
Quartz Composer Release Note

[Graphics & Imaging](#) > Quartz



2009-04-20



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

Newton and Snow Leopard are trademarks of Apple Inc.

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

Chapter 1 **Quartz Composer Release Notes for Mac OS X v10.6 5**

Introduction	5
Performance Improvements	5
Geometry Pipeline	5
Interaction	6
Workflow Improvements	6
Quartz Composer APIs	7
New Patches	8
Updated Patches	9
Security	10

Quartz Composer Release Notes for Mac OS X v10.6

Introduction

This document summarizes the major changes to Quartz Composer in Snow Leopard since the previous version released in Mac OS X v10.5.

Performance Improvements

■ OpenCL

Quartz Composer integrates seamlessly with the Open Computing Language (OpenCL), which is a new technology in Snow Leopard for performing general purpose computation on collections of arbitrary compute devices such as GPUs and CPUs. With Quartz Composer, you can easily leverage OpenCL, using the OpenCL Kernel patch, without having to set up OpenCL or OpenGL. You can quickly visualize the results produced by your OpenCL kernels simply by hovering over an output port or by interpreting the data as an image or geometry using the supplied patches.

■ Idle State Detection

The Quartz Composer engine can now determine which parts of the graph need to be executed, and when they need to be executed, to allow for frames to be rendered on demand. This can greatly decrease CPU usage for certain cases such as user or event driven compositions.

■ DOD/Dirty Region Optimization

The Quartz Composer engine is now able to identify which patches need to execute based on their representation on screen. This further decreases CPU and GPU usage for certain cases such as user or event driven compositions.

■ OpenGL State Change Minimization

The Quartz Composer engine now groups OpenGL states so as to minimize expensive state changes. This is particularly evident within the Iterator patch when drawing a large number of similar objects.

■ QCCompositionLayer

The `QCCompositionLayer` class has been optimized to allow for a large number of layers to exist simultaneously within Core Animation.

Geometry Pipeline

Quartz Composer now has a fully integrated geometry pipeline. This pipeline centers around the opaque `QCMesh` type, which contains information such as vertex locations, vertex colors, normals, texture coordinates and indices. The geometry pipeline integrates seamlessly with the rest of Quartz Composer. For example, geometry can be created and modified directly on the GPU using OpenCL.

■ Mesh Filter

Mesh Filter is a new Composition Repository protocol used to describe compositions that take a mesh as an input and return a mesh as an output. Quartz Composer includes a number of Mesh Filters as part of the system wide Composition Repository. In order to aid in the creation of Mesh Filters, a new template has been added to the Template sheet.

Interaction

The creation of compositions that leverage user interaction has been greatly improved in Snow Leopard. Interaction information such as position and hit detection are provided for the Billboard, Sprite and Mesh Renderer patches using the new Interaction patch. You can route interaction information to control which objects are interactive as well as modify interaction data to create novel interaction experiences.

■ Multitouch Gestures

There is now a setting on the Mouse and Interaction patches to Expose Multitouch Gestures. This option adds outputs for Rotation Angle, Magnify Value, X Swipe and Y Swipe. These outputs provide data only on hardware that supports Multitouch Gestures. Settings are also provided to integrate rotation and magnification.

Workflow Improvements

■ UI Update

The Quartz Composer graph user interface has been redesigned to make you smile.

■ Interactive Placement

Interactive Placement is a new rendering mode that can be found in the Viewer window's toolbar. This mode allows you to drag and drop objects, such as Sprites, Billboards and Mesh Renderers, within a composition.

■ Patch Library

The new Patch Library lets the user view patches in a number of different ways and create their own groups of patches. In addition, the Patch Library includes compositions from the Composition Repository, allowing you to leverage those compositions from your own compositions.

□ Add To Library

To add commonly used groups of patches to the Patch Library select the patches and from the menu. (Choose Editor > Add To Library.) The selected patches will be converted to a Virtual Macro and will be immediately added to the Patch Library.

□ Importing and Exporting Macros

Macros can be imported and exported using the action menu found within the Patch Library.

■ Virtual Macros

A Virtual Macro is a reference to a composition that allows the composition to be used in multiple places, requiring only changes to the original composition when updates are needed. By default, macros added to a composition from the Patch Library are added as Virtual Macros.

- ❑ **Distribution**
Compositions now store a local copy of all Virtual Macros. This allows compositions that include Virtual Macros to be distributed without the need to distribute and install the referenced macros.
 - ❑ **Editing**
The source composition referenced by a Virtual Macro can be edited by selecting the Macro within the Patch Library and selecting Edit from the action menu. Saving changes to the Macro will immediately update open compositions which use the Macro and will update other compositions once opened.
- **Patch Alignment**
Patches can now be aligned vertically by their ports. This allows for the creation of tidier graphs. This option is controlled via the Editor's preferences.
- **Mini Patches**
In order to increase graph readability, some utility patches such as the Input Splitter, Math and Logic patches are now much smaller.
- **Explode Macro**
To aid in graph refactoring, Macro patches can be effectively “unmade” by choosing Explode Macro from the patch's contextual menu. The sub-patches of the macro are brought to the current level and published port connections are recreated.
- **Document Creation**
The Document Creation sheet now uses a Quartz Composer composition. The composition allows the user to preview templates as well as browse tips provided by the Quartz Composer team.
- **Interface Builder**
QCVIEW objects can now be easily backed by a Core Animation layer from within Interface Builder by simply enabling Core Animation for the view.

Quartz Composer APIs

To take advantage of Idle State Optimizations made to the Quartz Composer engine, you can implement the following APIs.

- **QCRenderer:** Your application can now ask a composition if it needs to render.


```
- (NSTimeInterval) nextRecommendedRenderingTimeForTime:(NSTimeInterval)time
                        arguments:(NSDictionary*)arguments;
```
- **QCPlugIn:** Your plug-ins can now indicate when they should be executed next.


```
- (NSTimeInterval) nextRecommendedExecutionTime:(id<QCPlugInContext>)context
                        forTime:(NSTimeInterval)time
                        withArguments:(NSDictionary*)arguments;
```

New Patches

■ 3D Sound Player

This new patch allows for the playback of sound files on disk. These sounds can be positioned in space and pitch shifted. Supported formats include `.wav` and `.aiff`.

■ Mesh Importer

The Mesh Importer is used to import geometry from digital asset exchange (dae) files. By default, the Mesh Importer patch concatenates all meshes in the referenced file into a single mesh. If individual access to meshes is required, the Mesh Importer has a setting that provides a structure of meshes. Note that the Mesh Importer scales geometry to fit within a unit cube by default. This behavior can be toggled within the patch settings.

■ Fast Disc Blur And Fast Box Blur

New Core Image blur filters.

■ Feedback

The new Feedback patch makes it easier to create feedback loops within your compositions by providing dynamically created outputs, which correspond to published outputs at the current level within the graph.

■ Force

The force patch applies a uniform physical force to an object. A Sampling parameter is provided to determine when the force is applied. This patch can be used in conjunction with the Inertia patch to create complex physical behaviors. Note that this must be used within a feedback loop to work properly.

■ Get Mesh Component

The Get Mesh Component patch returns a specified mesh component from a given mesh. This includes vertices, colors, normals or indices.

■ Get Mesh Texture

The Get Mesh Texture patch returns a specified texture from a given mesh.

■ Inertia

The Inertia patch can be used to add physical inertia to objects according to Newton's first law of motion (a body in motion tends to stay in motion). A Sampling parameter is provided to determine when the inertia is added along with a Friction parameter used to adjust the behavior of the physical effect. Note that this must be used within a feedback loop to work properly.

■ Interaction

The new Interaction patch allows access to interaction data from Sprites, Billboards and Mesh Renderers. These objects can be interacted with directly or used as placeholder objects when interacting with other objects. To access interaction data, connect an Interaction patch's interaction port to a Sprite, Billboard or Mesh Renderer's interaction port. Then, connect the desired output ports of the Interaction patch to either the associated ports on the connected Renderer to control it directly or elsewhere within the composition.

■ Iterator

Counter, Pulse, Sample & Hold and Smooth patches now operate per iteration within the Iterator patch.

■ Logic Patch

This patch has been significantly reduced in size.

■ Math Patch

This patch has been significantly reduced in size.

- **Mesh Creator**

The Mesh Creator patch is used to create a mesh object from one or more individual mesh components. These components include vertices, colors, normals, texture coordinates, indices and an associated texture.
- **Mesh Renderer**

The Mesh Renderer is used to render mesh objects. A setting is provided to display the mesh as a wireframe.
- **Mesh Transform**

The Mesh Transform patch applies an affine transformation to a given mesh.
- **OpenCL Kernel**

The OpenCL Kernel patch allows for general purpose parallel programming across CPUs and GPUs. With this patch you can create and modify OpenCL kernels in realtime using a subset of C with parallel extensions. The OpenCL Kernel patch will automatically create input and output ports based on arguments provided to the OpenCL kernel. For information on writing OpenCL kernels please refer to the OpenCL Specification available at <http://khronos.org/opensl>. By default, the OpenCL Kernel patch will automatically determine the Local Work Size, Global Work Size and Output Dimensions for the kernel based on the size of input data. These parameters, along with the Compute Device, Image Pixel Format and Color Space, can be set manually by enabling the “Show Advanced Kernel Settings” within the patch settings. Manually setting the work sizes or output dimensions is required when input arrays differ from one another in size or differ from the size of output arrays. The Advanced Settings also allow for Performance Sampling using the “Sample Now” button to get an average execution time for a given kernel.
- **Set Mesh Component**

The Set Mesh Component patch allows you to specify a mesh component with a given mesh. This includes vertices, colors, normals or indices.
- **Set Mesh Texture**

The Set Mesh Texture patch allows you to specify a texture for a given mesh.

Updated Patches

- **3D Transform: Scale**

The 3D Transform patch now has Scale inputs. This allows geometry to be scaled in each of the X, Y and Z dimensions while maintaining geometric normals. The Rotation origin has been renamed to Origin for use as both the Scale and Rotation pivots.
- **Image With String: Justification**

New justification modes have been added to the Image With String patch. The patch now supports Left, Center, Right and Full justification.
- **Iterator**

Counter, Pulse, Sample & Hold and Smooth patches now operate per iteration within the Iterator patch.
- **Lighting: Shadows**

The lighting patch now supports shadows. These shadows can be turned on or off for each light and can be used to support colors independent of the light color.

The quality of the shadow can be controlled by the quality slider. When the quality is set to 0, antialiasing is not applied. When the quality is set to 1, antialiasing is applied to generate smooth shadows.

- **Line: Width**
The line patch now has a width input.
- **Logic Patch**
This patch has been significantly reduced in size.
- **Math Patch**
This patch has been significantly reduced in size.
- **OSC**
In order to maintain type stability, overflow bundles are no longer coalesced into structures. This prevents `NULL` values from being output when multiple bundles arrive between frames.

Security

- **iTunes Music Visualizers are no longer required to run in Safe Mode.**
This means that music visualizers can now access devices such as Video Input as well as leverage custom Plugins.