

NAME

Texture Reference Management –

Functions

- CUresult cuTexRefGetAddress** (CUdeviceptr *pdptr, CUtexref hTexRef)
Gets the address associated with a texture reference.
- CUresult cuTexRefGetAddressMode** (CUaddress_mode *pam, CUtexref hTexRef, int dim)
Gets the addressing mode used by a texture reference.
- CUresult cuTexRefGetArray** (CUarray *phArray, CUtexref hTexRef)
Gets the array bound to a texture reference.
- CUresult cuTexRefGetFilterMode** (CUfilter_mode *pfm, CUtexref hTexRef)
Gets the filter-mode used by a texture reference.
- CUresult cuTexRefGetFlags** (unsigned int *pFlags, CUtexref hTexRef)
Gets the flags used by a texture reference.
- CUresult cuTexRefGetFormat** (CUarray_format *pFormat, int *pNumChannels, CUtexref hTexRef)
Gets the format used by a texture reference.
- CUresult cuTexRefGetMaxAnisotropy** (int *pmaxAniso, CUtexref hTexRef)
Gets the maximum anisotropy for a texture reference.
- CUresult cuTexRefGetMipmapFilterMode** (CUfilter_mode *pfm, CUtexref hTexRef)
Gets the mipmap filtering mode for a texture reference.
- CUresult cuTexRefGetMipmapLevelBias** (float *pbias, CUtexref hTexRef)
Gets the mipmap level bias for a texture reference.
- CUresult cuTexRefGetMipmapLevelClamp** (float *pminMipmapLevelClamp, float *pmaxMipmapLevelClamp, CUtexref hTexRef)
Gets the min/max mipmap level clamps for a texture reference.
- CUresult cuTexRefGetMipmappedArray** (CUmipmappedArray *phMipmappedArray, CUtexref hTexRef)
Gets the mipmapped array bound to a texture reference.
- CUresult cuTexRefSetAddress** (size_t *ByteOffset, CUtexref hTexRef, CUdeviceptr dptr, size_t bytes)
Binds an address as a texture reference.
- CUresult cuTexRefSetAddress2D** (CUtexref hTexRef, const CUDA_ARRAY_DESCRIPTOR *desc, CUdeviceptr dptr, size_t Pitch)
Binds an address as a 2D texture reference.
- CUresult cuTexRefSetAddressMode** (CUtexref hTexRef, int dim, CUaddress_mode am)
Sets the addressing mode for a texture reference.
- CUresult cuTexRefSetArray** (CUtexref hTexRef, CUarray hArray, unsigned int Flags)
Binds an array as a texture reference.
- CUresult cuTexRefSetFilterMode** (CUtexref hTexRef, CUfilter_mode fm)
Sets the filtering mode for a texture reference.
- CUresult cuTexRefSetFlags** (CUtexref hTexRef, unsigned int Flags)
Sets the flags for a texture reference.
- CUresult cuTexRefSetFormat** (CUtexref hTexRef, CUarray_format fmt, int NumPackedComponents)
Sets the format for a texture reference.
- CUresult cuTexRefSetMaxAnisotropy** (CUtexref hTexRef, unsigned int maxAniso)
Sets the maximum anisotropy for a texture reference.
- CUresult cuTexRefSetMipmapFilterMode** (CUtexref hTexRef, CUfilter_mode fm)
Sets the mipmap filtering mode for a texture reference.
- CUresult cuTexRefSetMipmapLevelBias** (CUtexref hTexRef, float bias)
Sets the mipmap level bias for a texture reference.
- CUresult cuTexRefSetMipmapLevelClamp** (CUtexref hTexRef, float minMipmapLevelClamp, float maxMipmapLevelClamp)
Sets the mipmap min/max mipmap level clamps for a texture reference.
- CUresult cuTexRefSetMipmappedArray** (CUtexref hTexRef, CUmipmappedArray hMipmappedArray, unsigned int Flags)
Binds a mipmapped array to a texture reference.



Detailed Description

\brief texture reference management functions of the low-level CUDA driver API (**cuda.h**)

This section describes the texture reference management functions of the low-level CUDA driver application programming interface.

Function Documentation

CUresult cuTexRefGetAddress (CUdeviceptr * pdptr, CUtexref hTexRef)

Returns in *pdptr the base address bound to the texture reference hTexRef, or returns **CUDA_ERROR_INVALID_VALUE** if the texture reference is not bound to any device memory range.

Parameters:

pdptr - Returned device address
hTexRef - Texture reference

Returns:

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

See also:

cuTexRefSetAddress, **cuTexRefSetAddress2D**, **cuTexRefSetAddressMode**,
cuTexRefSetArray, **cuTexRefSetFilterMode**, **cuTexRefSetFlags**, **cuTexRefSetFormat**,
cuTexRefGetAddressMode, **cuTexRefGetArray**, **cuTexRefGetFilterMode**,
cuTexRefGetFlags, **cuTexRefGetFormat**

CUresult cuTexRefGetAddressMode (CUaddress_mode * pam, CUtexref hTexRef, int dim)

Returns in *pam the addressing mode corresponding to the dimension dim of the texture reference hTexRef. Currently, the only valid value for dim are 0 and 1.

Parameters:

pam - Returned addressing mode
hTexRef - Texture reference
dim - Dimension

Returns:

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

See also:

cuTexRefSetAddress, **cuTexRefSetAddress2D**, **cuTexRefSetAddressMode**,
cuTexRefSetArray, **cuTexRefSetFilterMode**, **cuTexRefSetFlags**, **cuTexRefSetFormat**,
cuTexRefGetAddress, **cuTexRefGetArray**, **cuTexRefGetFilterMode**, **cuTexRefGetFlags**,
cuTexRefGetFormat

CUresult cuTexRefGetArray (CUarray * phArray, CUtexref hTexRef)

Returns in *phArray the CUDA array bound to the texture reference hTexRef, or returns **CUDA_ERROR_INVALID_VALUE** if the texture reference is not bound to any CUDA array.

Parameters:

phArray - Returned array
hTexRef - Texture reference

Returns:

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

See also:

cuTexRefSetAddress, **cuTexRefSetAddress2D**, **cuTexRefSetAddressMode**,
cuTexRefSetArray, **cuTexRefSetFilterMode**, **cuTexRefSetFlags**, **cuTexRefSetFormat**,
cuTexRefGetAddress, **cuTexRefGetAddressMode**, **cuTexRefGetFilterMode**,
cuTexRefGetFlags, **cuTexRefGetFormat**



CUresult cuTexRefGetFilterMode (CUfilter_mode * pfm, CUtexref hTexRef)

Returns in *pfm the filtering mode of the texture reference hTexRef.

Parameters:

pfm - Returned filtering mode
hTexRef - Texture reference

Returns:

CUDA_SUCCESS, CUDA_ERROR_DEINITIALIZED,
 CUDA_ERROR_NOT_INITIALIZED, CUDA_ERROR_INVALID_CONTEXT,
 CUDA_ERROR_INVALID_VALUE

See also:

cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode,
 cuTexRefSetArray, cuTexRefSetFilterMode, cuTexRefSetFlags, cuTexRefSetFormat,
 cuTexRefGetAddress, cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFlags,
 cuTexRefGetFormat

CUresult cuTexRefGetFlags (unsigned int * pFlags, CUtexref hTexRef)

Returns in *pFlags the flags of the texture reference hTexRef.

Parameters:

pFlags - Returned flags
hTexRef - Texture reference

Returns:

CUDA_SUCCESS, CUDA_ERROR_DEINITIALIZED,
 CUDA_ERROR_NOT_INITIALIZED, CUDA_ERROR_INVALID_CONTEXT,
 CUDA_ERROR_INVALID_VALUE

See also:

cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode,
 cuTexRefSetArray, cuTexRefSetFilterMode, cuTexRefSetFlags, cuTexRefSetFormat,
 cuTexRefGetAddress, cuTexRefGetAddressMode, cuTexRefGetArray,
 cuTexRefGetFilterMode, cuTexRefGetFormat

CUresult cuTexRefGetFormat (CUarray_format * pFormat, int * pNumChannels, CUtexref hTexRef)

Returns in *pFormat and *pNumChannels the format and number of components of the CUDA array bound to the texture reference hTexRef. If pFormat or pNumChannels is NULL, it will be ignored.

Parameters:

pFormat - Returned format
pNumChannels - Returned number of components
hTexRef - Texture reference

Returns:

CUDA_SUCCESS, CUDA_ERROR_DEINITIALIZED,
 CUDA_ERROR_NOT_INITIALIZED, CUDA_ERROR_INVALID_CONTEXT,
 CUDA_ERROR_INVALID_VALUE

See also:

cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode,
 cuTexRefSetArray, cuTexRefSetFilterMode, cuTexRefSetFlags, cuTexRefSetFormat,
 cuTexRefGetAddress, cuTexRefGetAddressMode, cuTexRefGetArray,
 cuTexRefGetFilterMode, cuTexRefGetFlags

CUresult cuTexRefGetMaxAnisotropy (int * pmaxAniso, CUtexref hTexRef)

Returns the maximum anisotropy in pmaxAniso that's used when reading memory through the texture reference hTexRef.

Parameters:

pmaxAniso - Returned maximum anisotropy
hTexRef - Texture reference

Returns:

CUDA_SUCCESS, CUDA_ERROR_DEINITIALIZED,



**CUDA_ERROR_NOT_INITIALIZED, CUDA_ERROR_INVALID_CONTEXT,
CUDA_ERROR_INVALID_VALUE**

See also:

**cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode,
cuTexRefSetArray, cuTexRefSetFlags, cuTexRefSetFormat, cuTexRefGetAddress,
cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFilterMode,
cuTexRefGetFlags, cuTexRefGetFormat**

CUresult cuTexRefGetMipmapFilterMode (CUfilter_mode * pfm, CUtexref hTexRef)

Returns the mipmap filtering mode in *pfm* that's used when reading memory through the texture reference *hTexRef*.

Parameters:

pfm - Returned mipmap filtering mode
hTexRef - Texture reference

Returns:

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

See also:

**cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode,
cuTexRefSetArray, cuTexRefSetFlags, cuTexRefSetFormat, cuTexRefGetAddress,
cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFilterMode,
cuTexRefGetFlags, cuTexRefGetFormat**

CUresult cuTexRefGetMipmapLevelBias (float * pbias, CUtexref hTexRef)

Returns the mipmap level bias in *pbias* that's added to the specified mipmap level when reading memory through the texture reference *hTexRef*.

Parameters:

pbias - Returned mipmap level bias
hTexRef - Texture reference

Returns:

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

See also:

**cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode,
cuTexRefSetArray, cuTexRefSetFlags, cuTexRefSetFormat, cuTexRefGetAddress,
cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFilterMode,
cuTexRefGetFlags, cuTexRefGetFormat**

**CUresult cuTexRefGetMipmapLevelClamp (float * pminMipmapLevelClamp, float *
pmaxMipmapLevelClamp, CUtexref hTexRef)**

Returns the min/max mipmap level clamps in *pminMipmapLevelClamp* and *pmaxMipmapLevelClamp* that's used when reading memory through the texture reference *hTexRef*.

Parameters:

pminMipmapLevelClamp - Returned mipmap min level clamp
pmaxMipmapLevelClamp - Returned mipmap max level clamp
hTexRef - Texture reference

Returns:

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

See also:

**cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode,
cuTexRefSetArray, cuTexRefSetFlags, cuTexRefSetFormat, cuTexRefGetAddress,**



**cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFilterMode,
cuTexRefGetFlags, cuTexRefGetFormat**

CUresult cuTexRefGetMipmappedArray (CUmipmappedArray * phMipmappedArray, CUtexref hTexRef)

Returns in *phMipmappedArray the CUDA mipmapped array bound to the texture reference hTexRef, or returns **CUDA_ERROR_INVALID_VALUE** if the texture reference is not bound to any CUDA mipmapped array.

Parameters:

phMipmappedArray - Returned mipmapped array
hTexRef - Texture reference

Returns:

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

See also:

**cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode,
cuTexRefSetArray, cuTexRefSetFilterMode, cuTexRefSetFlags, cuTexRefSetFormat,
cuTexRefGetAddress, cuTexRefGetAddressMode, cuTexRefGetFilterMode,
cuTexRefGetFlags, cuTexRefGetFormat**

CUresult cuTexRefSetAddress (size_t * ByteOffset, CUtexref hTexRef, CUdeviceptr dptr, size_t bytes)

Binds a linear address range to the texture reference hTexRef. Any previous address or CUDA array state associated with the texture reference is superseded by this function. Any memory previously bound to hTexRef is unbound.

Since the hardware enforces an alignment requirement on texture base addresses,

cuTexRefSetAddress() passes back a byte offset in *ByteOffset that must be applied to texture fetches in order to read from the desired memory. This offset must be divided by the texel size and passed to kernels that read from the texture so they can be applied to the tex1Dfetch() function.

If the device memory pointer was returned from **cuMemAlloc()**, the offset is guaranteed to be 0 and NULL may be passed as the ByteOffset parameter.

The total number of elements (or texels) in the linear address range cannot exceed

CU_DEVICE_ATTRIBUTE_MAXIMUM_TEXTURE1D_LINEAR_WIDTH. The number of elements is computed as (bytes / bytesPerElement), where bytesPerElement is determined from the data format and number of components set using **cuTexRefSetFormat()**.

Parameters:

ByteOffset - Returned byte offset
hTexRef - Texture reference to bind
dptr - Device pointer to bind
bytes - Size of memory to bind in bytes

Returns:

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

See also:

**cuTexRefSetAddress2D, cuTexRefSetAddressMode, cuTexRefSetArray,
cuTexRefSetFilterMode, cuTexRefSetFlags, cuTexRefSetFormat, cuTexRefGetAddress,
cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFilterMode,
cuTexRefGetFlags, cuTexRefGetFormat**

CUresult cuTexRefSetAddress2D (CUtexref hTexRef, const CUDA_ARRAY_DESCRIPTOR * desc, CUdeviceptr dptr, size_t Pitch)

Binds a linear address range to the texture reference hTexRef. Any previous address or CUDA array state associated with the texture reference is superseded by this function. Any memory previously bound to hTexRef is unbound.

Using a tex2D() function inside a kernel requires a call to either **cuTexRefSetArray()** to bind the



corresponding texture reference to an array, or **cuTexRefSetAddress2D()** to bind the texture reference to linear memory.

Function calls to **cuTexRefSetFormat()** cannot follow calls to **cuTexRefSetAddress2D()** for the same texture reference.

It is required that **dptr** be aligned to the appropriate hardware-specific texture alignment. You can query this value using the device attribute **CU_DEVICE_ATTRIBUTE_TEXTURE_ALIGNMENT**. If an unaligned **dptr** is supplied, **CUDA_ERROR_INVALID_VALUE** is returned.

Pitch has to be aligned to the hardware-specific texture pitch alignment. This value can be queried using the device attribute **CU_DEVICE_ATTRIBUTE_TEXTURE_PITCH_ALIGNMENT**. If an unaligned **Pitch** is supplied, **CUDA_ERROR_INVALID_VALUE** is returned.

Width and **Height**, which are specified in elements (or texels), cannot exceed **CU_DEVICE_ATTRIBUTE_MAXIMUM_TEXTURE2D_LINEAR_WIDTH** and **CU_DEVICE_ATTRIBUTE_MAXIMUM_TEXTURE2D_LINEAR_HEIGHT** respectively.

Pitch, which is specified in bytes, cannot exceed **CU_DEVICE_ATTRIBUTE_MAXIMUM_TEXTURE2D_LINEAR_PITCH**.

Parameters:

hTexRef - Texture reference to bind
desc - Descriptor of CUDA array
dptr - Device pointer to bind
Pitch - Line pitch in bytes

Returns:

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

See also:

cuTexRefSetAddress, **cuTexRefSetAddressMode**, **cuTexRefSetArray**,
cuTexRefSetFilterMode, **cuTexRefSetFlags**, **cuTexRefSetFormat**, **cuTexRefGetAddress**,
cuTexRefGetAddressMode, **cuTexRefGetArray**, **cuTexRefGetFilterMode**,
cuTexRefGetFlags, **cuTexRefGetFormat**

CUresult cuTexRefSetAddressMode (CUtexref hTexRef, int dim, CUaddress_mode am)

Specifies the addressing mode **am** for the given dimension **dim** of the texture reference **hTexRef**. If **dim** is zero, the addressing mode is applied to the first parameter of the functions used to fetch from the texture; if **dim** is 1, the second, and so on. **CUaddress_mode** is defined as:

```
typedef enum CUaddress_mode_enum {
    CU_TR_ADDRESS_MODE_WRAP = 0,
    CU_TR_ADDRESS_MODE_CLAMP = 1,
    CU_TR_ADDRESS_MODE_MIRROR = 2,
    CU_TR_ADDRESS_MODE_BORDER = 3
} CUaddress_mode;
```

Note that this call has no effect if **hTexRef** is bound to linear memory. Also, if the flag, **CU_TRSF_NORMALIZED_COORDINATES**, is not set, the only supported address mode is **CU_TR_ADDRESS_MODE_CLAMP**.

Parameters:

hTexRef - Texture reference
dim - Dimension
am - Addressing mode to set

Returns:

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

See also:

cuTexRefSetAddress, **cuTexRefSetAddress2D**, **cuTexRefSetArray**, **cuTexRefSetFilterMode**,



cuTexRefSetFlags, cuTexRefSetFormat, cuTexRefGetAddress, cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFilterMode, cuTexRefGetFlags, cuTexRefGetFormat

CUresult cuTexRefSetArray (CUtexref hTexRef, CUarray hArray, unsigned int Flags)

Binds the CUDA array *hArray* to the texture reference *hTexRef*. Any previous address or CUDA array state associated with the texture reference is superseded by this function. *Flags* must be set to **CU_TRSA_OVERRIDE_FORMAT**. Any CUDA array previously bound to *hTexRef* is unbound.

Parameters:

hTexRef - Texture reference to bind
hArray - Array to bind
Flags - Options (must be **CU_TRSA_OVERRIDE_FORMAT**)

Returns:

CUDA_SUCCESS, CUDA_ERROR_DEINITIALIZED, CUDA_ERROR_NOT_INITIALIZED, CUDA_ERROR_INVALID_CONTEXT, CUDA_ERROR_INVALID_VALUE

See also:

cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode, cuTexRefSetFilterMode, cuTexRefSetFlags, cuTexRefSetFormat, cuTexRefGetAddress, cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFilterMode, cuTexRefGetFlags, cuTexRefGetFormat

CUresult cuTexRefSetFilterMode (CUtexref hTexRef, CUfilter_mode fm)

Specifies the filtering mode *fm* to be used when reading memory through the texture reference *hTexRef*. **CUfilter_mode_enum** is defined as:

```
typedef enum CUfilter_mode_enum {
    CU_TR_FILTER_MODE_POINT = 0,
    CU_TR_FILTER_MODE_LINEAR = 1
} CUfilter_mode;
```

Note that this call has no effect if *hTexRef* is bound to linear memory.

Parameters:

hTexRef - Texture reference
fm - Filtering mode to set

Returns:

CUDA_SUCCESS, CUDA_ERROR_DEINITIALIZED, CUDA_ERROR_NOT_INITIALIZED, CUDA_ERROR_INVALID_CONTEXT, CUDA_ERROR_INVALID_VALUE

See also:

cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode, cuTexRefSetArray, cuTexRefSetFlags, cuTexRefSetFormat, cuTexRefGetAddress, cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFilterMode, cuTexRefGetFlags, cuTexRefGetFormat

CUresult cuTexRefSetFlags (CUtexref hTexRef, unsigned int Flags)

Specifies optional flags via *Flags* to specify the behavior of data returned through the texture reference *hTexRef*. The valid flags are:

- **CU_TRSF_READ_AS_INTEGER**, which suppresses the default behavior of having the texture promote integer data to floating point data in the range [0, 1]. Note that texture with 32-bit integer format would not be promoted, regardless of whether or not this flag is specified;
- **CU_TRSF_NORMALIZED_COORDINATES**, which suppresses the default behavior of having the texture coordinates range from [0, Dim) where Dim is the width or height of the CUDA array. Instead, the texture coordinates [0, 1.0) reference the entire breadth of the array dimension;

Parameters:

hTexRef - Texture reference
Flags - Optional flags to set

Returns:

CUDA_SUCCESS, CUDA_ERROR_DEINITIALIZED,



**CUDA_ERROR_NOT_INITIALIZED, CUDA_ERROR_INVALID_CONTEXT,
CUDA_ERROR_INVALID_VALUE**

See also:

**cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode,
cuTexRefSetArray, cuTexRefSetFilterMode, cuTexRefSetFormat, cuTexRefGetAddress,
cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFilterMode,
cuTexRefGetFlags, cuTexRefGetFormat**

CUresult cuTexRefSetFormat (CUtexref hTexRef, CUarray_format fmt, int NumPackedComponents)

Specifies the format of the data to be read by the texture reference *hTexRef*. *fmt* and *NumPackedComponents* are exactly analogous to the *Format* and *NumChannels* members of the **CUDA_ARRAY_DESCRIPTOR** structure: They specify the format of each component and the number of components per array element.

Parameters:

hTexRef - Texture reference
fmt - Format to set
NumPackedComponents - Number of components per array element

Returns:

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

See also:

**cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode,
cuTexRefSetArray, cuTexRefSetFilterMode, cuTexRefSetFlags, cuTexRefGetAddress,
cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFilterMode,
cuTexRefGetFlags, cuTexRefGetFormat**

CUresult cuTexRefSetMaxAnisotropy (CUtexref hTexRef, unsigned int maxAniso)

Specifies the maximum anisotropy *maxAniso* to be used when reading memory through the texture reference *hTexRef*.

Note that this call has no effect if *hTexRef* is bound to linear memory.

Parameters:

hTexRef - Texture reference
maxAniso - Maximum anisotropy

Returns:

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

See also:

**cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode,
cuTexRefSetArray, cuTexRefSetFlags, cuTexRefSetFormat, cuTexRefGetAddress,
cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFilterMode,
cuTexRefGetFlags, cuTexRefGetFormat**

CUresult cuTexRefSetMipmapFilterMode (CUtexref hTexRef, CUfilter_mode fm)

Specifies the mipmap filtering mode *fm* to be used when reading memory through the texture reference *hTexRef*. *CUfilter_mode_enum* is defined as:

```
typedef enum CUfilter_mode_enum {
    CU_TR_FILTER_MODE_POINT = 0,
    CU_TR_FILTER_MODE_LINEAR = 1
} CUfilter_mode;
```

Note that this call has no effect if *hTexRef* is not bound to a mipmapped array.

Parameters:

hTexRef - Texture reference
fm - Filtering mode to set

Returns:



CUDA_SUCCESS, CUDA_ERROR_DEINITIALIZED,
 CUDA_ERROR_NOT_INITIALIZED, CUDA_ERROR_INVALID_CONTEXT,
 CUDA_ERROR_INVALID_VALUE

See also:

cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode,
 cuTexRefSetArray, cuTexRefSetFlags, cuTexRefSetFormat, cuTexRefGetAddress,
 cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFilterMode,
 cuTexRefGetFlags, cuTexRefGetFormat

CUresult cuTexRefSetMipmapLevelBias (CUtexref hTexRef, float bias)

Specifies the mipmap level bias *bias* to be added to the specified mipmap level when reading memory through the texture reference *hTexRef*.

Note that this call has no effect if *hTexRef* is not bound to a mipmapped array.

Parameters:

hTexRef - Texture reference
bias - Mipmap level bias

Returns:

CUDA_SUCCESS, CUDA_ERROR_DEINITIALIZED,
 CUDA_ERROR_NOT_INITIALIZED, CUDA_ERROR_INVALID_CONTEXT,
 CUDA_ERROR_INVALID_VALUE

See also:

cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode,
 cuTexRefSetArray, cuTexRefSetFlags, cuTexRefSetFormat, cuTexRefGetAddress,
 cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFilterMode,
 cuTexRefGetFlags, cuTexRefGetFormat

CUresult cuTexRefSetMipmapLevelClamp (CUtexref hTexRef, float minMipmapLevelClamp, float maxMipmapLevelClamp)

Specifies the min/max mipmap level clamps, *minMipmapLevelClamp* and *maxMipmapLevelClamp* respectively, to be used when reading memory through the texture reference *hTexRef*.

Note that this call has no effect if *hTexRef* is not bound to a mipmapped array.

Parameters:

hTexRef - Texture reference
minMipmapLevelClamp - Mipmap min level clamp
maxMipmapLevelClamp - Mipmap max level clamp

Returns:

CUDA_SUCCESS, CUDA_ERROR_DEINITIALIZED,
 CUDA_ERROR_NOT_INITIALIZED, CUDA_ERROR_INVALID_CONTEXT,
 CUDA_ERROR_INVALID_VALUE

See also:

cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode,
 cuTexRefSetArray, cuTexRefSetFlags, cuTexRefSetFormat, cuTexRefGetAddress,
 cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFilterMode,
 cuTexRefGetFlags, cuTexRefGetFormat

CUresult cuTexRefSetMipmappedArray (CUtexref hTexRef, CUmipmappedArray hMipmappedArray, unsigned int Flags)

Binds the CUDA mipmapped array *hMipmappedArray* to the texture reference *hTexRef*. Any previous address or CUDA array state associated with the texture reference is superseded by this function. *Flags* must be set to **CU_TRSA_OVERRIDE_FORMAT**. Any CUDA array previously bound to *hTexRef* is unbound.

Parameters:

hTexRef - Texture reference to bind
hMipmappedArray - Mipmapped array to bind
Flags - Options (must be **CU_TRSA_OVERRIDE_FORMAT**)



Returns:

CUDA_SUCCESS, CUDA_ERROR_DEINITIALIZED,
CUDA_ERROR_NOT_INITIALIZED, CUDA_ERROR_INVALID_CONTEXT,
CUDA_ERROR_INVALID_VALUE

See also:

cuTexRefSetAddress, cuTexRefSetAddress2D, cuTexRefSetAddressMode,
cuTexRefSetFilterMode, cuTexRefSetFlags, cuTexRefSetFormat, cuTexRefGetAddress,
cuTexRefGetAddressMode, cuTexRefGetArray, cuTexRefGetFilterMode,
cuTexRefGetFlags, cuTexRefGetFormat

Author

Generated automatically by Doxygen from the source code.

