
CGDataConsumer Reference

[Graphics & Imaging](#) > Quartz



2006-12-22



Apple Inc.
© 2003, 2006 Apple Computer, Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

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

Apple, the Apple logo, Mac, Mac OS, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

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

CGDataConsumer Reference 5

Overview	5
Functions by Task	5
Creating Data Consumers	5
Getting the CType ID	5
Retaining and Releasing Data Consumers	6
Functions	6
CGDataConsumerCreate	6
CGDataConsumerCreateWithCFData	6
CGDataConsumerCreateWithURL	7
CGDataConsumerGetTypeID	7
CGDataConsumerRelease	8
CGDataConsumerRetain	8
Callbacks	9
CGDataConsumerPutBytesCallback	9
CGDataConsumerReleaseInfoCallback	10
Data Types	10
CGDataConsumerCallbacks	10
CGDataConsumerRef	11

Document Revision History 13

Index 15

CGDataConsumer Reference

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

Overview

The `CGDataConsumerRef` opaque type abstracts the data-writing task and eliminates the need for applications to manage data through a raw memory buffer. You can use data consumer objects to write image or PDF data and all, except for [CGDataConsumerCreateWithCFData](#) (page 6), are available in Mac OS X v10.0 or later.

If your application runs in Mac OS X v10.4 or later, you should use `CGImageDestination` objects rather than data consumers. See *CGImageDestination Reference*.

Functions by Task

Creating Data Consumers

[CGDataConsumerCreate](#) (page 6)

Creates a data consumer that uses callback functions to write data.

[CGDataConsumerCreateWithURL](#) (page 7)

Creates a data consumer that writes data to a location specified by a URL.

[CGDataConsumerCreateWithCFData](#) (page 6)

Creates a data consumer that writes to a CFData object.

Getting the CType ID

[CGDataConsumerGetTypeID](#) (page 7)

Returns the Core Foundation type identifier for Quartz data consumers.

Retaining and Releasing Data Consumers

[CGDataConsumerRelease](#) (page 8)

Decrements the retain count of a data consumer.

[CGDataConsumerRetain](#) (page 8)

Increments the retain count of a data consumer.

Functions

CGDataConsumerCreate

Creates a data consumer that uses callback functions to write data.

```
CGDataConsumerRef CGDataConsumerCreate (
    void *info,
    const CGDataConsumerCallbacks *callbacks
);
```

Parameters

info

A pointer to data of any type or NULL. When Quartz calls the functions specified in the `callbacks` parameter, it passes this pointer as the `info` parameter.

callbacks

A pointer to a `CGDataConsumerCallbacks` structure that specifies the callback functions you implement to copy data sent to the consumer and to handle the consumer's basic memory management. For a complete description, see [CGDataConsumerCallbacks](#) (page 10).

Return Value

A new data consumer object. You are responsible for releasing this object using [CGDataConsumerRelease](#) (page 8).

Availability

Available in Mac OS X version 10.0 and later.

Related Sample Code

CarbonSketch

Declared In

`CGDataConsumer.h`

CGDataConsumerCreateWithCFData

Creates a data consumer that writes to a CFData object.

```
CGDataConsumerRef CGDataConsumerCreateWithCFData (
    CFMutableDataRef data
);
```

Parameters*data*

The CFData object to write to.

Return Value

A new data consumer object. You are responsible for releasing this object using [CGDataConsumerRelease](#) (page 8).

Discussion

You can use this function when you need to represent Quartz data as a CFData type. For example, you might create a CFData object that you then copy to the pasteboard.

Availability

Available in Mac OS X v10.4 and later.

Declared In

CGDataConsumer.h

CGDataConsumerCreateWithURL

Creates a data consumer that writes data to a location specified by a URL.

```
CGDataConsumerRef CGDataConsumerCreateWithURL (
    CFURLRef url
);
```

Parameters*url*

A CFURL object that specifies the data destination.

Return Value

A new data consumer object. You are responsible for releasing this object using [CGDataConsumerRelease](#) (page 8).

Availability

Available in Mac OS X version 10.0 and later.

Declared In

CGDataConsumer.h

CGDataConsumerGetTypeID

Returns the Core Foundation type identifier for Quartz data consumers.

```
CFTypeID CGDataConsumerGetTypeID (
    void
);
```

Return Value

The Core Foundation identifier for the opaque type [CGDataConsumerRef](#) (page 11).

Availability

Available in Mac OS X version 10.2 and later.

Declared In

CGDataConsumer.h

CGDataConsumerRelease

Decrements the retain count of a data consumer.

```
void CGDataConsumerRelease (  
    CGDataConsumerRef consumer  
);
```

Parameters

consumer

The data consumer to release.

Discussion

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

Availability

Available in Mac OS X version 10.0 and later.

Related Sample Code

CarbonSketch

Declared In

CGDataConsumer.h

CGDataConsumerRetain

Increments the retain count of a data consumer.

```
CGDataConsumerRef CGDataConsumerRetain (  
    CGDataConsumerRef consumer  
);
```

Parameters

consumer

The data consumer to retain.

Return Value

The same data consumer you passed in as the `consumer` parameter.

Discussion

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

Availability

Available in Mac OS X version 10.0 and later.

Declared In

CGDataConsumer.h

Callbacks

CGDataConsumerPutBytesCallback

Copies data from a Quartz-supplied buffer into a data consumer.

```
size_t (*CGDataConsumerPutBytesCallback) (
    void *info,
    const void *buffer,
    size_t count
);
```

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

```
size_t MyConsumerPutBytes (
    void *info,
    const void *buffer,
    size_t count
);
```

Parameters*info*A generic pointer to private data shared among your callback functions. This is the pointer supplied to [CGDataConsumerCreate](#) (page 6).*buffer*

The Quartz-supplied buffer from which you copy the specified number of bytes.

count

The number of bytes to copy.

Return Value

The number of bytes copied. If no more data can be written to the consumer, you should return 0.

DiscussionWhen Quartz is ready to send data to the consumer, your function is called. It should copy the specified number of bytes from *buffer* into some resource under your control—for example, a file.For information on how to associate your callback function with a data consumer, see [CGDataConsumerCreate](#) (page 6) and [CGDataConsumerCallbacks](#) (page 10).**Availability**

Available in Mac OS X v10.4 and later.

Declared In

CGDataConsumer.h

CGDataConsumerReleaseInfoCallback

Releases any private data or resources associated with the data consumer.

```
void (*CGDataConsumerReleaseInfoCallback) (
    void *info
);
```

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

```
void MyConsumerReleaseInfo (
    void *info
);
```

Parameters

info

A generic pointer to private data shared among your callback functions. This is the same pointer you supplied to [CGDataConsumerCreate](#) (page 6).

Discussion

When Quartz frees a data consumer that has an associated release function, the release function is called.

For information on how to associate your callback function with a data consumer, see [CGDataConsumerCreate](#) (page 6) and [CGDataConsumerCallbacks](#) (page 10).

Availability

Available in Mac OS X v10.4 and later.

Declared In

`CGDataConsumer.h`

Data Types

CGDataConsumerCallbacks

A structure that contains pointers to callback functions that manage the copying of data for a data consumer.

```
struct CGDataConsumerCallbacks {
    CGDataConsumerPutBytesCallback putBytes;
    CGDataConsumerReleaseInfoCallback releaseConsumer;
};
typedef struct CGDataConsumerCallbacks CGDataConsumerCallbacks;
```

Fields

`putBytes`

A pointer to a function that copies data to the data consumer. For more information, see [CGDataConsumerPutBytesCallback](#) (page 9).

`releaseConsumer`

A pointer to a function that handles clean-up for the data consumer, or `NULL`. For more information, see [CGDataConsumerReleaseInfoCallback](#) (page 10)

Discussion

The functions specified by the `CGDataConsumerCallbacks` structure are responsible for copying data that Quartz sends to your consumer and for handling the consumer's basic memory management. You supply a `CGDataConsumerCallbacks` structure to the function [CGDataConsumerCreate](#) (page 6) to create a data consumer.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CGDataConsumer.h`

CGDataConsumerRef

An opaque type that handles the storage of data supplied by Quartz functions.

```
typedef struct CGDataConsumer *CGDataConsumerRef;
```

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CGDataConsumer.h`

Document Revision History

This table describes the changes to *CGDataConsumer Reference*.

Date	Notes
2006-12-22	Updated for Mac OS X v10.5.
2005-04-29	Updated for Mac OS X v10.4.
	Changed the name of the <code>releaseInfo</code> field to <code>releaseConsumer</code> and then modified the fields in the data structure <code>CGDataConsumerCallbacks</code> (page 10) to match the field descriptions. Formerly, they showed a callback prototype in the field instead of the callback data type. The usage of this data structure remains the same.
	Added the function <code>CGDataConsumerCreateWithCFData</code> (page 6).
2004-08-31	Added introductory material.
2004-02-26	First version of this document. An earlier version of this information appeared in <i>Quartz 2D Reference</i> .

REVISION HISTORY

Document Revision History

Index

C

CGDataConsumerCallbacks **structure** [10](#)
CGDataConsumerCreate **function** [6](#)
CGDataConsumerCreateWithCFData **function** [6](#)
CGDataConsumerCreateWithURL **function** [7](#)
CGDataConsumerGetTypeID **function** [7](#)
CGDataConsumerPutBytesCallback **callback** [9](#)
CGDataConsumerRef **data type** [11](#)
CGDataConsumerRelease **function** [8](#)
CGDataConsumerReleaseInfoCallback **callback** [10](#)
CGDataConsumerRetain **function** [8](#)