CGFunction Reference

Graphics & Imaging > Quartz

2006-12-22

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

CGFunction Reference

Derived From:	CFType
Framework:	ApplicationServices/ApplicationServices.h
Companion guide	Quartz 2D Programming Guide
Declared in	CGFunction.h

Overview

The `CGFunctionRef` opaque type provides a general facility for defining and using callback functions. These functions can take an arbitrary number of floating-point input values and pass back an arbitrary number of floating-point output values.

Quartz uses CGFunction objects to implement shadings. *CGShading Reference* describes the parameters and semantics required for the callbacks used by CGFunction objects.

Functions by Task

Creating a CGFunction Object

`CGFunctionCreate` (page 6)
Creates a Quartz function.

Retaining and Releasing CGFunction Objects

`CGFunctionRelease` (page 7)
Decrements the retain count of a function object.

`CGFunctionRetain` (page 7)
Increments the retain count of a function object.

Getting the CFType ID

`CGFunctionGetTypeID` (page 7)
Returns the type identifier for Quartz function objects.

Functions

CGFunctionCreate

Creates a Quartz function.

```
CGFunctionRef CGFunctionCreate (
   void *info,
   size_t domainDimension,
   const CGFloat *domain,
   size_t rangeDimension,
   const CGFloat *range,
   const CGFunctionCallbacks *callbacks
);
```

Parameters

info

A pointer to user-defined storage for data that you want to pass to your callbacks. You need to make sure that the data persists for as long as it's needed, which can be beyond the scope in which the Quartz function is used.

domainDimension

The number of inputs.

domain

An array of (2*domainDimension) floats used to specify the valid intervals of input values. For each k from 0 to (domainDimension - 1), domain[2*k] must be less than or equal to domain[2*k+1], and the kth input value will be clipped to lie in the interval domain[2*k] ≤ input[k] ≤ domain[2*k+1]. If this parameter is NULL, then the input values are not clipped.

rangeDimension

The number of outputs.

range

An array of (2*rangeDimension) floats that specifies the valid intervals of output values. For each k from 0 to (rangeDimension - 1), range[2*k] must be less than or equal to range[2*k+1], and the kth output value will be clipped to lie in the interval range[2*k] ≤ output[k] ≤ range[2*k+1]. If this parameter is NULL, then the output values are not clipped.

callbacks

A pointer to a callback function table. This table should contain pointers to the callbacks you provide to implement the semantics of this Quartz function. Quartz makes a copy of your table, so, for example, you could safely pass in a pointer to a structure on the stack.

Return Value

The new Quartz function. You are responsible for releasing this object using CGFunctionRelease (page 7).

Availability

Available in Mac OS X version 10.2 and later.

Declared In

CGFunction.h

CGFunctionGetTypeID

Returns the type identifier for Quartz function objects.

```
CFTypeID CGFunctionGetTypeID (
    void
);
```

Return Value

The identifier for the opaque type `CGFunctionRef` (page 9).

Availability

Available in Mac OS X version 10.2 and later.

Declared In

`CGFunction.h`

CGFunctionRelease

Decrements the retain count of a function object.

```
void CGFunctionRelease (
   CGFunctionRef function
);
```

Parameters

`function`

The function object to release.

Discussion

This function is equivalent to `CFRelease`, except that it does not cause an error if the `function` parameter is `NULL`.

Availability

Available in Mac OS X version 10.2 and later.

Declared In

`CGFunction.h`

CGFunctionRetain

Increments the retain count of a function object.

```
CGFunctionRef CGFunctionRetain (
   CGFunctionRef function
);
```

Parameters

`function`

The same function object you passed in as the `function` parameter.

Return Value

Discussion

This function is equivalent to `CFRetain`, except that it does not cause an error if the `function` parameter is `NULL`.

Availability

Available in Mac OS X version 10.2 and later.

Declared In

`CGFunction.h`

Callbacks

CGFunctionEvaluateCallback

Performs custom operations on the supplied input data to produce output data.

```
typedef void (*CGFunctionEvaluateCallback) (
    void *info,
    const float *inData,
    float *outData
);
```

If you name your function `MyCGFunctionEvaluate`, you would declare it like this:

```
void MyCGFunctionEvaluate (
    void *info,
    const float *inData,
    float *outData
);
```

Parameters

info

The `info` parameter passed to `CGFunctionCreate` (page 6).

inData

An array of floats. The size of the array is that specified by the `domainDimension` parameter passed to the `CGFunctionCreate` (page 6) function.

outData

An array of floats. The size of the array is that specified by the `rangeDimension` parameter passed to the `CGFunctionCreate` (page 6) function.

Discussion

The callback you write is responsible for implementing the calculation of output values from the supplied input values. For example, if you want to implement a simple "squaring" function of one input argument to one output argument, your evaluation function might be:

```
void evaluateSquare(void *info, const float *inData, float *outData)
{
    outData[0] = inData[0] * inData[0];
}
```

Availability

Available in Mac OS X v10.2 and later.

Declared In

`CGFunction.h`

CGFunctionReleaseInfoCallback

Performs custom clean-up tasks when Quartz deallocates a CGFunction object.

```
typedef void (*CGFunctionReleaseInfoCallback) (
    void *info
);
```

If you name your function `MyCGFunctionReleaseInfo`, you would declare it like this:

```
void MyCGFunctionReleaseInfo (
    void *info
);
```

Parameters

`info`

The `info` parameter passed to `CGFunctionCreate` (page 6).

Availability

Available in Mac OS X v10.2 and later.

Declared In

`CGFunction.h`

Data Types

CGFunctionRef

An opaque type that represents a callback function.

```
typedef struct CGFunction *CGFunctionRef;
```

Availability

Available in Mac OS X version 10.2 and later.

Declared In

`CGFunction.h`

CGFunctionCallbacks

A structure that contains callbacks needed by a CGFunction object.

```
struct CGFunctionCallbacks
{
  unsigned int version;
  CGFunctionEvaluateCallback evaluate;
  CGFunctionReleaseInfoCallback releaseInfo
};

typedef struct CGFunctionCallbacks CGFunctionCallbacks;
```

Fields

`version`

The structure version number. For this structure, the version should be `0`.

`evaluate`

The callback that evaluates the function.

`releaseInfo`

If non-`NULL`, the callback used to release the `info` parameter passed to `CGFunctionCreate` (page 6).

Availability

Available in Mac OS X v10.2 and later.

Declared In

`CGFunction.h`

Document Revision History

This table describes the changes to *CGFunction Reference*.

Date	Notes
2006-12-22	Made minor editorial changes.
2005-07-07	Added information to CGFunctionCreate.
2005-04-29	Revised introduction.
2004-02-26	First version of this document. An earlier version of this information appeared in *Quartz 2D Reference*.

Index