# Technical Note TN 2068
Full Screen changes in QuickTime 6.1

CONTENTS

This technote discusses changes made to the QuickTime Full Screen API with the release of QuickTime 6.1

[Feb 21 2003]

## Introduction

The QuickTime Full Screen API consists of two function calls; `BeginFullScreen` and `EndFullScreen`. With the release of QuickTime 6.1 improvements were made to `BeginFullScreen` which enhance performance (allowing QuickTime to switch in and out of full-screen mode very quickly), and bring the API into full compliance with the Mac OS X Aqua Human Interface Guidelines.

This technical note discusses a change made to the CoreGraphics `WindowLevel` of the full-screen blanking window returned from `BeginFullScreen`, and the addition of three new flags which can be used when initiating full-screen mode.

```
// Begins full-screen mode for a specified graphics device.
OSErr BeginFullScreen(Ptr       *restoreState,
                      GDHandle  whichGD,
                      short     *desiredWidth,
                      short     *desiredHeight,
                      WindowRef *newWindow,
                      RGBColor  *eraseColor,
                      long      flags);


// Ends full-screen mode for a graphics device.
OSErr EndFullScreen(Ptr  fullState,
                    long flags);
```

For more information regarding the use of these APIs refer to the QuickTime Full Screen API Reference.

## Window Level Change

The full-screen mode for applications is considered a *special mode*. An example of this is iTunes' full-screen effects visualizer or iPhoto's full-screen slide show. This *special mode* prevents other applications' windows from coming to the forefront while your application's full-screen mode is engaged. When an application enters full-screen mode, it essentially becomes a *one-window* application.

The appropriate window level for full-screen applications on Mac OS X is the ScreenSaver level. This window level is defined as `kCGScreenSaverWindowLevel` which returns the key value using the function call `CGWindowLevelForKey(kCGScreenSaverWindowLevelKey)`. For further details see `CGWindowLevel.h`.

Windows at lower levels can never be placed visually on top of windows at higher levels, while windows at the same level are ordered relative to each other.

QuickTime 6.1 has changed the window level of its Full Screen Blanking Window
from the default level up to the `ScreenSaver` level which is appropriate for full-screen mode.

By making this change QuickTime is able to prevent certain types of annoying problems from occurring. One specific problem is related to certain *floating-window* applications always drawing themselves above the default window level, enabling them to float above all other windows. Clock.app is one such application.

The problem with this is, the application then has the ability to draw on top of your highly produced full-screen movie. Not the ideal user experience. The window level change prevents problems such as this from occurring.

## You broke my application!

However, some third-party applications may be adversely affected by this change.

`BeginFullScreen` was designed to give developers access to a blanking window into which to draw, display a movie and so on.

Because traditional Mac OS had no concept of distinct window levels maintained by a window server, a few developers noticed that they could call `BeginFullScreen`, make new Carbon-style windows, then display these windows on top of the blanking window by calling `ShowWindow`.

While this practice is technically not an incorrect use of the Full Screen API, the ability to display windows in this manner relies on the specific behavior of traditional Mac OS. To assume this same behavior on Mac OS X is incorrect.

## A simple fix

If your application was affected by this window level change, you can get the old behavior back by adding a call to `SetWindowGroup` after calling `BeginFullScreen`, as demonstrated in Listing 1.

Calling `SetWindowGroup` removes the window from its old group (if a window was already in a group) and assigns it to a new group, in our case it's going to be the group containing document class windows defined as `kDocumentWindowClass`.

As each window group is associated with a layer in the window layering hierarchy, this call will effectively change the layer of the blanking window to document window layer and will allow the placement of new Carbon windows over top of the blanking window.

**Note:**
This only affects Mac OS X.

**Listing 1.** Duplicate pre-QuickTime 6.1 behavior.

```
Ptr
BeginFullScreenTheOldWay(short *width, short *height,
                         WindowRef *outBlankingWindow,
                         long inFlags)
{
     OSErr err;
     Ptr theRestoreState = NULL;

     err = BeginFullScreen(&theRestoreState,
                           NULL,          // main screen
                           width, height,
                           outBlankingWindow,
                           NULL,          // black
                           inFlags);
     if (err) { outBlankingWindow = NULL; goto bail; }

     // restore old behavior
     SetWindowGroup(outBlankingWindow,
                    GetWindowGroupOfClass(kDocumentWindowClass));

bail:
     return theRestoreState;
}
```

Back to top

## Additional flags

BeginFullScreen accepts some flags which control certain aspects of the full-screen mode. For QuickTime 6.1 three new flags were added. These options can improve application performance and allow for display capture.

```
// New to QuickTime 6.1
enum {
  fullScreenDontSwitchMonitorResolution = 1L << 4,
  fullScreenCaptureDisplay = 1 << 5L,      /* mac os x only */
  fullScreenCaptureAllDisplays = 1 << 6L  /* mac os x only */
};
```

- fullScreenDontSwitchMonitorResolution - this flag allows you to initiate full-screen mode without changing the monitor's resolution.

  Previously you could get the same effect by calling BeginFullScreen and passing 0 for the desiredWidth and desiredHeight parameters, however using this flag now allows you to pass in a valid desired width and height. When this is done, BeginFullScreen will *not* change the monitors resolution as expected and will pass back (in the desiredWidth and desiredHeight parameters) the width and height that it *would have* switched to in order to accommodate the requested width and height. This may be useful if your application wanted to forgo the monitor resolution switch but still size its content in the blanking window as if a resolution switch had indeed occurred.

  The use of this flag also makes the resolution switch a much faster operation leaving the desktop icons and other application windows in place.

- fullScreenCaptureDisplay - this flag causes QuickTime to capture the display being used for full-screen through the use of CGDirect Display. Capturing the display prevents contention for the screen from other applications and system services. This doesn't look any different, but it will block the Command + Tab key.

- fullScreenCaptureAllDisplays - this flag is similar to the fullScreenCaptureDisplay flag except it will blank out all displays connected to the system, not just the one you are drawing to. This is visually similar to what is seen when iTunes switches to full-screen visualization mode.

Back to top

## References

Using Full Screen

Full Screen Flags

CGDirectDisplay API

Back to top

## Downloadables

Acrobat version of this Note (56K) Download

Back to top

Technical Notes by Date | Number | Technology | Title
Developer Documentation | Technical Q&As | Development Kits | Sample Code