CGImageDestination Reference

Graphics & Imaging > Quartz

2007-05-04

Apple Inc.
© 2004, 2007 Apple Inc.
All rights reserved.

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

CGImageDestination Reference

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

Overview

CGImageDestination objects, available in Mac OS X v10.4 or later, abstract the data-writing task. An image destination can represent a single image or multiple images. It can contain thumbnail images as well as properties for each image.

The functions described in this reference can write data to three kinds of destinations: a URL, a CFData object, and a data consumer. After creating a CGImageDestination object for the appropriate destination, you can add image data and set image properties. When you are finished adding data, call the function CGImageDestinationFinalize to write the image data and properties to the URL, CFData object, or data consumer.

Functions by Task

Creating Image Destinations

CGImageDestinationCreateWithDataConsumer (page 8)
Creates an image destination that writes to the specified data consumer.

CGImageDestinationCreateWithData (page 8)
Creates an image destination that writes to a Core Foundation mutable data object.

CGImageDestinationCreateWithURL (page 9)
Creates an image destination that writes to a location specified by a URL.

Adding Images

CGImageDestinationAddImage (page 6)
Adds an image to an image destination.

CGImageDestinationAddImageFromSource (page 7)
Adds an image from an image source to an image destination.

Getting Type Identifiers

`CGImageDestinationCopyTypeIdentifiers` (page 7)
: Returns an array of the uniform type identifiers (UTIs) that are supported for image destinations.

`CGImageDestinationGetTypeID` (page 10)
: Returns the unique type identifier of an image destination opaque type.

Setting Properties

`CGImageDestinationSetProperties` (page 10)
: Applies one or more properties to all images in an image destination.

Finalizing an Image Destination

`CGImageDestinationFinalize` (page 9)
: Writes image data and properties to the data, URL, or data consumer associated with the image destination.

Functions

CGImageDestinationAddImage

Adds an image to an image destination.

```
void CGImageDestinationAddImage (
    CGImageDestinationRef idst,
    CGImageRef image,
    CFDictionaryRef properties
);
```

Parameters

`idst`
: An image destination

`image`
: The image to add.

`properties`
: An optional dictionary that specifies the properties of the added image. The dictionary can contain any of the properties described in "`Destination Properties`" (page 11) or the image properties described in *CGImageProperties Reference*.

Discussion

The function logs an error if you add more images than what you specified when you created the image destination.

Availability

Available in Mac OS X version 10.4 and later.

Declared In
`CGImageDestination.h`

CGImageDestinationAddImageFromSource

Adds an image from an image source to an image destination.

```
void CGImageDestinationAddImageFromSource (
    CGImageDestinationRef idst,
    CGImageSourceRef isrc,
    size_t index,
    CFDictionaryRef properties
);
```

Parameters

`idst`

An image destination.

`isrc`

An image source.

`index`

An index that specifies the location of the image in the image source. The index is zero-based.

`properties`

A dictionary that specifies properties to overwrite or add to the source image properties. If a key in `properties` has the value `kCFNull`, the corresponding property in the image destination is removed. The dictionary can contain any of the properties described in "`Destination Properties`" (page 11) or the image properties described in *CGImageProperties Reference*.

Availability
Available in Mac OS X version 10.4 and later.

Declared In
`CGImageDestination.h`

CGImageDestinationCopyTypeIdentifiers

Returns an array of the uniform type identifiers (UTIs) that are supported for image destinations.

```
CFArrayRef CGImageDestinationCopyTypeIdentifiers (
    void
);
```

Return Value
Returns an array of the UTIs that are supported for image destinations. See Uniform Type Identifiers Overview for a list of system-declared and third-party UTIs that can be returned.

Availability
Available in Mac OS X version 10.4 and later.

Declared In
`CGImageDestination.h`

CGImageDestinationCreateWithData

Creates an image destination that writes to a Core Foundation mutable data object.

```
CGImageDestinationRef CGImageDestinationCreateWithData (
    CFMutableDataRef data,
    CFStringRef type,
    size_t count,
    CFDictionaryRef options
);
```

Parameters

data

The data object to write to. For more information on data objects, see *CFData Reference* and Data Objects.

type

The uniform type identifier (UTI) of the resulting image file. See *Uniform Type Identifiers Overview* for a list of system-declared and third-party UTIs.

count

The number of images (not including thumbnail images) that the image file will contain.

options

Reserved for future use. Pass `NULL`.

Return Value

An image destination. You are responsible for releasing this object using `CFRelease`.

Availability

Available in Mac OS X version 10.4 and later.

Declared In

`CGImageDestination.h`

CGImageDestinationCreateWithDataConsumer

Creates an image destination that writes to the specified data consumer.

```
CGImageDestinationRef CGImageDestinationCreateWithDataConsumer (
    CGDataConsumerRef consumer,
    CFStringRef type,
    size_t count,
    CFDictionaryRef options
);
```

Parameters

consumer

The data consumer to write to. For information on data consumers see *CGDataConsumer Reference* and *Quartz 2D Programming Guide*.

type

The uniform type identifier (UTI) of the resulting image file. See *Uniform Type Identifiers Overview* for a list of system-declared and third-party UTIs.

count

The number of images (not including thumbnail images) that the image file will contain.

options

Reserved for future use. Pass `NULL`.

Return Value

An image destination. You are responsible for releasing this object using `CFRelease`.

Availability

Available in Mac OS X version 10.4 and later.

Declared In

`CGImageDestination.h`

CGImageDestinationCreateWithURL

Creates an image destination that writes to a location specified by a URL.

```
CGImageDestinationRef CGImageDestinationCreateWithURL (
    CFURLRef url,
    CFStringRef type,
    size_t count,
    CFDictionaryRef options
);
```

Parameters

url

The URL to write to. If the URL already exists, the data at this location is overwritten.

type

The UTI (uniform type identifier) of the resulting image file. See *Uniform Type Identifiers Overview* for a list of system-declared and third-party UTIs.

count

The number of images (not including thumbnail images) that the image file will contain.

options

Reserved for future use. Pass `NULL`.

Return Value

An image destination. You are responsible for releasing this object using `CFRelease`.

Availability

Available in Mac OS X version 10.4 and later.

Declared In

`CGImageDestination.h`

CGImageDestinationFinalize

Writes image data and properties to the data, URL, or data consumer associated with the image destination.

```
bool CGImageDestinationFinalize (
   CGImageDestinationRef idst
);
```

Parameters

idst

An image destination.

Return Value

Returns `true` if the image is successfully written; `false` otherwise.

Discussion

You must call this function or the output of the image destination will not be valid. After calling this function, no additional data can be added to the image destination.

Availability

Available in Mac OS X version 10.4 and later.

Declared In

`CGImageDestination.h`

CGImageDestinationGetTypeID

Returns the unique type identifier of an image destination opaque type.

```
CFTypeID CGImageDestinationGetTypeID (
    void
);
```

Return Value

Returns the Core Foundation type ID for an image destination.

Discussion

A type identifier is an integer that identifies the opaque type to which a Core Foundation object belongs. You use type IDs in various contexts, such as when you are operating on heterogeneous collections.

Availability

Available in Mac OS X version 10.4 and later.

Declared In

`CGImageDestination.h`

CGImageDestinationSetProperties

Applies one or more properties to all images in an image destination.

```
void CGImageDestinationSetProperties (
    CGImageDestinationRef idst,
    CFDictionaryRef properties
);
```

Parameters

idst

An image destination.

properties

A dictionary that contains the properties to apply. You can set any of the properties described in "Destination Properties" (page 11) or the image properties described in *CGImageProperties Reference*.

Availability
Available in Mac OS X version 10.4 and later.

Declared In
`CGImageDestination.h`

Data Types

CGImageDestinationRef

An opaque type that represents an image destination.

```
typedef struct CGImageDestination *CGImageDestinationRef;
```

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CGImageDestination.h`

Constants

Destination Properties

Properties for a single image in an image destination.

```
const CFStringRef kCGImageDestinationLossyCompressionQuality
const CFStringRef kCGImageDestinationBackgroundColor
```

Constants

`kCGImageDestinationLossyCompressionQuality`

The desired compression quality to use when writing to an image destination. If present, the value associated with this key must be a `CFNumberRef` data type in the range `0.0` to `1.0`. A value of `1.0` specifies to use lossless compression if destination format supports it. A value of 0.0 implies to use maximum compression.

Available in Mac OS X v10.4 and later.

Declared in `CGImageDestination.h`.

`kCGImageDestinationBackgroundColor`

The desired background color to composite against when writing an image that has an alpha component to a destination format that does not support alpha. If present, the value associated with this key must be a `CGColorRef` data type without an alpha component of its own. If not present, and if a background color is needed, a white color is used.

Available in Mac OS X v10.4 and later.

Declared in `CGImageDestination.h`.

Declared In

`CGImageDestination.h`

Document Revision History

This table describes the changes to *CGImageDestination Reference*.

Date	Notes
2007-05-04	Made minor technical and editoral corrections.
2005-04-29	First version.

Index