



# cuDLA API

## API Reference Manual

# Table of Contents

<b>Chapter 1. Data Structures.....</b>	<b>1</b>
<b>cudaDevAttribute.....</b>	<b>1</b>
deviceVersion.....	1
unifiedAddressingSupported.....	1
<b>cudaExternalMemoryHandleDesc_t.....</b>	<b>1</b>
extBufObject.....	2
size.....	2
<b>cudaExternalSemaphoreHandleDesc_t.....</b>	<b>2</b>
extSyncObject.....	2
<b>CudlaFence.....</b>	<b>2</b>
fence.....	2
type.....	2
<b>cudaModuleAttribute.....</b>	<b>2</b>
inputTensorDesc.....	3
numInputTensors.....	3
numOutputTensors.....	3
outputTensorDesc.....	3
<b>cudaModuleTensorDescriptor.....</b>	<b>3</b>
<b>cudaSignalEvents.....</b>	<b>3</b>
devPtrs.....	3
eofFences.....	3
numEvents.....	3
<b>cudaTask.....</b>	<b>4</b>
inputTensor.....	4
moduleHandle.....	4
numInputTensors.....	4
numOutputTensors.....	4
outputTensor.....	4
signalEvents.....	4
waitEvents.....	4
<b>cudaWaitEvents.....</b>	<b>4</b>
numEvents.....	4
preFences.....	5
<b>Chapter 2. Data Fields.....</b>	<b>6</b>

---

# Chapter 1. Data Structures

Here are the data structures with brief descriptions:

[cudaDevAttribute](#)

[cudaExternalMemoryHandleDesc](#)

[cudaExternalSemaphoreHandleDesc](#)

[CudaFence](#)

[cudaModuleAttribute](#)

[cudaModuleTensorDescriptor](#)

[cudaSignalEvents](#)

[cudaTask](#)

[cudaWaitEvents](#)

## 1.1. `cudaDevAttribute` Union Reference

Device attribute.

`uint32_t cudaDevAttribute::deviceVersion`

DLA device version. Xavier has 1.0 and Orin has 2.0.

`uint8_t`

`cudaDevAttribute::unifiedAddressingSupported`

Returns 0 if unified addressing is not supported.

## 1.2. `cudaExternalMemoryHandleDesc_t` Struct Reference

External memory handle descriptor.

```
const void
*cudlaExternalMemoryHandleDesc_t::extBufObject
```

A handle representing an external memory object.

```
unsigned long long
cudlaExternalMemoryHandleDesc_t::size
```

Size of the memory allocation

### 1.3. cudlaExternalSemaphoreHandleDesc\_t Struct Reference

External semaphore handle descriptor.

```
const void
*cudlaExternalSemaphoreHandleDesc_t::extSyncObject
```

A handle representing an external synchronization object.

### 1.4. CudlaFence Struct Reference

Fence description.

```
void *CudlaFence::fence
```

Fence.

```
cudlaFenceType CudlaFence::type
```

Fence type.

### 1.5. cudlaModuleAttribute Union Reference

Module attribute.

`cudaModuleTensorDescriptor`  
`*cudaModuleAttribute::inputTensorDesc`

Returns an array of input tensor descriptors.

`uint32_t cudaModuleAttribute::numInputTensors`

Returns the number of input tensors.

`uint32_t cudaModuleAttribute::numOutputTensors`

Returns the number of output tensors.

`cudaModuleTensorDescriptor`  
`*cudaModuleAttribute::outputTensorDesc`

Returns an array of output tensor descriptors.

## 1.6. `cudaModuleTensorDescriptor` Struct Reference

Tensor descriptor.

## 1.7. `cudaSignalEvents` Struct Reference

Signal events for `cudaSubmitTask`

`const **cudaSignalEvents::devPtrs`

Array of registered synchronization objects (via `cudaImportExternalSemaphore`).

`CudaFence *cudaSignalEvents::eofFences`

Array of fences pointers for all the signal events corresponding to the synchronization objects.

`uint32_t cudaSignalEvents::numEvents`

Total number of signal events.

## 1.8. cudlaTask Struct Reference

Structure of Task.

`const **cudlaTask::inputTensor`

Array of input tensors.

`cudlaModule cudlaTask::moduleHandle`

cuDLA module handle.

`uint32_t cudlaTask::numInputTensors`

Number of input tensors.

`uint32_t cudlaTask::numOutputTensors`

Number of output tensors.

`const **cudlaTask::outputTensor`

Array of output tensors.

`cudlaSignalEvents *cudlaTask::signalEvents`

Signal events.

`const cudlaWaitEvents *cudlaTask::waitEvents`

Wait events.

## 1.9. cudlaWaitEvents Struct Reference

Wait events for cudlaSubmitTask.

`uint32_t cudlaWaitEvents::numEvents`

Total number of wait events.

```
const CudlaFence *cudlaWaitEvents::preFences
```

Array of fence pointers for all the wait events.

---

# Chapter 2. Data Fields

Here is a list of all documented struct and union fields with links to the struct/union documentation for each field:

**deviceVersion**

[cudlaDevAttribute](#)

**devPtrs**

[cudlaSignalEvents](#)

**eofFences**

[cudlaSignalEvents](#)

**extBufObject**

[cudlaExternalMemoryHandleDesc](#)

**extSyncObject**

[cudlaExternalSemaphoreHandleDesc](#)

**fence**

[CudlaFence](#)

**inputTensor**

[cudlaTask](#)

**inputTensorDesc**

[cudlaModuleAttribute](#)

**moduleHandle**

[cudlaTask](#)

**numEvents**

[cudlaWaitEvents](#)

[cudlaSignalEvents](#)

**numInputTensors**

[cudlaTask](#)

[cudlaModuleAttribute](#)

**numOutputTensors**

[cudlaTask](#)

[cudlaModuleAttribute](#)

**outputTensor**

[cudlaTask](#)

**outputTensorDesc**

[cudlaModuleAttribute](#)



**preFences**[cudaWaitEvents](#)**signalEvents**[cudaTask](#)**size**[cudaExternalMemoryHandleDesc](#)**type**[CudlaFence](#)**unifiedAddressingSupported**[cudaDevAttribute](#)**waitEvents**[cudaTask](#)

## Notice

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.

## OpenCL

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

## Trademarks

NVIDIA and the NVIDIA logo are trademarks or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

## Copyright

© 2021-2021 NVIDIA Corporation & affiliates. All rights reserved.