
QuickTime Initialization Guide

[QuickTime > Movie Basics](#)



2005-04-08



Apple Inc.
© 2004, 2005 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, and QuickTime 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

Chapter 1	Introduction to QuickTime Initialization Guide	7
------------------	---	----------

Chapter 2	Initializing QuickTime	9
------------------	-------------------------------	----------

Initializing QuickTime Media Layer for Windows 9

Checking the QuickTime Version 10

Initializing a QuickTime Environment 10

Document Revision History	13
----------------------------------	-----------

Listings

Chapter 2 **Initializing QuickTime 9**

- Listing 2-1 Calling InitializeQTML 9
- Listing 2-2 Getting the QuickTime version with Gestalt 10
- Listing 2-3 Initializing QTML and QuickTime for Windows 11

Introduction to QuickTime Initialization Guide

Before you can call QuickTime functions, your application must initialize QuickTime. You may also want to check the version of QuickTime installed on the user's computer to verify that it supports all the features your application uses.

If you are working on a Windows platform, you also need to verify that QuickTime is installed and initialize the QuickTime Media Layer (QTML).

If you are writing a threaded application, and intend to call QuickTime functions from multiple threads, you need to initialize QuickTime for each worker thread explicitly.

This document describes how and when to initialize QuickTime and the QuickTime Media Layer, and how to check to see if QuickTime is installed, as well as what version is installed.

If you are a QuickTime developer writing procedural C to create QuickTime applications or QuickTime components, you should read this document.

Initializing QuickTime

QuickTime must be initialized before applications or components can make calls to the QuickTime Movie Toolbox.

If you are writing a threaded application, and intend to call QuickTime functions from multiple threads, you need to initialize QuickTime for each worker thread explicitly.

Windows applications must initialize the QuickTime Media Layer (QTML), prior to initializing QuickTime itself. This also serves to verify that QuickTime for Windows is installed.

As part of the initialization process, it is usually advisable to check the installed version of QuickTime to ensure that all required features are present on the user's machine. This can also serve as a method to verify that QuickTime is installed prior to initializing it.

Initializing QuickTime Media Layer for Windows

If your application runs on a Windows operating system, you need to verify that QuickTime is installed and initialize the QuickTime Media Layer (QTML). A single call to `InitializeQTML` does both. [Listing 2-1](#) (page 9) shows the call to `InitializeQTML`.

Listing 2-1 Calling InitializeQTML

```
static void QuickTimeQTML (void)
{
    OSErr err;

    err = InitializeQTML(0L);
    CheckError (err, "InitializeQTML error" );
}
```

Use the `InitializeQTML` function to initialize QuickTime before calling any other QuickTime function. It is recommended that you call `InitializeQTML` at the very beginning of your program, in your `WinMain` function, before creating your main window.

WARNING: Do not call `InitializeQTML` from the `DllMain` procedure of a DLL! Instead, call `InitializeQTML` from the same routine that initializes your DLL.

`InitializeQTML` supports flags that allow you to specify how QuickTime should behave. For example, you can tell QuickTime to operate exclusively in full-screen mode, or to use the Windows Graphics Device Interface (GDI) for all drawing, rather than the `DirectDraw` or `DCI` services.

If QuickTime is not installed, the `InitializeQTML` function is not available and an error is returned.

If your application is distributed on a CD, you should include QuickTime installers on the disc and run the Windows installer if QuickTime is not present. (Including the installer requires a license from Apple, but there is currently no fee for such a license. To obtain a license, click the QuickTime link at <http://developer.apple.com/softwarelicensing/agreements/quicktime.html> and follow the instructions there.) Alternatively, you may prompt the user to download QuickTime at <http://www.apple.com/quicktime/download/>.

If you are writing a routine that does not know whether `InitializeQTML` has already been called, call `InitializeQTML` at the beginning of the routine and `TerminateQTML` at the end.

It does no harm to call `InitializeQTML` more than once, as long as each call is nested with a matching call to `TerminateQTML`. If this function has already been called, subsequent calls do nothing except increment a counter. Calls to `TerminateQTML` just decrement the counter (if it is nonzero). Only the first nested call and the last nested call to `TerminateQTML` do any actual work, so there is no penalty for having multiple nested calls.

Important: You should not make this call from a QuickTime component such as an image decompressor; it is provided only for host applications. QuickTime components can assume that QTML has been initialized.

Once the QuickTime Media Layer has been initialized, Windows programmers should follow the same steps that Mac OS programmers follow to test the QuickTime version and initialize QuickTime itself.

Checking the QuickTime Version

Your application may want to check the version of QuickTime installed on the user's system at runtime, to determine if all the needed features are supported. You should do this before initializing QuickTime.

To get the version of QuickTime installed on the user's computer, call the `Gestalt` function (the necessary parts of which are included in QuickTime for Windows). The `Gestalt` function returns the QuickTime version if the appropriate selectors are passed in, as shown in [Listing 2-2](#) (page 10).

Listing 2-2 Getting the QuickTime version with Gestalt

```
{
    /* check the version of QuickTime installed */

    long version;
    OSErr result;

    result = Gestalt(gestaltQuickTime,&version);
    if ((result == noErr) && (version >= 0x05020000))
    {
        /* we have version 5.0.2 or later */
    }
}
```

Initializing a QuickTime Environment

Your application initializes a working QuickTime environment by calling `EnterMovies`.

```
EnterMovies();
```

If you are writing a routine that does not know whether QuickTime has already been initialized, go ahead and call `EnterMovies`; repeated calls do no harm.

You do not need to balance calls to `EnterMovies` with calls to `ExitMovies`. Call `ExitMovies` only if you are done using QuickTime and want to free the resources as soon as possible without terminating your application. `ExitMovies` is called automatically when your application quits.

Windows users need to initialize QTML before calling `EnterMovies`, as shown in [Listing 2-3](#) (page 11).

Listing 2-3 Initializing QTML and QuickTime for Windows

```
static void InitQTMLandQuickTime (void)
{
    OSErr err;

    err = InitializeQTML(OL);
    CheckError (err, "InitializeQTML error" );
    EnterMovies();
}
```

Note: QuickTime components for Windows must call `EnterMovies` if they call any QuickTime functions, even though they do not call `InitializeQTML`.

`EnterMovies` initializes a single, non-reentrant QuickTime environment for your application.

If your application uses QuickTime on multiple threads simultaneously, call `EnterMoviesOnThread` from each thread that uses QuickTime to create a local QuickTime environment for that thread (requires QuickTime 6 or later).

For more information about threaded programming and QuickTime, see Technical Note TN2125, [Thread-safe programming in QuickTime](#).

Document Revision History

This table describes the changes to *QuickTime Initialization Guide*.

Date	Notes
2005-04-08	Fixed formatting problems. Changed title from "Initializing QuickTime."
2004-12-02	Corrected links and minor textual errors.
2004-10-18	TBD

REVISION HISTORY

Document Revision History