which enables handling both **NCHW** and **NHWC** outputs.

- ▶ The type of labels produced by the TensorFlow plugin have changed. In DALI version 0.1.2, it was always `tf.float32`. In this release, a new optional parameter called `label_type` is introduced to the TensorFlow plugin to control the type of label. The default value for `label_type` is `tf.int64` to better align with the label type in TFRecord.

### Using DALI 0.2 Beta

The 18.08 NVIDIA GPU Cloud (NGC) optimized container for MXNet, PyTorch, and TensorFlow, includes an older version of DALI. To upgrade to DALI 0.2 beta, follow the installation instructions in the [DALI Quick Start Guide](#).

Refer to the [DALI Developer Guide](#) for usage details.

### Known Issues

- ▶ This is a beta release, therefore, not all functionality is fully supported and working. This beta release is meant for testing and research.
- ▶ The DALI integrated ResNet-50 samples in the 18.08 NGC TensorFlow and PyTorch containers have lower than expected accuracy and performance results. We are working to address the issue in the next release.
- ▶ This is a beta release. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.

# Chapter 13.

## DALI RELEASE 0.1.2 BETA

This DALI 0.1.2 release is a beta release.

### Key Features and Enhancements

This DALI release includes the following key features and enhancements.

- ▶ Fixed compatibility with TensorFlow 1.9.
- ▶ Updated to nvJPEG v0.1.2 to fix batched decoding when a batch contains both gray-scale and color images.
- ▶ Added Tensorflow 1.7 support.
- ▶ Improved overlap support when using DALI with multi-GPU in MXNet and PyTorch.

### Using DALI 0.1.2 Beta

The 18.07 optimized container for MXNet, PyTorch, and TensorFlow, includes an older version of DALI. To upgrade to DALI 0.2 beta, follow the installation instructions in the [DALI Quick Start Guide](#).

Refer to the [DALI Developer Guide](#) for usage details.

### Known Issues

- ▶ This is a beta release, therefore, not all functionality is fully supported and working. This beta release is meant for testing and research.
- ▶ This is a beta release. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.



# Chapter 14.

## DALI RELEASE 0.1.1 BETA

This is the first release of DALI. This DALI 0.1.1 release is a beta release.

### Key Features and Enhancements

This DALI release includes the following key features and enhancements.

#### Performance

On dense GPU systems, deep learning applications can be significantly bottlenecked on the CPU, limiting the overall performance and scalability of training and inference tasks. DALI enables offloading key deep learning augmentation steps on to GPUs, alleviating CPU bottleneck on the deep learning preprocessing pipelines. This results in out-of-box performance of overall training workflow and efficient utilization of multi-GPU resources on the system.

#### Drop-in Integration

DALI comes with built-in plugins for key frameworks such as MXNet, TensorFlow, and PyTorch. This enables automatic integration with frameworks so that researchers and developers can get up and running with DALI easily and quickly.

#### Flexibility

DALI supports multiple input data formats that are commonly used in computer vision deep learning applications, for example, JPEG images, raw formats, Lightning Memory-Mapped Database (LMDB), RecordIO and TFRecord. The flexibility of input data formats allows portability of training workflows across different frameworks and models, and helps to avoid intermediate data conversion steps. DALI enables better code reuse and maintainability with optimized building blocks and support for different data formats.

### Using DALI 0.1.1 Beta

Ensure you are familiar with the following notes when using this release.

- ▶ To install DALI, see the [DALI Quick Start Guide](#).



If you are using the 18.07 NGC optimized container for MXNet, PyTorch, or TensorFlow, you do not need to reinstall DALI. DALI now comes included in the container. Instead, start with the [Getting Started Tutorial](#).

- ▶ To interact with the code via GitHub, see the [Getting Started Tutorial](#).
- ▶ To learn how to define, build, and run a DALI pipeline, see the [DALI Developer Guide](#).

### Known Issues

- ▶ This is a beta release, therefore, not all functionality is fully supported and working. This beta release is meant for testing and research.
- ▶ This is a beta release. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.

## Notice

THE INFORMATION IN THIS GUIDE AND ALL OTHER INFORMATION CONTAINED IN NVIDIA DOCUMENTATION REFERENCED IN THIS GUIDE IS PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE INFORMATION FOR THE PRODUCT, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the product described in this guide shall be limited in accordance with the NVIDIA terms and conditions of sale for the product.

THE NVIDIA PRODUCT DESCRIBED IN THIS GUIDE IS NOT FAULT TOLERANT AND IS NOT DESIGNED, MANUFACTURED OR INTENDED FOR USE IN CONNECTION WITH THE DESIGN, CONSTRUCTION, MAINTENANCE, AND/OR OPERATION OF ANY SYSTEM WHERE THE USE OR A FAILURE OF SUCH SYSTEM COULD RESULT IN A SITUATION THAT THREATENS THE SAFETY OF HUMAN LIFE OR SEVERE PHYSICAL HARM OR PROPERTY DAMAGE (INCLUDING, FOR EXAMPLE, USE IN CONNECTION WITH ANY NUCLEAR, AVIONICS, LIFE SUPPORT OR OTHER LIFE CRITICAL APPLICATION). NVIDIA EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR SUCH HIGH RISK USES. NVIDIA SHALL NOT BE LIABLE TO CUSTOMER OR ANY THIRD PARTY, IN WHOLE OR IN PART, FOR ANY CLAIMS OR DAMAGES ARISING FROM SUCH HIGH RISK USES.

NVIDIA makes no representation or warranty that the product described in this guide will be suitable for any specified use without further testing or modification. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to ensure the product is suitable and fit for the application planned by customer and to do 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 guide. NVIDIA does not accept any 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 guide, or (ii) customer product designs.

Other than the right for customer to use the information in this guide with the product, no other license, either expressed or implied, is hereby granted by NVIDIA under this guide. Reproduction of information in this guide is permissible only if reproduction is approved by NVIDIA in writing, is reproduced without alteration, and is accompanied by all associated conditions, limitations, and notices.

## Trademarks

NVIDIA, the NVIDIA logo, and cuBLAS, CUDA, cuDNN, cuFFT, cuSPARSE, DALI, DIGITS, DGX, DGX-1, Jetson, Kepler, NVIDIA Maxwell, NCCL, NVLink, Pascal, Tegra, TensorRT, and Tesla are trademarks and/or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

## Copyright

© 2019 NVIDIA Corporation. All rights reserved.