NAME

Direct3D 10 Interoperability [DEPRECATED] −

Enumerations

enum **CUD3D10map_flags**
enum **CUD3D10register_flags**

Functions

CUresult cuD3D10CtxCreate (**CUcontext** *pCtx, **CUdevice** *pCudaDevice, unsigned int Flags, ID3D10Device *pD3DDevice)
Create a CUDA context for interoperability with Direct3D 10.

CUresult cuD3D10CtxCreateOnDevice (**CUcontext** *pCtx, unsigned int flags, ID3D10Device *pD3DDevice, **CUdevice** cudaDevice)
Create a CUDA context for interoperability with Direct3D 10.

CUresult cuD3D10GetDirect3DDevice (ID3D10Device **ppD3DDevice)
Get the Direct3D 10 device against which the current CUDA context was created.

CUresult cuD3D10MapResources (unsigned int count, ID3D10Resource **ppResources)
Map Direct3D resources for access by CUDA.

CUresult cuD3D10RegisterResource (ID3D10Resource *pResource, unsigned int Flags)
Register a Direct3D resource for access by CUDA.

CUresult cuD3D10ResourceGetMappedArray (**CUarray** *pArray, ID3D10Resource *pResource, unsigned int SubResource)
Get an array through which to access a subresource of a Direct3D resource which has been mapped for access by CUDA.

CUresult cuD3D10ResourceGetMappedPitch (size_t *pPitch, size_t *pPitchSlice, ID3D10Resource *pResource, unsigned int SubResource)
Get the pitch of a subresource of a Direct3D resource which has been mapped for access by CUDA.

CUresult cuD3D10ResourceGetMappedPointer (**CUdeviceptr** *pDevPtr, ID3D10Resource *pResource, unsigned int SubResource)
Get a pointer through which to access a subresource of a Direct3D resource which has been mapped for access by CUDA.

CUresult cuD3D10ResourceGetMappedSize (size_t *pSize, ID3D10Resource *pResource, unsigned int SubResource)
Get the size of a subresource of a Direct3D resource which has been mapped for access by CUDA.

CUresult cuD3D10ResourceGetSurfaceDimensions (size_t *pWidth, size_t *pHeight, size_t *pDepth, ID3D10Resource *pResource, unsigned int SubResource)
Get the dimensions of a registered surface.

CUresult cuD3D10ResourceSetMapFlags (ID3D10Resource *pResource, unsigned int Flags)
Set usage flags for mapping a Direct3D resource.

CUresult cuD3D10UnmapResources (unsigned int count, ID3D10Resource **ppResources)
Unmap Direct3D resources.

CUresult cuD3D10UnregisterResource (ID3D10Resource *pResource)
Unregister a Direct3D resource.

Detailed Description

\brief deprecated Direct3D 10 interoperability functions of the low-level CUDA driver API (cudaD3D10.h)

This section describes deprecated Direct3D 10 interoperability functionality.

Enumeration Type Documentation

enum CUD3D10map_flags

Flags to map or unmap a resource

enum CUD3D10register_flags

Flags to register a resource

Function Documentation

CUresult cuD3D10CtxCreate (CUcontext * pCtx, CUdevice * pCudaDevice, unsigned int Flags, ID3D10Device * pD3DDevice)

Deprecated

This function is deprecated as of CUDA 5.0.

This function is deprecated and should no longer be used. It is no longer necessary to associate a CUDA context with a D3D10 device in order to achieve maximum interoperability performance.

Parameters:
pCtx - Returned newly created CUDA context
pCudaDevice - Returned pointer to the device on which the context was created
Flags - Context creation flags (see **cuCtxCreate()** for details)
pD3DDevice - Direct3D device to create interoperability context with

Returns:
CUDA_SUCCESS, **CUDA_ERROR_DEINITIALIZED**, **CUDA_ERROR_NOT_INITIALIZED**, **CUDA_ERROR_INVALID_VALUE**, **CUDA_ERROR_OUT_OF_MEMORY**, **CUDA_ERROR_UNKNOWN**

Note:
Note that this function may also return error codes from previous, asynchronous launches.

See also:
cuD3D10GetDevice, **cuGraphicsD3D10RegisterResource**

CUresult cuD3D10CtxCreateOnDevice (CUcontext * pCtx, unsigned int flags, ID3D10Device * pD3DDevice, CUdevice cudaDevice)

Deprecated
This function is deprecated as of CUDA 5.0.

This function is deprecated and should no longer be used. It is no longer necessary to associate a CUDA context with a D3D10 device in order to achieve maximum interoperability performance.

Parameters:
pCtx - Returned newly created CUDA context
flags - Context creation flags (see **cuCtxCreate()** for details)
pD3DDevice - Direct3D device to create interoperability context with
cudaDevice - The CUDA device on which to create the context. This device must be among the devices returned when querying CU_D3D10_DEVICES_ALL from **cuD3D10GetDevices**.

Returns:
CUDA_SUCCESS, **CUDA_ERROR_DEINITIALIZED**, **CUDA_ERROR_NOT_INITIALIZED**, **CUDA_ERROR_INVALID_VALUE**, **CUDA_ERROR_OUT_OF_MEMORY**, **CUDA_ERROR_UNKNOWN**

Note:
Note that this function may also return error codes from previous, asynchronous launches.

See also:
cuD3D10GetDevices, **cuGraphicsD3D10RegisterResource**

CUresult cuD3D10GetDirect3DDevice (ID3D10Device ** ppD3DDevice)

Deprecated
This function is deprecated as of CUDA 5.0.

This function is deprecated and should no longer be used. It is no longer necessary to associate a CUDA context with a D3D10 device in order to achieve maximum interoperability performance.

Parameters:
ppD3DDevice - Returned Direct3D device corresponding to CUDA context

Returns:
CUDA_SUCCESS, **CUDA_ERROR_DEINITIALIZED**, **CUDA_ERROR_NOT_INITIALIZED**, **CUDA_ERROR_INVALID_CONTEXT**

Note:
Note that this function may also return error codes from previous, asynchronous launches.

See also:
cuD3D10GetDevice

CUresult cuD3D10MapResources (unsigned int count, ID3D10Resource ** ppResources)

Deprecated

This function is deprecated as of CUDA 3.0.

Maps the `count` Direct3D resources in `ppResources` for access by CUDA.

The resources in `ppResources` may be accessed in CUDA kernels until they are unmapped. Direct3D should not access any resources while they are mapped by CUDA. If an application does so, the results are undefined.

This function provides the synchronization guarantee that any Direct3D calls issued before **cuD3D10MapResources()** will complete before any CUDA kernels issued after **cuD3D10MapResources()** begin.

If any of `ppResources` have not been registered for use with CUDA or if `ppResources` contains any duplicate entries, then **CUDA_ERROR_INVALID_HANDLE** is returned. If any of `ppResources` are presently mapped for access by CUDA, then **CUDA_ERROR_ALREADY_MAPPED** is returned.

Parameters:

count - Number of resources to map for CUDA
ppResources - Resources to map for CUDA

Returns:

CUDA_SUCCESS, **CUDA_ERROR_DEINITIALIZED**, **CUDA_ERROR_NOT_INITIALIZED**, **CUDA_ERROR_INVALID_CONTEXT**, **CUDA_ERROR_INVALID_HANDLE**, **CUDA_ERROR_ALREADY_MAPPED**, **CUDA_ERROR_UNKNOWN**

Note:

Note that this function may also return error codes from previous, asynchronous launches.

See also:

cuGraphicsMapResources

CUresult cuD3D10RegisterResource (ID3D10Resource * pResource, unsigned int Flags)

Deprecated

This function is deprecated as of CUDA 3.0.

Registers the Direct3D resource `pResource` for access by CUDA.

If this call is successful, then the application will be able to map and unmap this resource until it is unregistered through **cuD3D10UnregisterResource()**. Also on success, this call will increase the internal reference count on `pResource`. This reference count will be decremented when this resource is unregistered through **cuD3D10UnregisterResource()**.

This call is potentially high-overhead and should not be called every frame in interactive applications.

The type of `pResource` must be one of the following.

- ID3D10Buffer: Cannot be used with `Flags` set to CU_D3D10_REGISTER_FLAGS_ARRAY.
- ID3D10Texture1D: No restrictions.
- ID3D10Texture2D: No restrictions.
- ID3D10Texture3D: No restrictions.

The `Flags` argument specifies the mechanism through which CUDA will access the Direct3D resource. The following values are allowed.

- CU_D3D10_REGISTER_FLAGS_NONE: Specifies that CUDA will access this resource through a **CUdeviceptr**. The pointer, size, and (for textures), pitch for each subresource of this allocation may be queried through **cuD3D10ResourceGetMappedPointer()**, **cuD3D10ResourceGetMappedSize()**, and **cuD3D10ResourceGetMappedPitch()** respectively. This option is valid for all resource types.
- CU_D3D10_REGISTER_FLAGS_ARRAY: Specifies that CUDA will access this resource through a **CUarray** queried on a sub-resource basis through **cuD3D10ResourceGetMappedArray()**. This option is only valid for resources of type ID3D10Texture1D, ID3D10Texture2D, and ID3D10Texture3D.

Not all Direct3D resources of the above types may be used for interoperability with CUDA. The following are some limitations.

- The primary rendertarget may not be registered with CUDA.
- Resources allocated as shared may not be registered with CUDA.
- Textures which are not of a format which is 1, 2, or 4 channels of 8, 16, or 32-bit integer or floating-point data cannot be shared.
- Surfaces of depth or stencil formats cannot be shared.

If Direct3D interoperability is not initialized on this context then **CUDA_ERROR_INVALID_CONTEXT** is returned. If `pResource` is of incorrect type or is already registered, then **CUDA_ERROR_INVALID_HANDLE** is returned. If `pResource` cannot be registered, then **CUDA_ERROR_UNKNOWN** is returned.

Parameters:

pResource - Resource to register
Flags - Parameters for resource registration

Returns:

CUDA_SUCCESS, **CUDA_ERROR_DEINITIALIZED**, **CUDA_ERROR_NOT_INITIALIZED**, **CUDA_ERROR_INVALID_CONTEXT**, **CUDA_ERROR_INVALID_VALUE**, **CUDA_ERROR_INVALID_HANDLE**, **CUDA_ERROR_OUT_OF_MEMORY**, **CUDA_ERROR_UNKNOWN**

Note:

Note that this function may also return error codes from previous, asynchronous launches.

See also:

cuGraphicsD3D10RegisterResource

CUresult cuD3D10ResourceGetMappedArray (CUarray * pArray, ID3D10Resource * pResource, unsigned int SubResource)

Deprecated

This function is deprecated as of CUDA 3.0.

Returns in `*pArray` an array through which the subresource of the mapped Direct3D resource `pResource`, which corresponds to `SubResource` may be accessed. The value set in `pArray` may change every time that `pResource` is mapped.

If `pResource` is not registered, then **CUDA_ERROR_INVALID_HANDLE** is returned. If `pResource` was not registered with usage flags CU_D3D10_REGISTER_FLAGS_ARRAY, then **CUDA_ERROR_INVALID_HANDLE** is returned. If `pResource` is not mapped, then **CUDA_ERROR_NOT_MAPPED** is returned.

For usage requirements of the `SubResource` parameter, see **cuD3D10ResourceGetMappedPointer()**.

Parameters:

pArray - Returned array corresponding to subresource
pResource - Mapped resource to access
SubResource - Subresource of pResource to access

Returns:

CUDA_SUCCESS, **CUDA_ERROR_DEINITIALIZED**, **CUDA_ERROR_NOT_INITIALIZED**, **CUDA_ERROR_INVALID_CONTEXT**, **CUDA_ERROR_INVALID_VALUE**, **CUDA_ERROR_INVALID_HANDLE**, **CUDA_ERROR_NOT_MAPPED**

Note:

Note that this function may also return error codes from previous, asynchronous launches.

See also:

cuGraphicsSubResourceGetMappedArray

CUresult cuD3D10ResourceGetMappedPitch (size_t * pPitch, size_t * pPitchSlice, ID3D10Resource * pResource, unsigned int SubResource)

Deprecated

This function is deprecated as of CUDA 3.0.

Returns in `*pPitch` and `*pPitchSlice` the pitch and Z-slice pitch of the subresource of the mapped Direct3D resource `pResource`, which corresponds to `SubResource`. The values set in `pPitch` and `pPitchSlice` may change every time that `pResource` is mapped.

The pitch and Z-slice pitch values may be used to compute the location of a sample on a surface as follows.

For a 2D surface, the byte offset of the sample at position **x**, **y** from the base pointer of the surface is:

y * **pitch** + (**bytes per pixel**) * **x**

For a 3D surface, the byte offset of the sample at position **x**, **y**, **z** from the base pointer of the surface is:

z* **slicePitch** + **y** * **pitch** + (**bytes per pixel**) * **x**

Both parameters `pPitch` and `pPitchSlice` are optional and may be set to NULL.

If `pResource` is not of type IDirect3DBaseTexture10 or one of its sub-types or if `pResource` has not been registered for use with CUDA, then **CUDA_ERROR_INVALID_HANDLE** is returned. If `pResource` was not registered with usage flags CU_D3D10_REGISTER_FLAGS_NONE, then **CUDA_ERROR_INVALID_HANDLE** is returned. If `pResource` is not mapped for access by CUDA, then **CUDA_ERROR_NOT_MAPPED** is returned.

For usage requirements of the `SubResource` parameter, see **cuD3D10ResourceGetMappedPointer()**.

Parameters:

pPitch - Returned pitch of subresource
pPitchSlice - Returned Z-slice pitch of subresource
pResource - Mapped resource to access
SubResource - Subresource of pResource to access

Returns:

CUDA_SUCCESS, **CUDA_ERROR_DEINITIALIZED**, **CUDA_ERROR_NOT_INITIALIZED**, **CUDA_ERROR_INVALID_CONTEXT**, **CUDA_ERROR_INVALID_VALUE**, **CUDA_ERROR_INVALID_HANDLE**, **CUDA_ERROR_NOT_MAPPED**

Note:

Note that this function may also return error codes from previous, asynchronous launches.

See also:

cuGraphicsSubResourceGetMappedArray

CUresult cuD3D10ResourceGetMappedPointer (CUdeviceptr * pDevPtr, ID3D10Resource * pResource, unsigned int SubResource)

Deprecated

This function is deprecated as of CUDA 3.0.

Returns in `*pDevPtr` the base pointer of the subresource of the mapped Direct3D resource `pResource`, which corresponds to `SubResource`. The value set in `pDevPtr` may change every time that `pResource` is mapped.

If `pResource` is not registered, then **CUDA_ERROR_INVALID_HANDLE** is returned. If `pResource` was not registered with usage flags CU_D3D10_REGISTER_FLAGS_NONE, then **CUDA_ERROR_INVALID_HANDLE** is returned. If `pResource` is not mapped, then **CUDA_ERROR_NOT_MAPPED** is returned.

If `pResource` is of type ID3D10Buffer, then `SubResource` must be 0. If `pResource` is of any other type, then the value of `SubResource` must come from the subresource calculation in D3D10CalcSubResource().

Parameters:

pDevPtr - Returned pointer corresponding to subresource

pResource - Mapped resource to access
SubResource - Subresource of pResource to access

Returns:

CUDA_SUCCESS, **CUDA_ERROR_DEINITIALIZED**, **CUDA_ERROR_NOT_INITIALIZED**, **CUDA_ERROR_INVALID_CONTEXT**, **CUDA_ERROR_INVALID_VALUE**, **CUDA_ERROR_INVALID_HANDLE**, **CUDA_ERROR_NOT_MAPPED**

Note:

Note that this function may also return error codes from previous, asynchronous launches.

See also:

cuGraphicsResourceGetMappedPointer

CUresult cuD3D10ResourceGetMappedSize (size_t * pSize, ID3D10Resource * pResource, unsigned int SubResource)

Deprecated

This function is deprecated as of CUDA 3.0.

Returns in `*pSize` the size of the subresource of the mapped Direct3D resource `pResource`, which corresponds to `SubResource`. The value set in `pSize` may change every time that `pResource` is mapped.

If `pResource` has not been registered for use with CUDA, then **CUDA_ERROR_INVALID_HANDLE** is returned. If `pResource` was not registered with usage flags CU_D3D10_REGISTER_FLAGS_NONE, then **CUDA_ERROR_INVALID_HANDLE** is returned. If `pResource` is not mapped for access by CUDA, then **CUDA_ERROR_NOT_MAPPED** is returned.

For usage requirements of the `SubResource` parameter, see **cuD3D10ResourceGetMappedPointer()**.

Parameters:

pSize - Returned size of subresource
pResource - Mapped resource to access
SubResource - Subresource of pResource to access

Returns:

CUDA_SUCCESS, **CUDA_ERROR_DEINITIALIZED**, **CUDA_ERROR_NOT_INITIALIZED**, **CUDA_ERROR_INVALID_CONTEXT**, **CUDA_ERROR_INVALID_VALUE**, **CUDA_ERROR_INVALID_HANDLE**, **CUDA_ERROR_NOT_MAPPED**

Note:

Note that this function may also return error codes from previous, asynchronous launches.

See also:

cuGraphicsResourceGetMappedPointer

CUresult cuD3D10ResourceGetSurfaceDimensions (size_t * pWidth, size_t * pHeight, size_t * pDepth, ID3D10Resource * pResource, unsigned int SubResource)

Deprecated

This function is deprecated as of CUDA 3.0.

Returns in `*pWidth`, `*pHeight`, and `*pDepth` the dimensions of the subresource of the mapped Direct3D resource `pResource`, which corresponds to `SubResource`.

Because anti-aliased surfaces may have multiple samples per pixel, it is possible that the dimensions of a resource will be an integer factor larger than the dimensions reported by the Direct3D runtime.

The parameters `pWidth`, `pHeight`, and `pDepth` are optional. For 2D surfaces, the value returned in `*pDepth` will be 0.

If `pResource` is not of type IDirect3DBaseTexture10 or IDirect3DSurface10 or if `pResource` has not been registered for use with CUDA, then **CUDA_ERROR_INVALID_HANDLE** is returned.

For usage requirements of the `SubResource` parameter, see **cuD3D10ResourceGetMappedPointer()**.

Parameters:

pWidth - Returned width of surface
pHeight - Returned height of surface
pDepth - Returned depth of surface
pResource - Registered resource to access
SubResource - Subresource of pResource to access

Returns:

CUDA_SUCCESS, **CUDA_ERROR_DEINITIALIZED**, **CUDA_ERROR_NOT_INITIALIZED**, **CUDA_ERROR_INVALID_CONTEXT**, **CUDA_ERROR_INVALID_VALUE**, **CUDA_ERROR_INVALID_HANDLE**

Note:

Note that this function may also return error codes from previous, asynchronous launches.

See also:

cuGraphicsSubResourceGetMappedArray

CUresult cuD3D10ResourceSetMapFlags (ID3D10Resource * pResource, unsigned int Flags)

Deprecated

This function is deprecated as of CUDA 3.0.

Set flags for mapping the Direct3D resource `pResource`.

Changes to flags will take effect the next time `pResource` is mapped. The `Flags` argument may be any of the following.

- CU_D3D10_MAPRESOURCE_FLAGS_NONE: Specifies no hints about how this resource will be used. It is therefore assumed that this resource will be read from and written to by CUDA kernels. This is the default value.
- CU_D3D10_MAPRESOURCE_FLAGS_READONLY: Specifies that CUDA kernels which access this resource will not write to this resource.
- CU_D3D10_MAPRESOURCE_FLAGS_WRITEDISCARD: Specifies that CUDA kernels which access this resource will not read from this resource and will write over the entire contents of the resource, so none of the data previously stored in the resource will be preserved.

If `pResource` has not been registered for use with CUDA, then **CUDA_ERROR_INVALID_HANDLE** is returned. If `pResource` is presently mapped for access by CUDA then **CUDA_ERROR_ALREADY_MAPPED** is returned.

Parameters:

pResource - Registered resource to set flags for
Flags - Parameters for resource mapping

Returns:

CUDA_SUCCESS, **CUDA_ERROR_DEINITIALIZED**, **CUDA_ERROR_NOT_INITIALIZED**, **CUDA_ERROR_INVALID_CONTEXT**, **CUDA_ERROR_INVALID_VALUE**, **CUDA_ERROR_INVALID_HANDLE**, **CUDA_ERROR_ALREADY_MAPPED**

Note:

Note that this function may also return error codes from previous, asynchronous launches.

See also:

cuGraphicsResourceSetMapFlags

CUresult cuD3D10UnmapResources (unsigned int count, ID3D10Resource ** ppResources)

Deprecated

This function is deprecated as of CUDA 3.0.

Unmaps the `count` Direct3D resources in `ppResources`.

This function provides the synchronization guarantee that any CUDA kernels issued before **cuD3D10UnmapResources()** will complete before any Direct3D calls issued after **cuD3D10UnmapResources()** begin.

If any of `ppResources` have not been registered for use with CUDA or if `ppResources` contains any duplicate entries, then **CUDA_ERROR_INVALID_HANDLE** is returned. If any of `ppResources` are not presently mapped for access by CUDA, then **CUDA_ERROR_NOT_MAPPED** is returned.

Parameters:

count - Number of resources to unmap for CUDA
ppResources - Resources to unmap for CUDA

Returns:

CUDA_SUCCESS, **CUDA_ERROR_DEINITIALIZED**, **CUDA_ERROR_NOT_INITIALIZED**, **CUDA_ERROR_INVALID_CONTEXT**, **CUDA_ERROR_INVALID_VALUE**, **CUDA_ERROR_INVALID_HANDLE**, **CUDA_ERROR_NOT_MAPPED**, **CUDA_ERROR_UNKNOWN**

Note:

Note that this function may also return error codes from previous, asynchronous launches.

See also:

cuGraphicsUnmapResources

CUresult cuD3D10UnregisterResource (ID3D10Resource * pResource)

Deprecated

This function is deprecated as of CUDA 3.0.

Unregisters the Direct3D resource `pResource` so it is not accessible by CUDA unless registered again.

If `pResource` is not registered, then **CUDA_ERROR_INVALID_HANDLE** is returned.

Parameters:

pResource - Resources to unregister

Returns:

CUDA_SUCCESS, **CUDA_ERROR_DEINITIALIZED**, **CUDA_ERROR_NOT_INITIALIZED**, **CUDA_ERROR_INVALID_CONTEXT**, **CUDA_ERROR_INVALID_HANDLE**, **CUDA_ERROR_UNKNOWN**

Note:

Note that this function may also return error codes from previous, asynchronous launches.

See also:

cuGraphicsUnregisterResource

Author

Generated automatically by Doxygen from the source code.