



# NVIDIA DATA LOADING LIBRARY (DALI)

RN-09096-001 \_v01 | September 2018

## Release Notes



# TABLE OF CONTENTS

Chapter 1. DALI Overview.....	1
Chapter 2. DALI Release 0.3 Beta.....	2
Chapter 3. DALI Release 0.2 Beta.....	3
Chapter 4. DALI Release 0.1.2 Beta.....	5
Chapter 5. DALI Release 0.1.1 Beta.....	6

# Chapter 1.

## DALI OVERVIEW

NVIDIA® Data Loading Library™ (DALI) is a data loading library that 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 0.1 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 0.3 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.

### 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.

# Chapter 3.

## 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.

# Chapter 4.

## 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.

# Chapter 5.

## 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.

## 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

© 2018 NVIDIA Corporation. All rights reserved.