

---

# QCCompositionRenderer Protocol Reference

Graphics & Animation: 2D Drawing



2008-04-08



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

OpenGL is a registered trademark of Silicon Graphics, Inc.

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

## **QCCompositionRenderer Protocol Reference 5**

---

Overview	5
Tasks	5
Passing and Retrieving Values From a Composition	5
Getting Input and Output Keys	5
Getting Attributes	6
Storing Arbitrary Information	6
Saving and Restoring Input Values	6
Instance Methods	6
attributes	6
inputKeys	7
outputKeys	7
propertyListFromInputValues	7
setInputValuesWithPropertyList:	8
setValue:forInputKey:	8
userInfo	9
valueForInputKey:	9
valueForOutputKey:	10
valueForOutputKey:ofType:	10

## **Document Revision History 13**

---

## **Index 15**

---



# QCCompositionRenderer Protocol Reference

---

<b>Adopted by</b>	QCRenderer QCView QCCompositionLayer
<b>Framework</b>	/System/Library/Frameworks/Quartz.framework/Frameworks/QuartzComposer.framework
<b>Availability</b>	Available in Mac OS X v10.5 and later.
<b>Declared in</b>	QCRenderer.h

## Overview

The `QCRenderer` protocol defines the methods used to pass data to the input ports or retrieve data from the output ports of the root patch of a Quartz Composer composition. This protocol is adopted by the `QCRenderer`, `QCView`, and `QCCompositionLayer` classes.

## Tasks

### Passing and Retrieving Values From a Composition

- [setValue:forInputKey:](#) (page 8) *required method*  
Sets the value for an input port of a composition. (required)
- [valueForInputKey:](#) (page 9) *required method*  
Returns the value for an input port of a composition. (required)
- [valueForOutputKey:](#) (page 10) *required method*  
Returns the value for an output port of a composition. (required)
- [valueForOutputKey:ofType:](#) (page 10) *required method*  
Returns the current value on an output port (identified by its key) of the root patch of the composition. (required)

### Getting Input and Output Keys

- [inputKeys](#) (page 7) *required method*  
Returns an array that contains the keys that identify the input ports of the root patch of the composition. (required)

- [outputKeys](#) (page 7) *required method*  
Returns an array that contains the keys that identify the output ports of the root patch of the composition. (required)

## Getting Attributes

- [attributes](#) (page 6) *required method*  
Returns the attributes of the composition associated with the renderer. (required)

## Storing Arbitrary Information

- [userInfo](#) (page 9) *required method*  
Returns a mutable dictionary for storing arbitrary information. (required)

## Saving and Restoring Input Values

- [propertyListFromInputValues](#) (page 7) *required method*  
Returns a property list object that represents the current values for all the input keys of the composition. (required)
- [setInputValuesWithPropertyList:](#) (page 8) *required method*  
Sets the values for the input keys of the composition from a previously saved property list. (required)

# Instance Methods

## attributes

Returns the attributes of the composition associated with the renderer. (required)

- (NSDictionary \*)attributes

### Return Value

A dictionary that contains the attributes that describe the composition, including the input and output ports of the root patch.

### Discussion

The dictionary can define any of the attributes that are specified by the composition attribute keys. See [QCCompositionAttributeNameKey](#), [QCCompositionAttributeDescriptionKey](#), and [QCCompositionAttributeCopyrightKey](#).

The dictionary can also contain dictionaries that correspond to the keys that identify the input and output ports of the root patch of the composition. See [QCPortAttributeTypeKey](#), [QCPortAttributeNameKey](#), [QCPortAttributeMinimumValueKey](#), [QCPortAttributeMaximumValueKey](#), and [QCPortAttributeMenuItemKey](#).

### Availability

Available in Mac OS X v10.4 and later.

**See Also**

- [inputKeys](#) (page 7)
- [outputKeys](#) (page 7)

**Declared In**

QCRenderer.h

## inputKeys

Returns an array that contains the keys that identify the input ports of the root patch of the composition.  
(required)

```
- (NSArray *)inputKeys
```

**Return Value**

An array of keys associated with input ports.

**Availability**

Available in Mac OS X v10.4 and later.

**See Also**

- [outputKeys](#) (page 7)

**Declared In**

QCRenderer.h

## outputKeys

Returns an array that contains the keys that identify the output ports of the root patch of the composition.  
(required)

```
- (NSArray *)outputKeys
```

**Return Value**

An array of keys associated with input ports.

**Availability**

Available in Mac OS X v10.4 and later.

**See Also**

- [inputKeys](#) (page 7)

**Declared In**

QCRenderer.h

## propertyListFromInputValues

Returns a property list object that represents the current values for all the input keys of the composition.  
(required)

```
- (id)propertyListFromInputValues
```

**Return Value**

A property list object.

**Discussion**

This is a convenience method that allows you to easily save the set of input values on a composition. Typically, you store the set of values in application preferences.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

[setInputValuesWithPropertyList:](#) (page 8)

**Declared In**

QCRenderer.h

**setInputValuesWithPropertyList:**

Sets the values for the input keys of the composition from a previously saved property list. (required)

```
- (void) setInputValuesWithPropertyList:(id)plist
```

**Discussion**

This is a convenience method that allows you to restore the set of input values that you obtained previously by calling the method [propertyListFromInputValues](#) (page 7). If the property list object does not define a value for an input key, or if the value is not of the proper type, Quartz Composer does not set a value for the corresponding input port.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

QCRenderer.h

**setValue:forInputKey:**

Sets the value for an input port of a composition. (required)

```
- (BOOL)setValue:(id)value forInputKey:(NSString *)key
```

**Parameters**

*value*

The value to set for the input port. The input port must be at the root patch of the composition. The data type of the *value* argument must match the input port. See [QCPortAttributeTypeKey](#) for the data types accepted by a particular port type.

*key*

The key associated with the input port of the composition. This method throws an exception if *key* is invalid.

**Return Value**

Returns NO if it cannot set the value.

**Availability**

Available in Mac OS X v10.4 and later.

**See Also**

- [valueForInputKey:](#) (page 9)
- [valueForOutputKey:](#) (page 10)

**Related Sample Code**

TrackBall

**Declared In**

QCRenderer.h

**userInfo**

Returns a mutable dictionary for storing arbitrary information. (required)

- (NSMutableDictionary\*) userInfo

**Return Value**

A mutable dictionary.

**Discussion**

The `userInfo` dictionary is shared—there is one per Quartz Composer context. In fact, it is the same dictionary as the one available for the plug-in execution context for instances of the `QCPlugIn` class.

When you add information to the dictionary, make sure that you use unique keys, such as `"com.myCompany.foo"`.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

QCRenderer.h

**valueForInputKey:**

Returns the value for an input port of a composition. (required)

- (id)valueForInputKey:(NSString \*)key

**Parameters**

*key*

The key associated with an input port for the root patch of a composition. This method throws an exception if `key` is invalid.

**Return Value**

The value. The data type of returned value depends on the type of the input port. See `QCPortAttributeTypeKey` for more information.

**Availability**

Available in Mac OS X v10.4 and later.

**See Also**

- [setValue:forInputKey:](#) (page 8)
- [valueForOutputKey:](#) (page 10)

**Declared In**

QCRenderer.h

**valueForOutputKey:**

Returns the value for an output port of a composition. (required)

- (id)valueForOutputKey:(NSString \*)key

**Parameters***key*

The key associated with an output port for the root patch of a composition. This method throws an exception if *key* is invalid.

**Return Value**

The value. The data type of returned value depends on the type of the output port. See [QCPortAttributeTypeKey](#) for more information.

**Availability**

Available in Mac OS X v10.4 and later.

**See Also**

- [setValue:forInputKey:](#) (page 8)
- [valueForInputKey:](#) (page 9)

**Declared In**

QCRenderer.h

**valueForOutputKey:ofType:**

Returns the current value on an output port (identified by its key) of the root patch of the composition. (required)

- (id) valueForOutputKey:(NSString\*)key ofType:(NSString\*)type

**Parameters***key*

The key associated with an output port for the root patch of a composition. This method throws an exception if *key* is invalid.

*type*

A string that specifies the class.

**Return Value**

The value.

**Discussion**

The value type depends on the type of the port type, as shown in the following table

Port type	Value type
Boolean, Index, or Number	NSNumber or any object that responds to the methods integerValue, floatValue, or doubleValue
String	NSString or any object that responds to the methods stringValue or description
Color	NSColor, UIColor, or CGColor object
Image	UIImage, NSBitmapImageRep, CGImage object, CIImage, CVPixelBuffer object, CVOpenGLBuffer object, CVOpenGLTexture object, or an opaque QImage (that is, an optimized abstract image object only to be used with setValue: forKey: of another <QCCompositionRenderer>)
Structure	NSArray or NSDictionary

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [setValue:forInputKey:](#) (page 8)
- [valueForKey:](#) (page 9)

**Declared In**

QCRenderer.h



# Document Revision History

---

This table describes the changes to *QCCompositionRenderer Protocol Reference*.

Date	Notes
2008-04-08	Fixed minor typographical error.
2007-05-09	New document that describes the methods for passing and retrieving data from the ports of a composition.

## REVISION HISTORY

### Document Revision History

# Index

---

## A

---

attributes [protocol instance method 6](#)

## I

---

inputKeys [protocol instance method 7](#)

## O

---

outputKeys [protocol instance method 7](#)

## P

---

propertyListFromInputValues [protocol instance method 7](#)

## S

---

setInputValuesWithPropertyList: [protocol instance method 8](#)

setValue:forInputKey: [protocol instance method 8](#)

## U

---

userInfo [protocol instance method 9](#)

## V

---

valueForInputKey: [protocol instance method 9](#)

valueForOutputKey: [protocol instance method 10](#)

valueForOutputKey:ofType: [protocol instance method 10](#)