

TECHNOTE: On Changes to QuickTime Conferencing Components: New Error Codes, Behavior & Methods of Retrieving Errors

By Scott Kuechle
kuechle1@applelink.apple.com
Apple Developer Technical Support (DTS)

This Technote describes new error codes, documents new behavior, and provides information on retrieving errors involved in the various components of QuickTime Conferencing (QTC).

This Note is aimed at all developers whose projects include QuickTime Conferencing.

Note

The sections in this Technote refer to topics supported by different versions of QuickTime Conferencing. The title to each section includes the version of QTC in which that topic is first supported. ♦

Getting Stream Controller Errors When a Snapshot Fails, QTC 1.0.2

This section shows how applications can retrieve errors set by a QuickTime Conferencing version 1.0.2 Stream Controller Component in the case where a Snapshot fails, and is directed towards those developers who are using the QuickTime Conferencing Stream Controller Components API.

Snapshot Failed Errors

Under QuickTime Conferencing 1.0, a stream controller had no way of conveying errors which occur when a snapshot fails. The only indication that an error occurred was that the parameter passed into the action filter procedure for the `mtControllerActionSnapshot` and `mtControllerActionSnapshotData` action codes was NULL. But there was no way of getting the actual error number.

Beginning with QuickTime 1.0.2, a stream controller uses the Component Manager function `SetComponentInstanceError` to convey the error in the case of a failed snapshot. If an application gets a NULL parameter for the `mtControllerActionSnapshot` or `mtControllerActionSnapshotData` action codes, it can then call the Component Manager function `GetComponentInstanceError` to find out the specific error number.

Further Reference

- *Inside Macintosh: QuickTime Conferencing*, Stream Control Components, Chapter 4
- *Inside Macintosh: More Macintosh Toolbox*, Component Manager, Chapter 6

Error Codes for MTFLOWControlInit, QTC 1.0.2

The Flow Control Components chapter of *Inside Macintosh: QuickTime Conferencing* version describes the `MTFLOWControlInit` function, along with result codes that may be returned by this call. Since the publication of this document, however, some new error codes have been defined. This section describes the new error codes that were added beginning with QuickTime Conferencing version 1.0.2, and is for developers who are writing QuickTime Conferencing Flow Control Components.

New Error Codes for MTFLOWControlInit

Starting with QuickTime Conferencing version 1.0.2, the `MTFLOWControlInit` function can now return two new error codes, in addition to those listed in the Flow Control Components chapter of *Inside Macintosh: QuickTime Conferencing* page 7-13:

```
invalidComponentID  
memFullErr
```

This only applies to flow control components of type `kMTFLOWControlSender`. The outgoing flow control component tries to open the QTC SNMP regulating component (which was not documented for QTC 1.0). If the outgoing flow control component cannot find the component, `invalidComponentID` is returned. If the component cannot be opened due to memory errors, `memFullErr` is returned.

Further Reference

- *Inside Macintosh: QuickTime Conferencing*, Flow Control Components Chapter, 7-12

Conferencing MTRecorderChangedStreams & mtRenegotiatedErr, QTC 1.0.2

This section documents new behavior for the QuickTime Conferencing version 1.0.2 `MTRecorderChangedStreams` function. It is aimed at developers who are writing QuickTime Conferencing Recorder Components or Stream Directors and application developers who use the Recorder Components APIs.

MTRecorderChangedStreams & mtRenegotiatedErr

According to *Inside Macintosh: QuickTime Conferencing*, an application needs to call the `MTRecorderChangedStreams` function (see page 8-24) to inform recorder components about stream additions or deletions, and about changes to the format of streams it is recording. With QuickTime Conferencing version 1.0.2, however, this is no longer true.

Instead, the source and sink stream directors call the recorder's callback recording procedure routine, passing it a chunk with an the error field set to `mtRenegotiatedErr`. This indicates to the recorder that the stream formats and stream ID's have changed and that it needs to make the appropriate calls to the stream director to update the recording.

Developers of stream directors will need to call the recording procedure with a chunk which has the error field set to `mtRenegotiatedErr` whenever streams change when it is recording.

Developers of recorder components will need to look for the `mtRenegotiatedErr` value in the chunk error field of the media chunk and react appropriately to the new stream formats and ID's.

The error code `mtRenegotiatedErr` can be found in the QTC 1.0.2 interface files and is defined as follows:

```
mtRenegotiatedErr    -7886
```

Further Reference

- *Inside Macintosh: QuickTime Conferencing*, Recorder Components, Chapter 8, Stream Director Components, Chapter 5

QuickTime Conferencing Gestalt Selector Code, QTC 1.0.3

This section documents a new Gestalt Selector Code for QuickTime Conferencing version 1.0.3 and later. This section is aimed at all developers using the QuickTime Conferencing APIs.

New Gestalt Selector Code

A new Gestalt selector was added into QuickTime Conferencing version 1.0.3 and later. The new selector is:

```
#define gestaltQuickTimeConferencingInfo 'qtci'
```

This new selector was necessary for backwards and forwards compatability of the Apple Media Conference (AMC) application, and to provide applications with more information about the installed version of QTC.

The new Gestalt selector will return a pointer to the following structure:

```
struct QTCGestaltInfo {
    long QTCversion; /* standard binary version - i.e. 0x01044007
                    means "1.0.4a7" */
    char QTCversionString[20]; /* this is a pascal string, followed
                               by a zero char */
};
```

▲ **WARNING**

Do not modify or dispose of the pointer to this structure - treat it as read-only information.

The reason for this is that version 1.0.2 of Apple Media Conference (also, therefore, Connectix VideoPhone) expects the same QTC version number as returned by the existing `gestaltQuickTimeConferencing ('mtlk')` Gestalt selector, and all the currently shipped 1.0.2 applications would have failed to run if the old selector was changed from its 0x01028000 value — newer versions of QTC extensions essentially must lie about their version number to keep AMC 1.0.2 happy. Yes, that's also why Apple Media Conference 1.0.2 still says the QTC version is 1.0.2, even when a newer version of QTC is actually installed. ▲

Starting in QTC 1.0.3 and later, the new Gestalt was created to detach the dependency of AMC on specific releases of QTC and provide a way for applications to get and display the real QTC version number.

If the Gestalt selector points at a zero version number and empty string, it means that some QTC-related extension has already loaded, but QTC itself is not installed or has not yet loaded.

Note

The `QTCversionString` is a Pascal string. Since there's a zero character after it, however, you can add one byte to the address and use it as a C string. ♦

Further Reference

- *Inside Macintosh: QuickTime Conferencing*
- *Inside Macintosh: Operating System Utilities, Gestalt Manager, Chapter 1*

PowerTalk Browser Notes

The PowerTalk browser included with QTC 1.0.2 (and above) is *not* a dialog box. Therefore, any application-defined modal filter procedure provided when

calling the `MTBrowserBrowse` function (see *Inside Macintosh:QuickTime Conferencing* 3-13) for the PowerTalk browser must not make any Dialog Manager calls. Doing so may result in a program crash.

Also, when the PowerTalk browser is being used with the `MTBrowserBrowse` function, two additional errors not documented in the Browser chapter (chapter 3) of *Inside Macintosh:QuickTime Conferencing* may be returned:

```
invalidComponentID  
memFullErr
```

Essentially, the PowerTalk browser requires another component to do its work. If you specify the PowerTalk browser when calling the `MTBrowserBrowse` function it tries to find and open such a component. If the call can't find this component, `MTBrowserBrowse` will return `invalidComponentID`. `MTBrowserBrowse` will return the error `memFullErr` if the component can't be opened because of memory-related problems.

Speaker Designation “Off While Recording” Mislabeled

The Apple Media Conference application 1.0.2 (and above) speaker designation Off While Recording is somewhat misleading. The Off While Recording speaker designation can be set by first selecting the Sound Settings... menu item under the Settings menu. Once in the Sound settings dialog window select Source from the pop-up menu and specify Off While Recording in the Speaker pop-up menu.

Since there exists a recording function within the Apple Media Conference application that is separate and distinct from the conference or broadcast functions, many users assume that the Off While Recording designation will turn the speakers off only when they elect to make a recording. In fact, it shuts off the speaker when a connection is made.

MTTransportDisconnect does not return mtIncompatibleStateErr

The description of the `MTTransportDisconnect` function in *Inside Macintosh: QuickTime Conferencing*, page 9-28, the QTC 1.0.2 is incorrect.

The `MTTransportDisconnect` function can legally be called at any time to clear the state.

Interrupt-Time Processing with QTC Network Calls

Inside Macintosh: QuickTime Conferencing, page 10-18, lists the following QTC 1.0.2 (and above) Network Component functions that should not be called at interrupt time:

`MTNetworkExtractName`
`MTNetworkOptions`
`MTNetworkSetNotifyProc`
`MTNetworkSetReceiveMediaProc`

The following calls are safe to make at interrupt time:

`MTNetworkSend`
`MTNetworkReceive`
`MTNetworkSendMedia`

QTC SNMP MTAllowedBandwidth MIB variable

SNMP Administrators must note that if the QTC 1.0.2 (and above) SNMP Agent `MTAllowedBandwidth` MIB variable is set to zero, it will not stop all traffic on a given QTC connection. It will, however, cause the connection to slow down to a minimum throughput of 1 frame-per-second (fps). This is equivalent

to a value of 25000 for the `MTallowedBandwidth` MIB variable. It is not possible for an SNMP Administrator to stop traffic completely for a given QTC connection.

No QTC CFM-68K Support Yet

QTC 1.0.2 (and above) does not yet provide support for the 680x0 Macintosh Code Fragment Manager (CFM-68K). Such support would mean new QTC MPW Interface files and QTC interface libraries. The QTC team is investigating possibly providing these in future revisions of the product. In the meantime, developers who are creating 680x0 Macintosh QTC applications will need to continue to use static libraries.

Source Stream Director Restrictions

With QTC 1.0.2 (and above), it is not possible to set the various visual characteristics for a source stream director. Thus the following calls made to a source stream director will fail with the `cantDoThatInCurrentMode` (-9402) error:

```
MTDirectorSetMatrix  
MTDirectorSetRect  
MTDirectorSetClip
```

The same code will, however, work fine for a sink stream director. This behavior may be corrected in future versions of QTC.

Additionally, with QTC 1.0.2 (and above) it is possible to create a window that contains a sink view and its associated controller at some fixed offset (x,y) within the window. This is not currently possible for a given source view and its associated controller. A source controller may reside at some offset within a window but the corresponding source view must be placed at offset (0,0) in the same window. This behavior may be corrected in future versions of QTC.

Summary

There are many new error codes, new behaviors, and new information on retrieving errors involved in the various components of QuickTime Conferencing. If you're developing any projects that include QuickTime Conferencing, you'll want to be familiar with these changes. Check this Technote for information on changes in future releases of QTC.

Acknowledgments

Thanks to Jay Geagan, Steve Grigory, and Guy Riddle.