NVIDIA TensorRT

API Reference | NVIDIA Docs

PR-08724-001_v8.2.2 | December 2021

Table of Contents

List of Tables

Chapter 1. Added, Deprecated, And Removed APIs

1.1. API Changes For TensorRT 8.2.0 EA

The following tables show which APIs were added, deprecated, and removed for the NVIDIA® TensorRT™ 8.2.0 EA release.

C++ changes

Table 1. New C++ APIs

New C++ APIs
IAssertionLayer
IConditionLayer
IEinsumLayer
IScatterLayer

Python changes

Table 2. New Python APIs

New Python APIs
IAssertionLayer
IConditionLayer
IEinsumLayer
IScatterLayer

1.2. API Changes For TensorRT 8.0.1

The following tables show which APIs were added, deprecated, and removed for the TensorRT 8.0.1 release.

C++ changes

Table 3. New C++ APIs

New C++ APIs
class IDequantizeLayer
class IQuantizeLayer
class ITimingCache
IBuilder::buildSerializedNetwork()
IBuilderConfig::getTimingCache()
IBuilderConfig::setTimingCache()
IGpuAllocator::reallocate()
INetworkDefinition::addDequantize()
INetworkDefinition::addQuantize()
INetworkDefinition::setWeightsName()
IPluginRegistry::deregisterCreator()
IRefitter::getMissingWeights()
IRefitter::getAllWeights()
IRefitter::setNamedWeights()
IResizeLayer::getCoordinateTransformation()
IResizeLayer::getNearestRounding()
IResizeLayer::getSelectorForSinglePixel()
IResizeLayer::setCoordinateTransformation()
IResizeLayer::setNearestRounding()
IResizeLayer::setSelectorForSinglePixel()
IScaleLayer::setChannelAxis()
enum ResizeCoordinateTransformation
enum ResizeMode
BuilderFlag::kSPARSE_WEIGHTS

New C++ APIs
TacticSource::kCUDNN
TensorFormat::kDLA_HWC4
TensorFormat::kDLA_LINEAR
TensorFormat::kHWC16

Table 4. Removed C++ APIs

Removed C++ APIs
Core Library
DimensionType
Dims::Type
class DimsCHW
class DimsNCHW
class IOutputDimensionFormula
class IPlugin
class IPluginFactory
class IPluginLayer
class IRNNLayer
IBuilder::getEngineCapability()
IBuilder::allowGPUFallback()
IBuilder::buildCudaEngine()
IBuilder::canRunOnDLA()
IBuilder::createNetwork()
IBuilder::getAverageFindIterations()
IBuilder::getDebugSync()
IBuilder::getDefaultDeviceType()
IBuilder::getDeviceType()
IBuilder::getDLACore()
IBuilder::getFp16Mode()
IBuilder::getHalf2Mode()
IBuilder::getInt8Mode()
IBuilder::getMaxWorkspaceSize()
IBuilder::getMinFindIterations()

Removed C++ APIs
`IBuilder::getRefittable()`
`IBuilder::getStrictTypeConstraints()`
`IBuilder::isDeviceTypeSet()`
`IBuilder::reset()`
`IBuilder::resetDeviceType()`
`IBuilder::setAverageFindIterations()`
`IBuilder::setDebugSync()`
`IBuilder::setDefaultDeviceType()`
`IBuilder::setDeviceType()`
`IBuilder::setDLACore()`
`IBuilder::setEngineCapability()`
`IBuilder::setFp16Mode()`
`IBuilder::setHalf2Mode()`
`IBuilder::setInt8Calibrator()`
`IBuilder::setInt8Mode()`
`IBuilder::setMaxWorkspaceSize()`
`IBuilder::setMinFindIterations()`
`IBuilder::setRefittable()`
`IBuilder::setStrictTypeConstraints()`
`ICudaEngine::getWorkspaceSize()`
`IMatrixMultiplyLayer::getTranspose()`
`IMatrixMultiplyLayer::setTranspose()`
`INetworkDefinition::addMatrixMultiply()`
`INetworkDefinition::addPlugin()`
`INetworkDefinition::addPluginExt()`
`INetworkDefinition::addRNN()`
`INetworkDefinition::getConvolutionOutputDimensionsFormula()`
`INetworkDefinition::getDeconvolutionOutputDimensionsFormula()`
`INetworkDefinition::getPoolingOutputDimensionsFormula()`
`INetworkDefinition::setConvolutionOutputDimensionsFormula()`
`INetworkDefinition::setDeconvolutionOutputDimensionsFormula()`
`INetworkDefinition::setPoolingOutputDimensionsFormula()`

Removed C++ APIs
`ITensor::getDynamicRange()`
`TensorFormat::kNHWC8`
`TensorFormat::NCHW`
`TensorFormat::kNC2HW2`
Caffe Parser
`class IPluginFactory`
`class IPluginFactoryExt`
`setPluginFactory()`
`setPluginFactoryExt()`
UFF Parser
`class IPluginFactory`
`class IPluginFactoryExt`
`setPluginFactory()`
`setPluginFactoryExt()`

Table 5. Removed Plugins

Removed Plugins
`class INvPlugin`
`createLReLUPlugin()`
`createClipPlugin()`
`PluginType`
`struct SoftmaxTree`

For plugins based on `IPluginV2DynamicExt` and `IPluginV2IOExt`, certain methods with legacy function signatures (derived from `IPluginV2` and `IPluginV2Ext` base classes) which were deprecated and marked for removal in TensorRT 8.0 will no longer be available. Plugins using these interface methods must stop using them or implement the versions with updated signatures, as applicable.

Table 6. Unsupported plugin methods removed in TensorRT 8.0

Removed Plugins
`IPluginV2DynamicExt::canBroadcastInputAcrossBatch()`
`IPluginV2DynamicExt::isOutputBroadcastAcrossBatch()`

Removed Plugins
`IPluginV2DynamicExt::getTensorRTVersion()`
`IPluginV2IOExt::configureWithFormat()`
`IPluginV2IOExt::getTensorRTVersion()`

Table 7. Updated versions for supported plugin methods

Removed Plugin	Replaced with
	`IPluginV2DynamicExt::configurePlugin()`
	`IPluginV2DynamicExt::enqueue()`
	`IPluginV2DynamicExt::getOutputDimensions()`
	`IPluginV2DynamicExt::getWorkspaceSize()`
	`IPluginV2IOExt::configurePlugin()`
`IPluginV2DynamicExt::supportsFormat()`	`IPluginV2DynamicExt::supportsFormatCombination()`
`IPluginV2IOExt::supportsFormat()`	`IPluginV2IOExt::supportsFormatCombination()`

Python changes

Table 8. New Python APIs

New Python APIs
`class IDequantizeLayer`
`class IQuantizeLayer`
`class ITimingCache`
`Builder.build_serialized_network()`
`IBuilderConfig.get_timing_cache()`
`IBuilderConfig.set_timing_cache()`
`IGpuAllocator.reallocate()`
`INetworkDefinition.add_dequantize()`
`INetworkDefinition.add_quantize()`
`INetworkDefinition.set_weights_name()`
`IPluginRegistry.deregister_creator()`
`Refitter.get_all_weights()`
`Refitter.get_missing_weights()`
`Refitter::set_named_weights()`

New Python APIs
IResizeLayer.coordinate_transformation
IResizeLayer.nearest_rounding
IResizeLayer.selector_for_single_pixel
IScaleLayer.channel_axis
enum ResizeCoordinateTransformationDoc
enum ResizeMode
BuilderFlag.SPARSE_WEIGHTS
TacticSource.CUDNN
TensorFormat.DLA_HWC4
TensorFormat.DLA_LINEAR
TensorFormat.HWC16

Table 9. Removed Python APIs

Removed Python APIs
Core Library
class DimsCHW
class DimsNCHW
class IPlugin
class IPluginFactory
class IPluginLayer
class IRNNLayer
Builder.build_cuda_engine()
Builder.average_find_iterations
Builder.debug_sync
Builder.fp16_mode
IBuilder.int8_mode
Builder.max_workspace_size
Builder.min_find_iterations
Builder.refittable
Builder.strict_type_constraints
ICudaEngine.max_workspace_size
IMatrixMultiplyLayer.transpose0

Removed Python APIs
`INetworkDefinition.add_matrix_multiply_deprecated()`
`INetworkDefinition.add_plugin()`
`INetworkDefinition.add_plugin_ext()`
`INetworkDefinition.add_rnn()`
`INetworkDefinition.convolution_output_dimensions_formula`
`INetworkDefinition.deconvolution_output_dimensions_formula`
`INetworkDefinition.pooling_output_dimensions_formula`
`ITensor.get_dynamic_range()`
`Dims.get_type()`
`TensorFormat.HWC8`
`TensorFormat.NCHW`
`TensorFormat.NCHW2`
Caffe Parser
`class IPluginFactory`
`class IPluginFactoryExt`
`CaffeParser.plugin_factory`
`CaffeParser.plugin_factory_ext`
UFF Parser
`class IPluginFactory`
`class IPluginFactoryExt`
`UffParser.plugin_factory`
`UffParser.plugin_factory_ext`

Deprecated

For our deprecation policy, refer to the TensorRT Deprecation Policy section in the *TensorRT Developer Guide*.

Table 10. Deprecated APIs

Deprecated APIs	Replaced with

`nvinfer1::IResizeLayer::setAlignCorners`	`nvinfer1::IResizeLayer::setCoordinateTransformation`
	`nvinfer1::IResizeLayer::setSelectorForSinglePixel`
	`nvinfer1::IResizeLayer::setNearestRounding`

`nvinfer1::IResizeLayer::getAlignCorners`	

Chapter 2. C++ API

The NVIDIA® TensorRT™ C++ API allows developers to import, calibrate, generate and deploy networks using C++. Networks can be imported directly from ONNX. They may also be created programmatically by instantiating individual layers and setting parameters and weights directly.

Within the core C++ API in `NvInfer.h`, the following APIs are included:

- Builder API
- Execution API
- Network Definition API
- ONNX Parser API
- Plugin API

To view this API, see TensorRT C++ API.

For more information about the C++ API, including sample code, see *NVIDIA TensorRT Developer Guide*.

Chapter 3. Python API

The NVIDIA® TensorRT™ Python API enables developers in Python based development environments and those looking to experiment with TensorRT to easily parse models (for example, from ONNX) and generate and run PLAN files.

To view this API, see TensorRT Python API.

For more information about the Python API, including sample code, see TensorRT Developer Guide.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com