

---

# CGPDFDictionary Reference

Graphics & Animation: 2D Drawing



2008-10-15



Apple Inc.  
© 2003, 2008 Apple 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, iPhone, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

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

## **CGPDFDictionary Reference 5**

---

Overview	5
Functions by Task	5
Applying a Function to All Entries	5
Getting Data from a Dictionary	6
Functions	6
CGPDFDictionaryApplyFunction	6
CGPDFDictionaryGetArray	7
CGPDFDictionaryGetBoolean	8
CGPDFDictionaryGetCount	8
CGPDFDictionaryGetDictionary	8
CGPDFDictionaryGetInteger	9
CGPDFDictionaryGetName	10
CGPDFDictionaryGetNumber	10
CGPDFDictionaryGetObject	11
CGPDFDictionaryGetStream	11
CGPDFDictionaryGetString	12
Callbacks	12
CGPDFDictionaryApplierFunction	12
Data Types	13
CGPDFDictionaryRef	13

## **Document Revision History 15**

---



# CGPDFDictionary Reference

---

<b>Derived From:</b>	None
<b>Framework:</b>	ApplicationServices/ApplicationServices.h
<b>Companion guide</b>	Quartz 2D Programming Guide
<b>Declared in</b>	CGPDFDictionary.h

## Overview

The `CGPDFDictionaryRef` opaque type encapsulates a PDF dictionary whose key-value pairs can specify any kind of PDF object, including another dictionary. Dictionary objects are the main building blocks of a PDF document. A key-value pair within a dictionary is called an entry. In a PDF dictionary, the key must be an array of characters. Within a given dictionary, the keys are unique—that is, no two keys in a single dictionary are equal (as determined by `strcmp`). The value associated with a key can be any kind of PDF object, including another dictionary. Dictionary objects are the main building blocks of a PDF document.

Many functions that retrieve values from a PDF dictionary take the form:

```
bool CGPDFDictionaryGet<DataType> (  
    CGPDFDictionaryRef dictionary,  
    const char *key,  
    <DataType>Ref *value  
);
```

These functions test whether there is an object associated with the specified key. If there is an object associated with the specified key, they test its data type. If there is no associated object, or if there is but it is not of the expected type, the function returns `false`. If there is an object associated with the specified key and it is of the expected type, the function returns `true` and the object is passed back in the `value` parameter.

This opaque type is not derived from `CType` and therefore there are no functions for retaining and releasing it. `CGPDFDictionary` objects exist only as constituent parts of a `CGPDFDocument` object, and they are managed by their container.

## Functions by Task

### Applying a Function to All Entries

[CGPDFDictionaryApplyFunction](#) (page 6)

Applies a function to each entry in a dictionary.

## Getting Data from a Dictionary

[CGPDFDictionaryGetArray](#) (page 7)

Returns whether there is a PDF array associated with a specified key in a PDF dictionary and, if so, retrieves that array.

[CGPDFDictionaryGetBoolean](#) (page 8)

Returns whether there is a PDF Boolean value associated with a specified key in a PDF dictionary and, if so, retrieves the Boolean value.

[CGPDFDictionaryGetCount](#) (page 8)

Returns the number of entries in a PDF dictionary.

[CGPDFDictionaryGetDictionary](#) (page 8)

Returns whether there is another PDF dictionary associated with a specified key in a PDF dictionary and, if so, retrieves that dictionary.

[CGPDFDictionaryGetInteger](#) (page 9)

Returns whether there is a PDF integer associated with a specified key in a PDF dictionary and, if so, retrieves that integer.

[CGPDFDictionaryGetName](#) (page 10)

Returns whether an object with a specified key in a PDF dictionary is a PDF name reference (represented as a constant C string) and, if so, retrieves that name.

[CGPDFDictionaryGetNumber](#) (page 10)

Returns whether there is a PDF number associated with a specified key in a PDF dictionary and, if so, retrieves that number.

[CGPDFDictionaryGetObject](#) (page 11)

Returns whether there is a PDF object associated with a specified key in a PDF dictionary and, if so, retrieves that object.

[CGPDFDictionaryGetStream](#) (page 11)

Returns whether there is a PDF stream associated with a specified key in a PDF dictionary and, if so, retrieves that stream.

[CGPDFDictionaryGetString](#) (page 12)

Returns whether there is a PDF string associated with a specified key in a PDF dictionary and, if so, retrieves that string.

## Functions

### **CGPDFDictionaryApplyFunction**

Applies a function to each entry in a dictionary.

```
void CGPDFDictionaryApplyFunction (
    CGPDFDictionaryRef dict,
    CGPDFDictionaryApplierFunction function,
    void *info
);
```

**Parameters***dictionary*

A PDF dictionary. If this parameter is not a valid PDF dictionary, the behavior is undefined.

*function*

The function to apply to each entry in the dictionary.

*info*

A pointer to contextual information to pass to the function.

**Discussion**

This function enumerates all of the entries in the dictionary, calling the function once for each. The current key, its associated value, and the contextual information are passed to the function (see also [CGPDFDictionaryApplierFunction](#) (page 12)).

**Availability**

Available in Mac OS X version 10.3 and later.

**Declared In**

CGPDFDictionary.h

**CGPDFDictionaryGetArray**

Returns whether there is a PDF array associated with a specified key in a PDF dictionary and, if so, retrieves that array.

```
bool CGPDFDictionaryGetArray (
    CGPDFDictionaryRef dict,
    const char *key,
    CGPDFArrayRef *value
);
```

**Parameters***dictionary*

A PDF dictionary. If this parameter is not a valid PDF dictionary, the behavior is undefined.

*key*

The key for the value to retrieve.

*value*

On input, an uninitialized pointer to a PDF array. If the value associated with the specified key is a PDF array, then on return contains that array; otherwise the value is unspecified.

**Return Value**

Returns `true` if there is a PDF array associated with the specified key; otherwise, `false`.

**Availability**

Available in Mac OS X version 10.3 and later.

**Declared In**

CGPDFDictionary.h

**CGPDFDictionaryGetBoolean**

Returns whether there is a PDF Boolean value associated with a specified key in a PDF dictionary and, if so, retrieves the Boolean value.

```
bool CGPDFDictionaryGetBoolean (
    CGPDFDictionaryRef dict,
    const char *key,
    CGPDFBoolean *value
);
```

**Parameters**

*dictionary*

A PDF dictionary. If this parameter is not a valid PDF dictionary, the behavior is undefined.

*key*

The key for the value to retrieve.

*value*

On input, a pointer to a PDF Boolean value. If the value associated with the specified key is a PDF Boolean value, then on return contains that value; otherwise the value is unspecified.

**Return Value**

Returns `true` if there is a PDF Boolean value associated with the specified key; otherwise, `false`.

**Availability**

Available in Mac OS X version 10.3 and later.

**Declared In**

CGPDFDictionary.h

**CGPDFDictionaryGetCount**

Returns the number of entries in a PDF dictionary.

```
size_t CGPDFDictionaryGetCount (
    CGPDFDictionaryRef dict
);
```

**Parameters**

*dictionary*

A PDF dictionary. If this parameter is not a valid PDF dictionary, the behavior is undefined.

**Return Value**

Returns the number of entries in the dictionary.

**Availability**

Available in Mac OS X version 10.3 and later.

**Declared In**

CGPDFDictionary.h

**CGPDFDictionaryGetDictionary**

Returns whether there is another PDF dictionary associated with a specified key in a PDF dictionary and, if so, retrieves that dictionary.



```
bool CGPDFDictionaryGetDictionary (
    CGPDFDictionaryRef dict,
    const char *key,
    CGPDFDictionaryRef *value
);
```

**Parameters***dictionary*

A PDF dictionary. If this parameter is not a valid PDF dictionary, the behavior is undefined.

*key*

The key for the value to retrieve.

*value*

On input, a pointer to a PDF dictionary. If the value associated with the specified key is a PDF dictionary, then on return contains that dictionary; otherwise the value is unspecified.

**Return Value**

Returns `true` if there is a PDF dictionary associated with the specified key; otherwise, `false`.

**Availability**

Available in Mac OS X version 10.3 and later.

**Declared In**

CGPDFDictionary.h

**CGPDFDictionaryGetInteger**

Returns whether there is a PDF integer associated with a specified key in a PDF dictionary and, if so, retrieves that integer.

```
bool CGPDFDictionaryGetInteger (
    CGPDFDictionaryRef dict,
    const char *key,
    CGPDFInteger *value
);
```

**Parameters***dictionary*

A PDF dictionary. If this parameter is not a valid PDF dictionary, the behavior is undefined.

*key*

The key for the value to retrieve.

*value*

On input, a pointer to a PDF integer. If the value associated with the specified key is a PDF integer, then on return contains that value; otherwise the value is unspecified.

**Return Value**

Returns `true` if there is a PDF integer associated with the specified key; otherwise, `false`.

**Availability**

Available in Mac OS X version 10.3 and later.

**Declared In**

CGPDFDictionary.h

## CGPDFDictionaryGetName

Returns whether an object with a specified key in a PDF dictionary is a PDF name reference (represented as a constant C string) and, if so, retrieves that name.

```
bool CGPDFDictionaryGetName (
    CGPDFDictionaryRef dict,
    const char *key,
    const char **value
);
```

### Parameters

*dictionary*

A PDF dictionary. If this parameter is not a valid PDF dictionary, the behavior is undefined.

*key*

The key for the value to retrieve.

*value*

On input, a pointer to a PDF name reference, represented as a constant C string. If the value associated with the specified key is a reference to a PDF name, then on return, the variable points to the name; otherwise, the value is undefined.

### Return Value

Returns `true` if there is a character array associated with the specified key; otherwise, `false`.

### Availability

Available in Mac OS X version 10.3 and later.

### Declared In

CGPDFDictionary.h

## CGPDFDictionaryGetNumber

Returns whether there is a PDF number associated with a specified key in a PDF dictionary and, if so, retrieves that number.

```
bool CGPDFDictionaryGetNumber (
    CGPDFDictionaryRef dict,
    const char *key,
    CGPDFReal *value
);
```

### Parameters

*dictionary*

A PDF dictionary. If this parameter is not a valid PDF dictionary, the behavior is undefined.

*key*

The key for the value to retrieve.

*value*

On input, a pointer to a PDF number. If the value associated with the specified key is a PDF number (real or integer), then on return contains that value; otherwise the value is unspecified.

### Return Value

Returns `true` if there is a PDF number associated with the specified key; otherwise, `false`.

**Availability**

Available in Mac OS X version 10.3 and later.

**Declared In**

CGPDFDictionary.h

**CGPDFDictionaryGetObject**

Returns whether there is a PDF object associated with a specified key in a PDF dictionary and, if so, retrieves that object.

```
bool CGPDFDictionaryGetObject (
    CGPDFDictionaryRef dict,
    const char *key,
    CGPDFObjectRef *value
);
```

**Parameters**

*dictionary*

A PDF dictionary. If this parameter is not a valid PDF dictionary, the behavior is undefined.

*key*

The key for the value to retrieve.

*value*

On input, a pointer to a PDF object. If the value associated with the specified key is a PDF object, then on return contains that object; otherwise the value is unspecified.

**Return Value**

Returns `true` if there is a PDF object associated with the specified key; otherwise, `false`.

**Availability**

Available in Mac OS X version 10.3 and later.

**Declared In**

CGPDFDictionary.h

**CGPDFDictionaryGetStream**

Returns whether there is a PDF stream associated with a specified key in a PDF dictionary and, if so, retrieves that stream.

```
bool CGPDFDictionaryGetStream (
    CGPDFDictionaryRef dict,
    const char *key,
    CGPDFStreamRef *value
);
```

**Parameters**

*dictionary*

A PDF dictionary. If this parameter is not a valid PDF dictionary, the behavior is undefined.

*key*

The key for the value to be retrieved.

*value*

On input, a pointer to a PDF stream. If the value associated with the specified key is a PDF stream, then on return contains that stream; otherwise, the value is unspecified.

**Return Value**

Returns `true` if there is a PDF stream associated with the specified key; otherwise, `false`.

**Availability**

Available in Mac OS X version 10.3 and later.

**Declared In**

CGPDFDictionary.h

## CGPDFDictionaryGetString

Returns whether there is a PDF string associated with a specified key in a PDF dictionary and, if so, retrieves that string.

```
bool CGPDFDictionaryGetString (
    CGPDFDictionaryRef dict,
    const char *key,
    CGPDFStringRef *value
);
```

**Parameters**

*dictionary*

A PDF dictionary. If this parameter is not a valid PDF dictionary, the behavior is undefined.

*key*

The key for the value to retrieve.

*value*

On input, a pointer to a PDF string. If the value associated with the specified key is a PDF string, then on return contains that string; otherwise the value is unspecified.

**Return Value**

Returns `true` if there is a PDF string associated with the specified key; otherwise, `false`.

**Availability**

Available in Mac OS X version 10.3 and later.

**Declared In**

CGPDFDictionary.h

## Callbacks

### CGPDFDictionaryApplierFunction

Performs custom processing on a key-value pair from a PDF dictionary, using optional contextual information.

```
typedef void (*CGPDFDictionaryApplierFunction) (
    const char *key,
    CGPDFObjectRef value,
    void *info,
);
```

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

```
void MyFunction (
    const char *key,
    CGPDFObjectRef object,
    void *info
);
```

### Parameters

*key*

The current key in the dictionary.

*object*

The value in the dictionary associated with the key.

*info*

The contextual information that you provided in the `info` parameter in [CGPDFDictionaryApplyFunction](#) (page 6).

### Discussion

`CGPDFDictionaryApplierFunction` defines the callback for `CGPDFDictionaryApplyFunction`, that enumerates all of the entries in the dictionary, calling your custom applier function once for each entry. The current key, its associated value, and the contextual information are passed to your applier function using the `key`, `value`, and `info` parameters respectively.

### Availability

Available in Mac OS X v10.3 and later.

### Declared In

`CGPDFDictionary.h`

## Data Types

### CGPDFDictionaryRef

An opaque type that encapsulates a PDF dictionary.

```
typedef struct CGPDFDictionary *CGPDFDictionaryRef;
```

### Availability

Available in Mac OS X v10.3 and later.

### Declared In

`CGPDFDictionary.h`



# Document Revision History

---

This table describes the changes to *CGPDFDictionary Reference*.

Date	Notes
2008-10-15	Updated the note about not retaining and releasing this object.
2006-12-22	Made minor editorial changes.
2006-04-04	Updated the Introduction.
2005-04-29	Revised introduction.
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