
Quartz Composer WebKit Plug-in JavaScript Reference

[Internet & Web](#) > [Web Content](#)



2009-01-06



Apple Inc.
© 2009 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, Mac, Mac OS, Quartz, and Safari are trademarks of Apple Inc., registered in the United States and other countries.

Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. 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

Introduction [Introduction to Quartz Composer WebKit Plug-in JavaScript Reference](#) 5

[Who Should Read This Document?](#) 5
[Organization of This Document](#) 5
[See Also](#) 5

Chapter 1 [Quartz Composer WebKit Plug-in JavaScript Reference](#) 7

[Methods by Task](#) 7
[Methods](#) 8
 [attributes](#) 8
 [copyImageToPasteboard](#) 8
 [getInputValue](#) 8
 [getOutputValue](#) 8
 [inputKeys](#) 9
 [loaded](#) 9
 [outputKeys](#) 9
 [pause](#) 10
 [paused](#) 10
 [play](#) 10
 [playing](#) 11
 [setInputValue](#) 11
 [stop](#) 11

[Document Revision History](#) 13

Introduction to Quartz Composer WebKit Plug-in JavaScript Reference

Quartz Composer is a development tool for processing and rendering graphical data using an easy-to-use visual programming approach and the rich media technologies available in Mac OS X. The Quartz Composer WebKit plug-in, available in Mac OS X v10.4.7 and later, displays Quartz Composer compositions in webpages and Dashboard widgets.

Who Should Read This Document?

Quartz Composer WebKit Plug-in JavaScript Reference is for web content and Dashboard widget creators who want to manipulate a Quartz Composer composition used in a webpage or Dashboard widget via JavaScript. You will find a reference for the JavaScript API available from the Quartz Composer WebKit plug-in.

To learn how to create a composition using Quartz Composer, read *Quartz Composer Programming Guide*.

Note: *Quartz Composer WebKit Plug-in JavaScript Reference* was previously titled *Quartz Composer Web Kit Plug-in Programming Guide*. Earlier versions of the document described how to include and manipulate a Quartz Composer composition in a web page or Dashboard widget. You can now find this information in *Webpages and Widgets*.

Organization of This Document

This document contains the following chapter:

- [“Quartz Composer WebKit Plug-in JavaScript Reference”](#) (page 7) documents the JavaScript API available when using the Quartz Composer WebKit plug-in.

See Also

To learn more about Quartz Composer and creating compositions, along with including a composition in a web page or Dashboard widget, read *Quartz Composer Programming Guide*.

Safari HTML Reference, *Safari CSS Reference*, and *WebKit DOM Programming Topics* provide information on creating content for WebKit-based browsers like Safari.

For more on Dashboard widget creation, read *Dashboard Tutorial* and *Dashboard Programming Topics*.

INTRODUCTION

Introduction to Quartz Composer WebKit Plug-in JavaScript Reference

Quartz Composer WebKit Plug-in JavaScript Reference

The JavaScript API provided by the Quartz Composer WebKit plug-in includes methods for changing a composition's state and manipulating its published inputs and outputs. To learn how to include a Quartz Composer composition in a web page or Dashboard widget, read [Webpages and Widgets](#).

Note: All of the method descriptions described in [“Methods”](#) (page 8) assume that the method is called on the composition object as obtained from the DOM. You can obtain the object by calling `document.getElementById`:

```
var composition = document.getElementById("myComposition");
```

Obtaining and manipulating the composition is discussed further in [Webpages and Widgets](#).

Methods by Task

Composition Metadata

- [“attributes”](#) (page 8)

Composition Playback

- [“pause”](#) (page 10)
- [“play”](#) (page 10)
- [“stop”](#) (page 11)

Composition Status

- [“loaded”](#) (page 9)
- [“paused”](#) (page 10)
- [“playing”](#) (page 11)

Pasteboard Operations

- [“copyImageToPasteboard”](#) (page 8)

Working with Published Values

- [“getInputValue”](#) (page 8)
- [“getOutputValue”](#) (page 8)
- [“inputKeys”](#) (page 9)
- [“outputKeys”](#) (page 9)
- [“setInputValue”](#) (page 11)

Methods

attributes

Returns an array of strings with the composition's attributes.

```
composition.attributes()
```

Discussion

The attributes returned are the composition's metadata, including its name, copyright, and description values, along with information about each port of the composition.

copyImageToPasteboard

Copies the current context of the composition to the pasteboard.

```
composition.copyImageToPasteboard()
```

Discussion

Use this method in conjunction with an `oncopy` handler.

getInputValue

Returns the value of a published input.

```
composition.getInputValue("key")
```

Discussion

The value of `key` is any of the inputs that you published in Quartz Composer.

If the returned value is a color, an object is returned with `red`, `green`, `blue`, and `alpha` properties. Each value is a floating-point number, on a scale from 0 to 1. Call `htmlColor()` on the object to return its HTML hexadecimal color value or `cssColor()` to return its `rgba` value.

No images are returned when calling `getInputValue`.

See Also

["inputKeys"](#) (page 9)

["outputKeys"](#) (page 9)

["setInputValue"](#) (page 11)

["getOutputValue"](#) (page 8)

getOutputValue

Returns the value of a published output.

```
composition.getOutputValue("key")
```


Discussion

The value of `key` is any of the outputs that you published in Quartz Composer.

If the returned value is a color, an object is returned with `red`, `green`, `blue`, and `alpha` properties. Each value is a floating-point number, on a scale from 0 to 1. Call `htmlColor()` on the object to return its HTML hexadecimal color value or `cssColor()` to return its `rgba` value.

No images are returned when calling `getOutputValue`.

See Also

[“outputKeys”](#) (page 9)

[“inputKeys”](#) (page 9)

[“setInputValue”](#) (page 11)

[“getInputValue”](#) (page 8)

inputKeys

Returns an array of strings with the names of a composition’s published inputs.

```
composition.inputKeys()
```

See Also

[“setInputValue”](#) (page 11)

[“getInputValue”](#) (page 8)

[“outputKeys”](#) (page 9)

[“getOutputValue”](#) (page 8)

loaded

Returns whether the composition is loaded.

```
composition.loaded()
```

See Also

[“playing”](#) (page 11)

[“paused”](#) (page 10)

outputKeys

Returns an array of strings with the names of a composition’s published outputs.

```
composition.outputKeys()
```

See Also[“getOutputValue”](#) (page 8)[“inputKeys”](#) (page 9)[“setInputValue”](#) (page 11)[“getInputValue”](#) (page 8)

pause

Pauses playback of a composition.

```
composition.pause()
```

Discussion

If a composition wasn't playing, calling this method does nothing.

See Also[“play”](#) (page 10)[“stop”](#) (page 11)

paused

Returns whether the composition is paused.

```
composition.paused()
```

Discussion

If the composition is stopped (as opposed to paused), `paused` returns `false`.

See Also[“playing”](#) (page 11)[“loaded”](#) (page 9)

play

Begins or pauses playback of a composition.

```
composition.play()
```

Discussion

The `play` method starts playback of a composition if it was stopped or hasn't played yet, pauses playback if the composition is already playing, and resumes playback if the composition was paused.

See Also[“pause”](#) (page 10)[“stop”](#) (page 11)

playing

Returns whether the composition is playing.

```
composition.playing()
```

See Also

[“paused”](#) (page 10)

[“loaded”](#) (page 9)

setInputValue

Gives the composition the provided value for published input.

```
composition.setInputValue("key", value)
```

Discussion

The value of `key` is any of the inputs that you published in Quartz Composer.

If you're setting an input to a color value, you can pass in:

- An object like that returned by [“getInputValue”](#) (page 8) and [“getOutputValue”](#) (page 8), with `red`, `green`, `blue`, and `alpha` properties
- An HTML hexadecimal color value, like `#FFFFFF`
- An HTML color name, like `red` or `green`
- A CSS `rgb` or `rgba` value

If the input value can't be parsed, no change is made. Also, images aren't allowed as an input value.

See Also

[“inputKeys”](#) (page 9)

[“outputKeys”](#) (page 9)

[“getInputValue”](#) (page 8)

[“getOutputValue”](#) (page 8)

stop

Stops playback of a composition.

```
composition.stop()
```

Discussion

The `stop` method releases all resources associated with the composition and clears its visuals from the page. After calling `stop`, you can call `play` to restart the composition.

See Also

[“play”](#) (page 10)

[“pause”](#) (page 10)

Document Revision History

This table describes the changes to *Quartz Composer WebKit Plug-in JavaScript Reference*.

Date	Notes
2009-01-06	Updated the description of the attributes method.
2008-10-15	Corrected typos.
2006-12-05	Retitled document to Quartz Composer Web Kit Plug-in JavaScript Reference. Quartz Composer Web Kit Plug-in usage and HTML attributes moved to Quartz Composer Programming Guide
2006-08-07	New document that explains how to use the Quartz Composer plug-in included with Mac OS X v.10.4.7.

REVISION HISTORY

Document Revision History