QuickTime Virtual Reality Reference

QuickTime > Virtual Reality

2006-11-10

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Contents

QuickTime Virtual Reality Reference

Framework: Frameworks/QuickTime.framework

Declared in QuickTimeVR.h

Overview

QuickTime Virtual Reality (QTVR) is Apple's cross-platform technology for creating 360-degree panoramas and object movies. Developers can use QTVR to turn photos and computer renderings into interactive 3D views and then link these into entire 3D worlds.

Functions by Task

Accessing Image Buffers

`QTVRRefreshBackBuffer` (page 73)
Refreshes the QTVR back buffer.

`QTVRSetBackBufferImagingProc` (page 77)
Installs or removes a QTVR back buffer imaging procedure.

`QTVRSetPrescreenImagingCompleteProc` (page 90)
Installs or removes a prescreen buffer imaging completion procedure.

Converting Angles and Points

`QTVRAnglesToCoord` (page 20)
Obtains a floating-point coordinate determined by a pair of pan and tilt angles.

`QTVRColumnToPan` (page 22)
Get the pan angle that corresponds to a column number in the object image array.

`QTVRCoordToAngles` (page 23)
Get the pan and tilt angles of a floating-point coordinate in a panorama.

`QTVRGetAngularUnits` (page 28)
Obtains the type of unit currently used when specifying angles.

`QTVRPanToColumn` (page 70)
Obtains the column number in the object image array that corresponds to a pan angle.

`QTVRPtToAngles` (page 71)
Obtains the pan and tilt angles of a point.

QTVRRowToTilt (page 74)
Obtains the tilt angle that corresponds to a row number in a QTVR object image array.

QTVRSetAngularUnits (page 75)
Sets the type of unit used when specifying QTVR angles.

QTVRTiltToRow (page 99)
Obtains the row number in the QTVR object image array that corresponds to a tilt angle.

QTVRWrapAndConstrain (page 101)
Preflights a change in the viewing or control characteristics of a QTVR object or panoramic node.

Determining Viewing Limits and Constraints

QTVRGetConstraints (page 34)
Obtains the current constraints of a QuickTime VR movie.

QTVRGetConstraintStatus (page 34)
Obtains the set of constraints active for the current view.

QTVRGetViewingLimits (page 52)
Obtains the current viewing limits of a QuickTime VR movie.

QTVRSetConstraints (page 79)
Sets the constraints of a VR movie.

Getting Scene and Node Information

QTVRGetCurrentNodeID (page 37)
Obtains the current node of a movie.

QTVRGetNodeInfo (page 45)
Obtains the node information atom container that describes a node and all the hot spots in the node.

QTVRGetNodeType (page 46)
Obtains the type of a movie node.

QTVRGetVRWorld (page 57)
Obtains the VR world atom container for a movie.

QTVRGoToNodeID (page 58)
Sets the current node of a movie.

Handling Events

QTVRGetMouseDownTracking (page 44)
Obtains the current state of mouse-down tracking.

QTVRGetMouseOverTracking (page 44)
Obtains the current state of mouse-over tracking.

QTVRMouseDown (page 61)
Handles the user's clicking the mouse button when the cursor is in a QuickTime VR movie for which mouse-down tracking is disabled.

QTVRMouseEnter (page 63)
Handles the user's moving the cursor into a QuickTime VR movie for which mouse-over tracking is disabled.

QTVRMouseLeave (page 63)
Handles the user's moving the cursor out of a QuickTime VR movie for which mouse-over tracking is disabled.

QTVRMouseStillDown (page 64)
Handles the user's holding down the mouse button while the cursor is in a QuickTime VR movie for which mouse-down tracking is disabled.

QTVRMouseStillDownExtended (page 65)
Handles the user's holding down the mouse button while the cursor is in a QuickTime VR movie for which mouse-down tracking is disabled.

QTVRMouseUp (page 66)
Handles the user's releasing the mouse button while the cursor is in a QuickTime VR movie for which mouse-down tracking is disabled.

QTVRMouseUpExtended (page 67)
Handles the user's releasing the mouse button while the cursor is in a QuickTime VR movie for which mouse-down tracking is disabled.

QTVRMouseWithin (page 69)
Handles the user's leaving the cursor in a QuickTime VR movie for which mouse-over tracking is disabled.

QTVRSetMouseDownTracking (page 87)
Sets the state of mouse-down tracking.

QTVRSetMouseOverTracking (page 89)
Sets the state of mouse-over tracking.

Intercepting QuickTime VR Manager Routines

QTVRCallInterceptedProc (page 22)
Calls an intercepted QuickTime VR function from within an intercept procedure.

QTVRInstallInterceptProc (page 59)
Installs or removes an intercept procedure for a QuickTime VR Manager function.

Managing Hot Spots

QTVREnableHotSpot (page 25)
Enables or disables one or more QTVR hot spots.

QTVRGetHotSpotRegion (page 40)
Obtains the region occupied by a hot spot.

QTVRGetHotSpotType (page 41)
Obtains the type of a QuickTime VR hot spot.

QTVRGetVisibleHotSpots (page 56)
Obtains a list of the currently visible hot spots in a QuickTime VR movie.

QTVRPtToHotSpotID (page 72)
Obtains the ID of the hot spot, if any, that lies beneath a point.

QTVRSetMouseOverHotSpotProc (page 88)
Installs or removes a mouse over hot spot procedure.

QTVRTriggerHotSpot (page 99)
Triggers a QTVR hot spot.

Managing Imaging Characteristics

QTVRBeginUpdateStream (page 21)
Begins a stream of immediate updates to a QuickTime VR movie.

QTVREnableTransition (page 26)
Enables or disables a transition effect.

QTVREndUpdateStream (page 27)
Ends a stream of immediate updates to a QuickTime VR movie.

QTVRGetImagingProperty (page 42)
Obtains the current value of an imaging property of a movie.

QTVRGetVisible (page 56)
Obtains a movie's visibility state.

QTVRSetImagingProperty (page 84)
Sets the value of an imaging property of a movie.

QTVRSetTransitionProperty (page 93)
Sets the value of a transition property.

QTVRSetVisible (page 97)
Sets a VR movie's visibility state.

QTVRUpdate (page 101)
Forces an immediate update of a QuickTime VR movie image.

Managing Object Nodes

QTVREnableFrameAnimation (page 24)
Enables or disables frame animation for an object node.

QTVREnableViewAnimation (page 27)
Enables or disables view animation for an object node.

QTVRGetAnimationSetting (page 29)
Obtains the current state of an animation setting for an object node.

QTVRGetControlSetting (page 35)
Obtains the current state of a control setting for an object node.

QTVRGetCurrentMouseMode (page 36)
Obtains the current mouse control modes.

QTVRGetCurrentViewDuration (page 37)
Obtains the duration of the current view of an object node.

QTVRGetFrameAnimation (page 39)
Obtains the current state of frame animation for an object node.

`QTVRGetFrameRate` (page 39)
Obtains the current frame rate of an object node.

`QTVRGetViewAnimation` (page 50)
Obtains the current state of view animation for an object node.

`QTVRGetViewCurrentTime` (page 51)
Obtains the current time in the current view.

`QTVRGetViewRate` (page 54)
Obtains the current view rate of an object node.

`QTVRGetViewState` (page 54)
Obtains the value of a view state.

`QTVRGetViewStateCount` (page 55)
Obtains the number of view states of an object node.

`QTVRSetAnimationSetting` (page 76)
Sets the state of an animation setting for an object node.

`QTVRSetControlSetting` (page 80)
Sets the state of a control setting for a QTVR object node.

`QTVRSetFrameRate` (page 83)
Sets the frame rate of an object node.

`QTVRSetViewCurrentTime` (page 94)
Sets the time in the current QTVR view.

`QTVRSetViewRate` (page 96)
Sets the view rate of a QTVR object node.

`QTVRSetViewState` (page 97)
Sets the value of a QTVR view state.

Managing QuickTime VR Movie Instances

`QTVRGetQTVRInstance` (page 47)
Obtains an instance of a QuickTime VR movie.

`QTVRGetQTVRTrack` (page 48)
Obtains a QTVR track contained in a QuickTime movie to use in the QTVRGetQTVRInstance call.

Managing QuickTime VR Movie Interactions

`QTVRGetInteractionProperty` (page 43)
Obtains the value of an interaction property.

`QTVRReplaceCursor` (page 73)
Replaces any of the standard QuickTime VR cursors with your own custom cursor.

`QTVRSetEnteringNodeProc` (page 81)
Installs or removes a node-entering procedure.

`QTVRSetInteractionProperty` (page 85)
Sets the value of an interaction property.

QTVRSetLeavingNodeProc (page 86)
Installs or removes a node-leaving procedure.

Managing VR Memory

QTVRGetAvailableResolutions (page 30)
Obtains the image resolutions present in the current node.

QTVRGetBackBufferMemInfo (page 30)
Obtains information about the internal back buffer that QuickTime VR maintains for caching panoramic images.

QTVRGetBackBufferSettings (page 32)
Obtains information about the resolution, pixel format, and size of the back buffer maintained internally by QuickTime VR for caching a panoramic image in a particular pixel format.

QTVRSetBackBufferPrefs (page 78)
Sets the resolution, pixel format, and size of the back buffer maintained internally by QuickTime VR for caching a panoramic image in a particular pixel format.

Manipulating Viewing Angles and Zooming

QTVRGetFieldOfView (page 38)
Obtains the vertical field of view of a QuickTime VR movie.

QTVRGetPanAngle (page 46)
Obtains the pan angle of a QuickTime VR movie.

QTVRGetTiltAngle (page 49)
Obtains the tilt angle of a QuickTime VR movie.

QTVRGetViewCenter (page 51)
Obtains the view center of a QuickTime VR movie.

QTVRInteractionNudge (page 60)
Translates the image and displays the new view or rotates the object in a particular direction and displays its new appearance.

QTVRNudge (page 69)
Turns one step in a particular direction and displays the new view.

QTVRSetFieldOfView (page 82)
Sets the vertical field of view of a QuickTime VR movie.

QTVRSetPanAngle (page 89)
Sets the pan angle of a QuickTime VR movie.

QTVRSetTiltAngle (page 92)
Sets the tilt angle of a QuickTime VR movie.

QTVRSetViewCenter (page 94)
Sets the view center of a QuickTime VR movie.

QTVRShowDefaultView (page 98)
Displays the default view of a QTVR node.

Supporting Functions

Functions

DisposeQTVRBackBufferImagingUPP

Disposes of a QTVRBackBufferImagingUPP pointer.

```
void DisposeQTVRBackBufferImagingUPP (
   QTVRBackBufferImagingUPP userUPP
);
```

Parameters

userUPP

A `QTVRBackBufferImagingUPP` pointer. See `Universal Procedure Pointers`.

Return Value
You can access this function's error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrbackbuffer
vrbackbuffer.win
vrmovies
vrmovies.win
vrscript

Declared In
`QuickTimeVR.h`

DisposeQTVREnteringNodeUPP

Disposes of a QTVREnteringNodeUPP pointer.

```
void DisposeQTVREnteringNodeUPP (
   QTVREnteringNodeUPP userUPP
);
```

Parameters

userUPP
A `QTVREnteringNodeUPP` pointer. See `Universal Procedure Pointers`.

Return Value
You can access this function's error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

DisposeQTVRImagingCompleteUPP

Disposes of a QTVRImagingCompleteUPP pointer.

```
void DisposeQTVRImagingCompleteUPP (
   QTVRImagingCompleteUPP userUPP
);
```

Parameters

userUPP

A `QTVRImagingCompleteUPP` pointer. See `Universal Procedure Pointers`.

Return Value
You can access this function's error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

DisposeQTVRInterceptUPP

Disposes of a QTVRInterceptUPP pointer.

```
void DisposeQTVRInterceptUPP (
   QTVRInterceptUPP userUPP
);
```

Parameters

userUPP

A `QTVRInterceptUPP` pointer. See `Universal Procedure Pointers`.

Return Value
You can access this function's error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

DisposeQTVRLeavingNodeUPP

Disposes of a QTVRLeavingNodeUPP pointer.

```
void DisposeQTVRLeavingNodeUPP (
   QTVRLeavingNodeUPP userUPP
);
```

Parameters

userUPP

A `QTVRLeavingNodeUPP` pointer. See `Universal Procedure Pointers`.

Return Value
You can access this function's error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

DisposeQTVRMouseOverHotSpotUPP

Disposes of a QTVRMouseOverHotSpotUPP pointer.

```
void DisposeQTVRMouseOverHotSpotUPP (
   QTVRMouseOverHotSpotUPP userUPP
);
```

Parameters

userUPP

A `QTVRMouseOverHotSpotUPP` pointer. See `Universal Procedure Pointers`.

Return Value
You can access this function's error returns through `GetMoviesError` and `GetMoviesStickyError`.

Version Notes
Introduced in QuickTime 4.1.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

NewQTVRBackBufferImagingUPP

Allocates a Universal Procedure Pointer for the QTVRBackBufferImagingProc callback.

```
QTVRBackBufferImagingUPP NewQTVRBackBufferImagingUPP (
   QTVRBackBufferImagingProcPtr userRoutine
);
```

Parameters

userRoutine

A pointer to your application-defined function.

Return Value
A new UPP; see `Universal Procedure Pointers`.

Discussion
This function is used with Macintosh PowerPC systems. See *Inside Macintosh: PowerPC System Software*.

Version Notes
Introduced in QuickTime 4.1. Replaces `NewQTVRBackBufferImagingProc`.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrbackbuffer
vrbackbuffer.win
vrmovies
vrmovies.win
vrscript.win

Declared In
`QuickTimeVR.h`

NewQTVREnteringNodeUPP

Allocates a Universal Procedure Pointer for the QTVREnteringNodeProc callback.

```
QTVREnteringNodeUPP NewQTVREnteringNodeUPP (
   QTVREnteringNodeProcPtr userRoutine
);
```

Parameters

`userRoutine`
A pointer to your application-defined function.

Return Value
A new UPP; see `Universal Procedure Pointers`.

Discussion
This function is used with Macintosh PowerPC systems. See *Inside Macintosh: PowerPC System Software*.

Version Notes
Introduced in QuickTime 4.1. Replaces `NewQTVREnteringNodeProc`.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win
vrspeech

Declared In
`QuickTimeVR.h`

NewQTVRImagingCompleteUPP

Allocates a Universal Procedure Pointer for the QTVRImagingCompleteProc callback.

```
QTVRImagingCompleteUPP NewQTVRImagingCompleteUPP (
   QTVRImagingCompleteProcPtr userRoutine
);
```

Parameters

userRoutine

A pointer to your application-defined function.

Return Value

A new UPP; see `Universal Procedure Pointers`.

Discussion

This function is used with Macintosh PowerPC systems. See *Inside Macintosh: PowerPC System Software*.

Version Notes

Introduced in QuickTime 4.1. Replaces `NewQTVRImagingCompleteProc`.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrscript

vrscript.win

Declared In

`QuickTimeVR.h`

NewQTVRInterceptUPP

Allocates a Universal Procedure Pointer for the QTVRInterceptProc callback.

```
QTVRInterceptUPP NewQTVRInterceptUPP (
   QTVRInterceptProcPtr userRoutine
);
```

Parameters

userRoutine

A pointer to your application-defined function.

Return Value

A new UPP; see `Universal Procedure Pointers`.

Discussion

This function is used with Macintosh PowerPC systems. See *Inside Macintosh: PowerPC System Software*.

Version Notes

Introduced in QuickTime 4.1. Replaces `NewQTVRInterceptProc`.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win
vrspeech

Declared In
`QuickTimeVR.h`

NewQTVRLeavingNodeUPP

Allocates a Universal Procedure Pointer for the QTVRLeavingNodeProc callback.

```
QTVRLeavingNodeUPP NewQTVRLeavingNodeUPP (
   QTVRLeavingNodeProcPtr userRoutine
);
```

Parameters

`userRoutine`
A pointer to your application-defined function.

Return Value
A new UPP; see `Universal Procedure Pointers`.

Discussion
This function is used with Macintosh PowerPC systems. See *Inside Macintosh: PowerPC System Software*.

Version Notes
Introduced in QuickTime 4.1. Replaces `NewQTVRLeavingNodeProc`.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

NewQTVRMouseOverHotSpotUPP

Allocates a Universal Procedure Pointer for the QTVRMouseOverHotSpotProc callback.

```
QTVRMouseOverHotSpotUPP NewQTVRMouseOverHotSpotUPP (
   QTVRMouseOverHotSpotProcPtr userRoutine
);
```

Parameters

`userRoutine`
A pointer to your application-defined function.

Return Value
A new UPP; see `Universal Procedure Pointers`.

Discussion
This function is used with Macintosh PowerPC systems. See *Inside Macintosh: PowerPC System Software*.

Version Notes
Introduced in QuickTime 4.1. Replaces `NewQTVRMouseOverHotSpotProc`.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrcursors
vrcursors.win
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

QTVRAnglesToCoord

Obtains a floating-point coordinate determined by a pair of pan and tilt angles.

```
OSErr QTVRAnglesToCoord (
   QTVRInstance qtvr,
   float panAngle,
   float tiltAngle,
   QTVRFloatPoint *coord
);
```

Parameters

qtvr
An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

panAngle
A pan angle.

tiltAngle
A tilt angle.

coord
On entry, a pointer to a `QTVRFloatPoint` structure. On return, that structure is set to the coordinate of the specified movie that corresponds to the specified pan and tilt angles.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Special Considerations
`QTVRAnglesToCoord` is valid only for panoramic nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeVR.h

QTVRBeginUpdateStream

Begins a stream of immediate updates to a QuickTime VR movie.

```
OSErr QTVRBeginUpdateStream (
   QTVRInstance qtvr,
   QTVRImagingMode imagingMode
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

imagingMode

An imaging mode (see below). See these constants:

kQTVRStatic

kQTVRMotion

kQTVRAllModes

Return Value
See Error Codes. Returns noErr if there is no error.

Discussion
This function configures the QuickTime VR movie specified by the qtvr parameter for a stream of immediate updates to its movie image. After calling QTVRBeginUpdateStream, you perform the updates by calling QTVRUpdate (page 101). When you are finished performing the updates, call QTVREndUpdateStream (page 27). Issuing a stream of image updates in this manner is significantly faster than simply calling QTVRUpdate repeatedly (that is, not within a begin/end update sequence). Each call to this function must be balanced by a call to QTVREndUpdateStream, but you can nest these calls.

Special Considerations

After you call this function and before you call QTVREndUpdateStream (page 27), you must not change any of the QuickTime VR movie's imaging properties.

Calling this function locks down large blocks of memory. As a result, you should minimize the amount of time before calling QTVREndUpdateStream.

This function is valid only for panoramic nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
QuickTimeVR.h

QTVRCallInterceptedProc

Calls an intercepted QuickTime VR function from within an intercept procedure.

```
OSErr QTVRCallInterceptedProc (
   QTVRInstance qtvr,
   QTVRInterceptRecord *qtvrMsg
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

qtvrMsg

A pointer to a QTVRInterceptRecord structure that specifies the function that your procedure is intercepting and the parameters for that function. This should be the same intercept record passed to your intercept procedure.

Return Value
See Error Codes. Returns noErr if there is no error.

Discussion
This function executes the QuickTime VR Manager function indicated by the selector field of the qtvrMsg intercept record. The parameters passed to that function are the QuickTime VR movie specified by the qtvr parameter and any other parameters contained in the parameter field of the qtvrMsg record. You can, if you wish, change the parameters in that field before calling this function.

You can call this function more than once in your intercept procedure. In addition, the QuickTime VR Manager will call the intercepted function again unless your intercept procedure returns TRUE in the cancel parameter.

Special Considerations

You should call QTVRCallInterceptedProc only in an intercept procedure.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrspeech

Declared In
QuickTimeVR.h

QTVRColumnToPan

Get the pan angle that corresponds to a column number in the object image array.

```
float QTVRColumnToPan (
   QTVRInstance qtvr,
   short column
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

column

A column number.

Return Value

The pan angle that corresponds to the zero-based column number in the object image array specified by the `column` parameter.

Special Considerations

This function is valid only for object nodes.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeVR.h`

QTVRCoordToAngles

Get the pan and tilt angles of a floating-point coordinate in a panorama.

```
OSErr QTVRCoordToAngles (
   QTVRInstance qtvr,
   QTVRFloatPoint *coord,
   float *panAngle,
   float *tiltAngle
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

coord

On entry, a pointer to a `QTVRFloatPoint` structure that specifies a coordinate in the full panorama.

panAngle

On entry, a pointer to a floating-point value. On return, that value contains the pan angle of the specified coordinate.

tiltAngle

On entry, a pointer to a floating-point value. On return, that value contains the tilt angle of the specified coordinate.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function returns, in the floating-point values pointed to by the `panAngle` and `tiltAngle` parameters, the pan and tilt angles of the point specified by the `coord` parameter. This function is useful for setting up angles in a back buffer imaging procedure; if you know a coordinate in the back buffer, you can call `QTVRCoordToAngles` to get the corresponding angles.

Special Considerations

`QTVRCoordToAngles` is valid only for panoramic nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVREnableFrameAnimation

Enables or disables frame animation for an object node.

```
OSErr QTVREnableFrameAnimation (
   QTVRInstance qtvr,
   Boolean enable
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

enable

A Boolean value that indicates whether to enable (TRUE) or disable (FALSE) frame animation for the specified object node.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function enables or disables the frame animation state for the object node specified by the `qtvr` parameter, according to the value of the `enable` parameter. The current frame rate, set by the function `QTVRSetFrameRate` (page 83), is unaffected by the state of frame animation of an object node.

Special Considerations

This function is valid only for object nodes. You should use this function instead of standard QuickTime functions to control object animation.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

QTVREnableHotSpot

Enables or disables one or more QTVR hot spots.

```
OSErr QTVREnableHotSpot (
   QTVRInstance qtvr,
   UInt32 enableFlag,
   UInt32 hotSpotValue,
   Boolean enable
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

enableFlag

The kind of hot spot or hot spots to enable or disable (see below). See these constants:

`kQTVRHotSpotID`
`kQTVRHotSpotType`
`kQTVRAllHotSpots`

hotSpotValue

The desired hot spot or spots, defined by the specified enabled flag (see below).

enable

A Boolean value that indicates whether the specified hot spots are to be enabled (TRUE) or disabled (FALSE).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function either enables or disables the hot spot or spots specified by the `enableFlag` and `hotSpotValue` parameters, according to the value of the `enable` parameter. The hot spots are always selected from among the hot spots in the current node of the QuickTime VR movie specified by the `qtvr` parameter.

Normally, all hot spots in a node are enabled (that is, the cursor automatically changes shape when it is moved over a hot spot, and the `QTVRTriggerHotSpot` (page 99) function is called internally when the user clicks a hot spot). When a hot spot is disabled, QuickTime VR behaves as if the hot spot were not present.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

QTVREnableTransition

Enables or disables a transition effect.

```
OSErr QTVREnableTransition (
   QTVRInstance qtvr,
   UInt32 transitionType,
   Boolean enable
);
```

Parameters

qtvr
An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

transitionType
A type of transition property (see below). Currently only one constant is available for this parameter. See these constants:
`kQTVRTransitionSwing`

enable
A Boolean value that indicates whether the specified transition property is to be enabled (TRUE) or disabled (FALSE).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function enables or disables the transition property specified by the `transitionType` parameter for the movie specified by the `qtvr` parameter, as indicated by the value of the `enable` parameter. Once a transition effect is enabled, it is used at the appropriate time until it is disabled by a subsequent call to this function.

Special Considerations
`QTVREnableTransition` is valid only for panoramic nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
QuickTimeVR.h

QTVREnableViewAnimation

Enables or disables view animation for an object node.

```
OSErr QTVREnableViewAnimation (
   QTVRInstance qtvr,
   Boolean enable
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

enable

A Boolean value that indicates whether to enable (TRUE) or disable (FALSE) view animation for the specified object node.

Return Value
See Error Codes. Returns noErr if there is no error.

Discussion
You should use this function instead of standard QuickTime functions to control object animation.

Special Considerations

This function is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript

vrscript.win

Declared In
QuickTimeVR.h

QTVREndUpdateStream

Ends a stream of immediate updates to a QuickTime VR movie.

```
OSErr QTVREndUpdateStream (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

Return Value

See Error Codes. Returns noErr if there is no error.

Discussion

This function unlocks the memory locked by the matching call to QTVRBeginUpdateStream (page 21) for the QuickTime VR movie specified by the qtvr parameter and reverses any other actions performed by that call. Each call to QTVRBeginUpdateStream must be balanced by a call to this function, but you can nest these calls. For nested calls, only the final call to this function unlocks the memory locked by the first call to QTVRBeginUpdateStream.

Special Considerations

QTVREndUpdateStream is valid only for panoramic nodes.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrscript

vrscript.win

Declared In

QuickTimeVR.h

QTVRGetAngularUnits

Obtains the type of unit currently used when specifying angles.

```
QTVRAngularUnits QTVRGetAngularUnits (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

Return Value

The type of unit currently used (see below).

Discussion

This function returns, as its function result, a constant that indicates the type of angular unit currently used by the movie instance specified by the qtvr parameter. Angular values you pass to other QuickTime VR functions, such as QTVRSetPanAngle (page 89), are interpreted in those units.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRGetAnimationSetting

Obtains the current state of an animation setting for an object node.

```
OSErr QTVRGetAnimationSetting (
   QTVRInstance qtvr,
   QTVRObjectAnimationSetting setting,
   Boolean *enable
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

setting

An animation setting (see below). See these constants:

`kQTVRPalindromeViewFrames`
`kQTVRDontLoopViewFrames`
`kQTVRPlayEveryViewFrame`
`kQTVRSyncViewToFrameRate`
`kQTVRPalindromeViews`
`kQTVRPlayStreamingViews`

enable

On entry, a pointer to a Boolean value. On return, that value is set to TRUE if the specified animation setting is currently enabled for the specified object node or to FALSE otherwise.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Special Considerations

This function is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRGetAvailableResolutions

Obtains the image resolutions present in the current node.

```
OSErr QTVRGetAvailableResolutions (
   QTVRInstance qtvr,
   UInt16 *resolutionsMask
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

resolutionsMask

On entry, a pointer to an unsigned short integer. On return, that integer is set to a bitmask that encodes the image resolutions (see below) available at the current node. See these constants:

`kQTVRDefaultRes`

`kQTVRFullRes`

`kQTVRHalfRes`

`kQTVRQuarterRes`

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

A single node can contain multiple resolutions of a panorama or an object. The lowest order bit is always set and corresponds to the base resolution of the node. Each succeeding bit corresponds to a resolution that is half that (both horizontally and vertically) of the preceding bit. If an image with a corresponding resolution is present in the current node, that bit is set.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeVR.h`

QTVRGetBackBufferMemInfo

Obtains information about the internal back buffer that QuickTime VR maintains for caching panoramic images.

```
OSErr QTVRGetBackBufferMemInfo (
   QTVRInstance qtvr,
   UInt32 geometry,
   UInt16 resolution,
   UInt32 cachePixelFormat,
   SInt32 *minCacheBytes,
   SInt32 *suggestedCacheBytes,
   SInt32 *fullCacheBytes
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

geometry

The geometry parameter (see below) specifies the type and orientation of the panorama data. See these constants:

`kQTVRUseMovieGeometry`

`kQTVRVerticalCylinder`

resolution

The resolution for which the information is desired (see below). See these constants:

`kQTVRDefaultRes`

`kQTVRFullRes`

`kQTVRHalfRes`

`kQTVRQuarterRes`

cachePixelFormat

The desired pixel format for the back buffer. This value should be one of the defined pixel formats (see below). See these constants:

`kQTVRMinimumCache`

`kQTVRSuggestedCache`

`kQTVRFullCache`

minCacheBytes

On entry, a pointer to a long integer. On return, that long integer is set to the minimum size, in bytes, of the back buffer required to display the specified panorama with a severely limited maximum field of view. Set this parameter to `NIL` to prevent this information from being returned.

suggestedCacheBytes

On entry, a pointer to a long integer. On return, that long integer is set to the minimum size, in bytes, of the back buffer required to display the specified panorama with full wide-angle zooming. Set this parameter to `NIL` to prevent this information from being returned.

fullCacheBytes

On entry, a pointer to a long integer. On return, that long integer is set to the minimum size, in bytes, of the back buffer required to have the entire panorama in memory at once. That is the default size of the panorama back buffer. Set this parameter to `NIL` to prevent this information from being returned.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion
You can use this function to get information about the size of the back buffer that would be required for caching a panoramic image of a specified pixel format, geometry, and resolution. This is a "what-if" function: you specify a resolution and a pixel format, and this function returns several buffer sizes. You can use this information, in conjunction with `QTVRSetBackBufferPrefs` (page 78), to exercise some control over the size of the back buffer.

The resolution at which an image is to be displayed is specified by the `resolution` parameter. You can use a resolution that is not in the movie file. Relative to that resolution and the pixel depth determined by the `cachePixelFormat` parameter, this function returns, through the `minCacheBytes` parameter, the minimum size of the buffer needed to display the movie. Using a buffer of that size, however, may result in a severely limited maximum field of view. You can call the `QTVRGetViewingLimits` (page 52) function to determine the `actual` maximum field of view.

To allow full wide-angle zooming, you should use a buffer whose size is specified by either the `suggestedCacheBytes` parameter or the `fullCacheBytes` parameter.

Special Considerations

This function is valid only for panoramic nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRGetBackBufferSettings

Obtains information about the resolution, pixel format, and size of the back buffer maintained internally by QuickTime VR for caching a panoramic image in a particular pixel format.

```
OSErr QTVRGetBackBufferSettings (
   QTVRInstance qtvr,
   UInt32 *geometry,
   UInt16 *resolution,
   UInt32 *cachePixelFormat,
   SInt16 *cacheSize
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

geometry

The type and orientation of the panorama data (see below). See these constants:

`kQTVRUseMovieGeometry`

`kQTVRVerticalCylinder`

resolution

On entry, a pointer to an unsigned short integer. On return, that integer is set to the index of the current image resolution (see below). See these constants:

kQTVRDefaultRes

kQTVRFullRes

kQTVRHalfRes

kQTVRQuarterRes

cachePixelFormat

On entry, a pointer to a long integer. On return, that long integer is set to the pixel format of the current panorama back buffer (see below). See these constants:

cacheSize

On entry, a pointer to a short integer. On return, that integer is set to a value that describes the size of the current panorama back buffer. See these constants:

kQTVRMinimumCache

kQTVRSuggestedCache

kQTVRFullCache

Return Value
See Error Codes. Returns noErr if there is no error.

Discussion
This function returns, through the resolution parameter, the index of the current resolution for the QuickTime VR movie specified by the qtvr parameter. The index indicates which bit in the mask value returned by QTVRGetAvailableResolutions (page 30) specifies the current resolution. For example, if the returned index is 1, the base resolution is being used. If the returned index is 2, then a resolution of half the base resolution is being used. This function also returns the pixel format and the cache size in the cachePixelFormat and cacheSize parameters, respectively.

The QuickTime VR file might not contain an image track corresponding to the resolution indicated by the resolution value returned. The QuickTime VR Manager may have set a lower resolution because memory is low, or the resolution may have been set by a call to QTVRSetBackBufferPrefs (page 78).

Special Considerations

This function is valid only for panoramic nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrbackbuffer

vrbackbuffer.win

vrcursors

vrmakepano.win

vrmovies.win

Declared In
QuickTimeVR.h

QTVRGetConstraints

Obtains the current constraints of a QuickTime VR movie.

```
OSErr QTVRGetConstraints (
   QTVRInstance qtvr,
   UInt16 kind,
   float *minValue,
   float *maxValue
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

kind

The type of constraints to be returned (see below). See these constants:

`kQTVRPan`

`kQTVRTilt`

`kQTVRFieldOfView`

minValue

On entry, a pointer to a floating-point value. On return, the current minimum constraint of the specified type is copied into that value.

maxValue

On entry, a pointer to a floating-point value. On return, the current maximum constraint of the specified type is copied into that value.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function returns, in the floating-point values pointed to by the `minValue` and `maxValue` parameters, the current minimum and maximum constraints of the type specified by the `kind` parameter. The values returned by this function are unaffected by the current control settings.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrscript

vrscript.win

Declared In

`QuickTimeVR.h`

QTVRGetConstraintStatus

Obtains the set of constraints active for the current view.

```
UInt32 QTVRGetConstraintStatus (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

Return Value

A long integer (see below) whose bits encode the constraints currently active for the QuickTime VR movie specified by the `qtvr` parameter.

Discussion

The values returned by `QTVRGetConstraintStatus` are unaffected by the current control settings.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeVR.h`

QTVRGetControlSetting

Obtains the current state of a control setting for an object node.

```
OSErr QTVRGetControlSetting (
   QTVRInstance qtvr,
   QTVRControlSetting setting,
   Boolean *enable
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

setting

A control setting (see below). See these constants:

`kQTVRWrapPan`
`kQTVRWrapTilt`
`kQTVRCanZoom`
`kQTVRReverseHControl`
`kQTVRReverseVControl`
`kQTVRSwapHVControl`
`kQTVRTranslation`

enable

On entry, a pointer to a Boolean value. On return, that value is set to TRUE if the specified control setting is currently enabled for the specified object node or to FALSE otherwise.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function returns, through the `enable` parameter, the current state of the control setting specified by the `setting` parameter for the object node specified by the `qtvr` parameter. If enable is TRUE, the specified setting is currently enabled; otherwise, the setting is disabled.

Special Considerations

`QTVRGetControlSetting` is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrbackbuffer
vrbackbuffer.win
vrcursors
vrmakepano.win
vrmovies.win

Declared In
`QuickTimeVR.h`

QTVRGetCurrentMouseMode

Obtains the current mouse control modes.

```
UInt32 QTVRGetCurrentMouseMode (
   QTVRInstance qtvr
);
```

Parameters

qtvr
An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

Return Value
A constant (see below) that describes the current mouse control modes.

Discussion
The value returned by this function is an unsigned long integer that encodes the current mouse control modes. If a bit in the integer is set, the corresponding mode is one of the current mouse modes. The mode bits are addressed using the above constants. Notice that several modes can be returned. That means a return value could have both zooming and translating set, for example.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeVR.h

QTVRGetCurrentNodeID

Obtains the current node of a movie.

```
UInt32 QTVRGetCurrentNodeID (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

Return Value
The ID of the current node of the specified movie.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrbackbuffer
vrbackbuffer.win
vrmovies.win
vrscript
vrscript.win

Declared In
QuickTimeVR.h

QTVRGetCurrentViewDuration

Obtains the duration of the current view of an object node.

```
TimeValue QTVRGetCurrentViewDuration (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

Return Value
The duration of the current view of the object node specified by the qtvr parameter.

Special Considerations

This function is valid only for object nodes. You cannot change a node's view duration.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRGetFieldOfView

Obtains the vertical field of view of a QuickTime VR movie.

```
float QTVRGetFieldOfView (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

Return Value
The current vertical field of view of the QuickTime VR movie specified by the `qtvr` parameter. The vertical field of view is a floating-point value that specifies the angle created by the two lines that connect the viewpoint to the top and bottom of the image.

Discussion
The following code fragment illustrates the use of this function:

```
// QTVRGetFieldOfView coding example
#define kDirIn       4L
#define kDirOut      5L
void MyZoomInOrOut (QTVRInstance theInstance, long theDir)
{
    float    theFloat;
    theFloat =QTVRGetFieldOfView(theInstance);
    switch (theDir) {
        case kDirIn:
            theFloat =theFloat / 2.0;
            break;
        case kDirOut:
            theFloat =theFloat * 2.0;
            break;
        default:
            break;
    }
    QTVRSetFieldOfView(theInstance, theFloat);
    QTVRUpdate(theInstance, kQTVRStatic);
}
```

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win
vrspeech

Declared In
`QuickTimeVR.h`

QTVRGetFrameAnimation

Obtains the current state of frame animation for an object node.

```
Boolean QTVRGetFrameAnimation (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

Return Value
TRUE if frame animation is currently enabled, FALSE otherwise.

Special Considerations

This function is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRGetFrameRate

Obtains the current frame rate of an object node.

```
float QTVRGetFrameRate (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

Return Value
The current frame rate of the object node specified by the `qtvr` parameter. A frame rate is a floating-point value in the range from -100.0 to +100.0.

Discussion
An object node's default frame rate is stored in the movie file.

Special Considerations
`QTVRGetFrameRate` is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

QTVRGetHotSpotRegion

Obtains the region occupied by a hot spot.

```
OSErr QTVRGetHotSpotRegion (
   QTVRInstance qtvr,
   UInt32 hotSpotID,
   RgnHandle hotSpotRegion
);
```

Parameters

qtvr
An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

hotSpotID
A hot spot ID.

hotSpotRegion
On entry, an initialized handle to a region. On return, this region is rewritten with the region occupied by the hot spot having the specified ID.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
The returned region is clipped to the bounds of the movie's graphics world. You can obtain the regions of all visible hot spots by calling `QTVRGetVisibleHotSpots` (page 56) and then calling this function for each hot spot ID in the list.

Special Considerations

The first time you call this function, a significant amount of memory might need to be allocated. Accordingly, you should always check for Memory Manager errors returned by this function. Your application is responsible for disposing of the memory occupied by the returned region.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

QTVRGetHotSpotType

Obtains the type of a QuickTime VR hot spot.

```
OSErr QTVRGetHotSpotType (
   QTVRInstance qtvr,
   UInt32 hotSpotID,
   OSType *hotSpotType
);
```

Parameters

qtvr
An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

hotSpotID
A hot spot ID.

hotSpotType
On entry, a pointer to a long integer. On return, that long integer contains the type of the hot spot specified by the hot spot ID.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function gets the type of a hot spot whose ID you specify. In combination with the `kQTVRGetHotSpotTypeSelector` intercept selector (see `QTVRInstallInterceptProc` (page 59)), this allows an application to change a hot spot's type dynamically. For example, an application can take an existing movie and cause VR to display the cursors for a type of hotspot different from the one the movie was originally authored with. In combination with intercepting `kQTVRTriggerHotSpotSelector`, this would allow an Internet plugin to override undefined or link hotspots in movies to make them appear and act as though they are URL links instead. If `kQTVRTriggerHotSpotSelector` is not intercepted, VR will attempt to act on the hotspot in the normal way; by storing both link and URL data in a file, the exact behavior can be determined at runtime by an application to allow linking to either another node locally or a remote URL link, depending on system configuration or other arbitrary considerations.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrcursors
vrcursors.win
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

QTVRGetImagingProperty

Obtains the current value of an imaging property of a movie.

```
OSErr QTVRGetImagingProperty (
   QTVRInstance qtvr,
   QTVRImagingMode imagingMode,
   UInt32 imagingProperty,
   SInt32 *propertyValue
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

imagingMode

An imaging mode (see below). See these constants:

`kQTVRStatic`
`kQTVRMotion`
`kQTVRAllModes`

imagingProperty

An imaging property (see below). See these constants:

`kQTVRImagingCorrection`
`kQTVRImagingQuality`
`kQTVRImagingDirectDraw`
`kQTVRImagingCurrentMode`

propertyValue

On entry, a pointer to a long integer. On return, that long integer contains the current value of the specified imaging property for the specified mode.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function returns, in the long integer pointed to by the `propertyValue` parameter, the current value of the property specified by the `imagingProperty` parameter when the QuickTime VR movie specified by the `qtvr` parameter is in the mode specified by the `imagingMode` parameter.

Special Considerations

`QTVRGetImagingProperty` is valid only for panoramic nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRGetInteractionProperty

Obtains the value of an interaction property.

```
OSErr QTVRGetInteractionProperty (
   QTVRInstance qtvr,
   UInt32 property,
   void *value
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

property

An interaction property type (see below). See these constants:

`kQTVRInteractionMouseClickHysteresis`
`kQTVRInteractionMouseClickTimeout`
`kQTVRInteractionPanTiltSpeed`
`kQTVRInteractionZoomSpeed`
`kQTVRInteractionTranslateOnMouseDown`
`kQTVRInteractionMouseMotionScale`
`kQTVRInteractionNudgeMode`

value

On entry, a pointer to a block of memory. On return, that memory contains the current value of the specified interaction property.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function returns, in the block of memory pointed to by the `value` parameter, the current value of the property specified by the `property` parameter for the QuickTime VR movie specified by the `qtvr` parameter. That block of memory must be large enough to hold the returned value.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRGetMouseDownTracking

Obtains the current state of mouse-down tracking.

```
Boolean QTVRGetMouseDownTracking (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

Return Value

A Boolean value that indicates whether QuickTime VR is currently handling mouse-down tracking for the QuickTime VR movie specified by the `qtvr` parameter (TRUE) or not (FALSE).

Discussion

By default, QuickTime VR tracks mouse clicks in a QuickTime VR movie and triggers hot spots as necessary.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeVR.h`

QTVRGetMouseOverTracking

Obtains the current state of mouse-over tracking.

```
Boolean QTVRGetMouseOverTracking (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

Return Value

A Boolean value that indicates whether QuickTime VR is currently handling mouse-over tracking for the QuickTime VR movie specified by the `qtvr` parameter (TRUE) or not (FALSE).

Discussion

By default, QuickTime VR tracks mouse movements in a QuickTime VR movie and changes the shape of the cursor as appropriate.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In
QuickTimeVR.h

QTVRGetNodeInfo

Obtains the node information atom container that describes a node and all the hot spots in the node.

```
OSErr QTVRGetNodeInfo (
   QTVRInstance qtvr,
   UInt32 nodeID,
   QTAtomContainer *nodeInfo
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

nodeID

A node ID. Set this parameter to kQTVRCurrentNode to receive information about the current node.

nodeInfo

On return, a pointer to an atom container that contains information about the specified node.

Return Value
See Error Codes. Returns noErr if there is no error.

Discussion
This function returns, in the nodeInfo parameter, a pointer to an atom container that contains information about the node specified by the nodeID parameter in the movie specified by the qtvr parameter. The atom container includes information about all the hot spots contained in that node. You can use the QuickTime atom functions to extract atoms from that container. You can also use those functions to access the hot spot atom list. All hot spot atoms are contained in the hot spot parent atom.

Special Considerations

The node information atom container returned by this function is a copy of the atom container maintained internally by the QuickTime VR Manager. You should dispose of the node information atom container, by calling QTDisposeAtomContainer, when you're finished using it.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrbackbuffer
vrbackbuffer.win
vrcursors
vrmakepano.win
vrmovies.win

Declared In
QuickTimeVR.h

QTVRGetNodeType

Obtains the type of a movie node.

```
OSType QTVRGetNodeType (
   QTVRInstance qtvr,
   UInt32 nodeID
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

nodeID

A node ID. Pass `kQTVRCurrentNode` for the current node.

Return Value

The type of the node specified by the `nodeID` parameter in the QuickTime VR movie specified by the `qtvr` parameter.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrbackbuffer

vrbackbuffer.win

vrmovies.win

vrscript

vrscript.win

Declared In

`QuickTimeVR.h`

QTVRGetPanAngle

Obtains the pan angle of a QuickTime VR movie.

```
float QTVRGetPanAngle (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

Return Value

A floating-point value that represents the current pan angle of the QuickTime VR movie specified by the `qtvr` parameter.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrbackbuffer
vrmovies.win
vrscript
vrscript.win
vrspeech

Declared In
`QuickTimeVR.h`

QTVRGetQTVRInstance

Obtains an instance of a QuickTime VR movie.

```
OSErr QTVRGetQTVRInstance (
   QTVRInstance *qtvr,
   Track qtvrTrack,
   MovieController mc
);
```

Parameters

qtvr
On return, an instance of the specified QuickTime VR movie.

qtvrTrack
A QTVR track contained in a QuickTime movie. You can obtain a reference to this track by calling `QTVRGetQTVRTrack` (page 48).

mc
An identifier for the movie controller to be associated with the new QuickTime VR movie instance. You obtain this identifier from `OpenComponent` or `OpenDefaultComponent`, or from `NewMovieController`.

Return Value
If `qtvrTrack` does not specify a QTVR track, this function returns `NIL` in the `qtvr` parameter and an error code as its function result. See `Error Codes`. Returns `noErr` if there is no error.

Discussion
You need a QuickTime VR movie instance to call most other QuickTime VR functions. This function returns, in the `qtvr` parameter, an instance of the QuickTime VR movie specified by the `qtvrTrack` parameter. Here's an example of code that gets a QTVR instance:

```
// QTVRGetQTVRInstance coding example
// See "Discovering QuickTime," page 390
QTVRInstance MyGetQTVRInstanceFromMC (MovieController mc)
{
    Track               track =NIL;
    QTVRInstance        qtvrinstance =NIL;
    Movie               movie =NIL;
    //Get the movie from the movie controller.
    movie =MCGetMovie(mc);
    if (movie !=NIL) {
```

```
        //Get the first QTVR track in the movie.
        track =QTVRGetQTVRTrack(movie, 1);
        //Get a QTVR instance for that QTVR track.
        if (track !=NIL) {
            QTVRGetQTVRInstance(qtvrinstance, track, mc);
            //Set our units to be degrees.
            if (qtvrinstance !=NIL)
                QTVRSetAngularUnits(qtvrinstance, kQTVRDegrees);
        }
    }
    return qtvrinstance;
}
```

Special Considerations

It's not necessary to dispose of a QuickTime VR movie instance.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrbackbuffer
vrbackbuffer.win
vrmovies.win
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

QTVRGetQTVRTrack

Obtains a QTVR track contained in a QuickTime movie to use in the QTVRGetQTVRInstance call.

```
Track QTVRGetQTVRTrack (
   Movie theMovie,
   SInt32 index
);
```

Parameters

`theMovie`
A QuickTime movie.

`index`
The index of the desired QTVR track.

Return Value
A track identifier for the QTVR track that has the index specified by the `index` parameter in the QuickTime movie specified by the `theMovie` parameter. If there is no such track, or the movie is not a QTVR movie, this function returns the value `NIL`.

Discussion
Here's an example of using this function to help get an instance of a QTVR movie running in a movie controller:

```
// QTVRGetQTVRTrack coding example
// See "Discovering QuickTime," page 390
QTVRInstance MyGetQTVRInstanceFromMC (MovieController mc)
{
    Track               track =NIL;
    QTVRInstance        qtvrinstance =NIL;
    Movie               movie =NIL;
    //Get the movie from the movie controller.
    movie =MCGetMovie(mc);
    if (movie !=NIL) {
        //Get the first QTVR track in the movie.
        track =QTVRGetQTVRTrack(movie, 1);
        //Get a QTVR instance for that QTVR track.
        if (track !=NIL) {
            QTVRGetQTVRInstance(qtvrinstance, track, mc);
            //Set our units to be degrees.
            if (qtvrinstance !=NIL)
                QTVRSetAngularUnits(qtvrinstance, kQTVRDegrees);
        }
    }
    return qtvrinstance;
}
```

Version Notes
Introduced in QuickTime 3 or earlier. QuickTime VR 2.1 supports files with at most one QTVR track; hence the value for the `index` parameter of a movie made with QuickTime VR 2.1 is always 1. Panorama and object movies made with QuickTime VR version 1.0 have no QTVR track. This function returns the track ID of the panorama track for version 1.0 panorama movies and the track ID of the image video track for version 1.0 object movies.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrbackbuffer
vrbackbuffer.win
vrmovies.win
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

QTVRGetTiltAngle

Obtains the tilt angle of a QuickTime VR movie.

```
float QTVRGetTiltAngle (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

Return Value

A floating-point value that represents the current tilt angle of the QuickTime VR movie specified by the `qtvr` parameter.

Discussion

When a cylindrical panorama is zoomed all the way out (to its maximum vertical field of view), it can no longer be tilted because its entire vertical surface is exposed. Attempting to set the tilt angle will result in a `ConstraintReachedErr` error (except for the degenerate case of setting the tilt angle to its current value).

The tilt angle may not be zero when the panorama is fully zoomed out; it may be tilted by one line of pixels. The tilt angle is small in this case, typically 0.006, but its exact magnitude depends on the height of the panorama; the taller the panorama, the smaller the error.

Do not test for a tilt angle of exactly 0.0 or attempt to adjust the tilt angle of a fully zoomed-out panorama.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrbackbuffer

vrmovies.win

vrscript

vrscript.win

vrspeech

Declared In

`QuickTimeVR.h`

QTVRGetViewAnimation

Obtains the current state of view animation for an object node.

```
Boolean QTVRGetViewAnimation (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

Return Value
A Boolean value that indicates the current state of view animation for the object node specified by the `qtvr` parameter. It is TRUE if view animation is enabled, FALSE otherwise.

Special Considerations

This function is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRGetViewCenter

Obtains the view center of a QuickTime VR movie.

```
OSErr QTVRGetViewCenter (
   QTVRInstance qtvr,
   QTVRFloatPoint *viewCenter
);
```

Parameters

`qtvr`

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

`viewCenter`

On entry, a pointer to a `QTVRFloatPoint` structure. On return, that structure contains the current view center of the specified movie.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Special Considerations

This function is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRGetViewCurrentTime

Obtains the current time in the current view.

```
TimeValue QTVRGetViewCurrentTime (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

Return Value

The current time in the current view of the object node specified by the `qtvr` parameter. The returned value is always greater than or equal to 0 and less than or equal to the value returned by `QTVRGetCurrentViewDuration` (page 37).

Special Considerations

`QTVRGetViewCurrentTime` is valid only for object nodes.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrscript

vrscript.win

Declared In

`QuickTimeVR.h`

QTVRGetViewingLimits

Obtains the current viewing limits of a QuickTime VR movie.

```
OSErr QTVRGetViewingLimits (
   QTVRInstance qtvr,
   UInt16 kind,
   float *minValue,
   float *maxValue
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

kind

The type of viewing limits to be returned (see below). See these constants:

`kQTVRPan`

`kQTVRTilt`

`kQTVRFieldOfView`

minValue

On entry, a pointer to a floating-point value. On return, the minimum viewing limit of the specified type is copied into that value.

maxValue

On entry, a pointer to a floating-point value. On return, the maximum viewing limit of the specified type is copied into that value.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function returns, in the floating-point values pointed to by the `minValue` and `maxValue` parameters, the current minimum and maximum values for angles whose type is specified by the `kind` parameter. The maximum field of view of a panoramic node can be limited by the size of the back buffer and the current aspect ratio of the movie's graphics world. The values returned by this function are unaffected by the current control settings.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRGetViewParameter

Undocumented

```
OSErr QTVRGetViewParameter (
   QTVRInstance qtvr,
   UInt32 viewParameter,
   void *value,
   UInt32 flagsIn,
   UInt32 *flagsOut
);
```

Parameters

qtvr

Undocumented

viewParameter

Undocumented

value

Undocumented

flagsIn

Undocumented

flagsOut

Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeVR.h

QTVRGetViewRate

Obtains the current view rate of an object node.

```
float QTVRGetViewRate (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

Return Value
The current view rate of the object node specified by the qtvr parameter. A view rate is a floating-point value in the range from -100.0 to +100.0. An object node's default view rate is stored in the movie file.

Special Considerations

This function is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
QuickTimeVR.h

QTVRGetViewState

Obtains the value of a view state.

```
OSErr QTVRGetViewState (
   QTVRInstance qtvr,
   QTVRViewStateType viewStateType,
   UInt16 *state
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

viewStateType

A view state type (see below). See these constants:

`kQTVRDefault`

`kQTVRCurrent`

`kQTVRMouseDown`

state

On entry, a pointer to a short integer. On return, that integer is set to the current value of the specified type of view state.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Special Considerations

This function is valid only for object nodes.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeVR.h`

QTVRGetViewStateCount

Obtains the number of view states of an object node.

```
UInt16 QTVRGetViewStateCount (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

Return Value

The number of view states associated with the object node specified by the `qtvr` parameter. The number of view states in an object movie is defined by the movie file.

Special Considerations

This function is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRGetVisible

Obtains a movie's visibility state.

```
Boolean QTVRGetVisible (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

Return Value
A Boolean value that indicates whether the QuickTime VR movie specified by the `qtvr` parameter is visible (TRUE) or not (FALSE).

Special Considerations

This function is valid only for panoramic nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRGetVisibleHotSpots

Obtains a list of the currently visible hot spots in a QuickTime VR movie.

```
UInt32 QTVRGetVisibleHotSpots (
   QTVRInstance qtvr,
   Handle hotSpots
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

hotSpots

On entry, a valid handle to a block of memory. On return, that block of memory is filled with a list of the IDs of the visible hot spots in the specified QuickTime VR movie. If necessary, the handle is resized to hold all the hot spot IDs. Accordingly, the handle must be unlocked at the time you call this function.

Return Value
The number of hot spot IDs returned though the hotSpots parameter.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

QTVRGetVRWorld

Obtains the VR world atom container for a movie.

```
OSErr QTVRGetVRWorld (
   QTVRInstance qtvr,
   QTAtomContainer *VRWorld
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

VRWorld

On return, a pointer to an atom container that contains information about the specified movie.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function returns, in the `VRWorld` parameter, a pointer to an atom container that contains general scene information about the QuickTime VR movie specified by the `qtvr` parameter, as well as a list of all the nodes in that movie. You can use the QuickTime atom functions to extract atoms from that container.

Special Considerations

The VR world atom container returned by this function is a copy of the atom container maintained internally by the QuickTime VR Manager. You should dispose of the VR world atom container by calling `QTDisposeAtomContainer` when you're finished using it.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrbackbuffer
vrbackbuffer.win
vrcursors
vrmakepano.win
vrmovies.win

Declared In
`QuickTimeVR.h`

QTVRGoToNodeID

Sets the current node of a movie.

```
OSErr QTVRGoToNodeID (
   QTVRInstance qtvr,
   UInt32 nodeID
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

nodeID

The ID of the node you want to be the current node. The QuickTime VR Manager defines several constants (see below) for specific nodes. See these constants:

`kQTVRCurrentNode`
`kQTVRPreviousNode`
`kQTVRDefaultNode`

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Special Considerations

Setting the current node also sets the pan, tilt, and field of view of the new current node to their default values. As a result, if you wish to set non-default angles, you should call this function before you call `QTVRSetPanAngle` (page 89), `QTVRSetTiltAngle` (page 92), or `QTVRSetFieldOfView` (page 82).

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

QTVRInstallInterceptProc

Installs or removes an intercept procedure for a QuickTime VR Manager function.

```
OSErr QTVRInstallInterceptProc (
   QTVRInstance qtvr,
   QTVRProcSelector selector,
   QTVRInterceptUPP interceptProc,
   SInt32 refCon,
   UInt32 flags
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

selector

A selector (see below) that indicates which QuickTime VR function to intercept. See these constants:

`kQTVRSetPanAngleSelector`
`kQTVRSetTiltAngleSelector`
`kQTVRSetFieldOfViewSelector`
`kQTVRSetViewCenterSelector`
`kQTVRMouseEnterSelector`
`kQTVRMouseWithinSelector`
`kQTVRMouseLeaveSelector`
`kQTVRMouseDownSelector`
`kQTVRMouseStillDownSelector`
`kQTVRMouseUpSelector`
`kQTVRTriggerHotSpotSelector`
`kQTVRGetHotSpotTypeSelector`

interceptProc

A Universal Procedure Pointer for a `QTVRInterceptProc` callback. Set this parameter to `NIL` to remove a previously installed intercept procedure.

refCon

A reference constant to be passed to your intercept callback. Use this parameter to point to a data structure containing any information your callback needs.

flags

Unused. Set this parameter to 0.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function installs the procedure specified by the `interceptProc` parameter as an intercept procedure for the QuickTime VR function specified by the `selector` parameter for the QuickTime VR movie specified by the `qtvr` parameter. Your intercept procedure is called whenever QuickTime VR is about to execute the function you are intercepting.

Your procedure can simply replace the intercepted function, by returning TRUE in the `cancel` parameter; or it can call through to the intercepted function, by calling `QTVRCallInterceptedProc` (page 22); or it can allow the intercepted function to execute when the intercept procedure returns, by returning FALSE in the `cancel` parameter.

Here's an example of using this function:

```
// QTVRInstallInterceptProc coding example
// See "Discovering QuickTime," page 398
QTVRInterceptUPP MyInstallInterceptProcedure (QTVRInstance qtvrinstance)
{
    QTVRInterceptUPP    lpfnIntercept;
    lpfnIntercept =NewQTVRInterceptProc(MyInterceptProc);
    QTVRInstallInterceptProc(qtvrinstance, kQTVRSetPanAngleSelector,
                                            lpfnIntercept, 0, 0);
    return lpfnIntercept
}
```

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win
vrspeech

Declared In
`QuickTimeVR.h`

QTVRInteractionNudge

Translates the image and displays the new view or rotates the object in a particular direction and displays its new appearance.

```
OSErr QTVRInteractionNudge (
   QTVRInstance qtvr,
   QTVRNudgeControl direction
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

direction

The direction of the nudge (see below). The type of adjustment depends on the nudge interaction mode, which you can set with `QTVRSetInteractionProperty` (page 85). If the nudge interaction mode is `kQTVRNudgeRotate`, the action of `QTVRInteractionNudge` is to rotate the object in the specified direction. If the nudge interaction mode is `kQTVRNudgeTranslate`, the action of `QTVRInteractionNudge` is to translate the image in the specified direction. If the nudge interaction mode is `kQTVRUNudgeSameAsMouse`, the action of `QTVRInteractionNudge` is determined by the current mouse mode, which you can determine by calling `QTVRGetCurrentMouseMode` (page 36). See these constants:

`kQTVRRight`
`kQTVRUpRight`
`kQTVRUp`
`kQTVRUpLeft`
`kQTVRLeft`
`kQTVRDown`
`kQTVRDownRight`

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function adjusts the current view of the movie specified by the `qtvr` parameter as indicated by the `direction` parameter. The type of adjustment depends on the property setting for nudge interaction mode, which you can set with `QTVRSetInteractionProperty` (page 85).

Special Considerations

This function is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRMouseDown

Handles the user's clicking the mouse button when the cursor is in a QuickTime VR movie for which mouse-down tracking is disabled.

```
OSErr QTVRMouseDown (
   QTVRInstance qtvr,
   Point pt,
   UInt32 when,
   UInt16 modifiers,
   UInt32 *hotSpotID,
   WindowRef w
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

pt

The current location of the cursor, in the local coordinates of the graphics world specified by the `w` parameter.

when

The time, in the number of ticks (sixtieths of a second) since system startup, when the mouse-down event was posted.

modifiers

A short integer, each bit of which is represented by a constant (see below) that provides information about the state of the modifier keys and the mouse button at the time the event was posted. More than one bit may be set. See these constants:

hotSpotID

On entry, a pointer to a long integer. On return, that long integer contains the ID of the hot spot that lies beneath the specified point, or the value 0 if no hot spot lies beneath that point.

w

A pointer to a graphics world.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function returns, in the long integer pointed to by the `hotSpotID` parameter, the ID of the hot spot in the QuickTime VR movie specified by the `qtvr` parameter that lies directly under the point specified by the `pt` parameter. If no hot spot lies under that point, the long integer is set to 0. `QTVRMouseDown` also performs any other tasks that are typically performed when the user clicks the mouse button when the cursor is in a QuickTime VR movie.

Special Considerations

You need to call `QTVRMouseDown` only if you have disabled mouse-down tracking for the specified QuickTime VR movie.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeVR.h`

QTVRMouseEnter

Handles the user's moving the cursor into a QuickTime VR movie for which mouse-over tracking is disabled.

```
OSErr QTVRMouseEnter (
   QTVRInstance qtvr,
   Point pt,
   UInt32 *hotSpotID,
   WindowRef w
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

pt

The current location of the cursor, in the local coordinates of the graphics world specified by the `w` parameter.

hotSpotID

On entry, a pointer to a long integer. On return, that long integer contains the ID of the hot spot that lies beneath the specified point, or the value 0 if no hot spot lies beneath that point.

w

A pointer to a graphics world.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function returns, in the long integer pointed to by the `hotSpotID` parameter, the ID of the hot spot in the QuickTime VR movie specified by the `qtvr` parameter that lies directly under the point specified by the `pt` parameter. If no hot spot lies under that point, the long integer is set to 0. `QTVRMouseEnter` also performs any other tasks that are typically performed when the user first moves the cursor into a QuickTime VR movie.

Special Considerations

You need to call `QTVRMouseEnter` only if you have disabled mouse-over tracking for the specified QuickTime VR movie.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeVR.h`

QTVRMouseLeave

Handles the user's moving the cursor out of a QuickTime VR movie for which mouse-over tracking is disabled.

```
OSErr QTVRMouseLeave (
   QTVRInstance qtvr,
   Point pt,
   WindowRef w
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

pt

The current location of the cursor, in the local coordinates of the graphics world specified by the `w` parameter.

w

A pointer to a graphics world.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function performs any tasks that are typically performed when the user moves the cursor out of a QuickTime VR movie.

Special Considerations

You need to call `QTVRMouseLeave` only if you have disabled mouse-over tracking for the specified QuickTime VR movie.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeVR.h`

QTVRMouseStillDown

Handles the user's holding down the mouse button while the cursor is in a QuickTime VR movie for which mouse-down tracking is disabled.

```
OSErr QTVRMouseStillDown (
   QTVRInstance qtvr,
   Point pt,
   UInt32 *hotSpotID,
   WindowRef w
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

pt

The current location of the cursor, in the local coordinates of the graphics world specified by the `w` parameter.

hotSpotID

On entry, a pointer to a long integer. On return, that long integer contains the ID of the hot spot that lies beneath the specified point, or the value 0 if no hot spot lies beneath that point.

w

A pointer to a graphics world.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function returns, in the long integer pointed to by the `hotSpotID` parameter, the ID of the hot spot in the QuickTime VR movie specified by the `qtvr` parameter that lies directly under the point specified by the `pt` parameter. If no hot spot lies under that point, the long integer is set to 0. This function also performs any other tasks that are typically performed when the user holds down the mouse button when the cursor is in a QuickTime VR movie. You should call this function repeatedly for as long as the user holds down the mouse button while the cursor is in the specified QuickTime VR movie.

Special Considerations

You need to call this function only if you have disabled mouse-down tracking for the specified QuickTime VR movie. Applications running on operating systems other than Mac OS should use the extended form of this function, `QTVRMouseStillDownExtended` (page 65).

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRMouseStillDownExtended

Handles the user's holding down the mouse button while the cursor is in a QuickTime VR movie for which mouse-down tracking is disabled.

```
OSErr QTVRMouseStillDownExtended (
   QTVRInstance qtvr,
   Point pt,
   UInt32 *hotSpotID,
   WindowRef w,
   UInt32 when,
   UInt16 modifiers
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

pt

The current location of the cursor, in the local coordinates of the graphics world specified by the `w` parameter.

hotSpotID

On entry, a pointer to a long integer. On return, that long integer contains the ID of the hot spot that lies beneath the specified point, or the value 0 if no hot spot lies beneath that point.

w

A pointer to a graphics world.

when

The current time as the number of ticks (sixtieths of a second) since system startup.

modifiers

A short integer, each bit of which is represented by a constant (see below) that provides information about the state of the modifier keys and the mouse button at the time the event was posted. More than one bit may be set. See these constants:

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function uses the same intercept as `QTVRMouseStillDown` (page 64) but has two additional parameters. Applications that intercept this function should always check the `paramCount` field to make sure it is 5 before using the last two fields. You should call this function repeatedly for as long as the user holds down the mouse button while the cursor is in the specified QuickTime VR movie.

Special Considerations

You need to call this function only if you have disabled mouse-down tracking for the specified QuickTime VR movie. Internally, QuickTime VR always uses this function instead of `QTVRMouseStillDown`. Developers implementing their own mouse down tracking don't need to use the extended version unless they also intercept the procedure and need the added parameters.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRMouseUp

Handles the user's releasing the mouse button while the cursor is in a QuickTime VR movie for which mouse-down tracking is disabled.

```
OSErr QTVRMouseUp (
   QTVRInstance qtvr,
   Point pt,
   UInt32 *hotSpotID,
   WindowRef w
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

pt

The current location of the cursor, in the local coordinates of the graphics world specified by the `w` parameter.

hotSpotID

On entry, a pointer to a long integer. On return, that long integer contains the ID of the hot spot that lies beneath the specified point, or the value 0 if no hot spot lies beneath that point.

w

A pointer to a graphics world.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function returns, in the long integer pointed to by the `hotSpotID` parameter, the ID of the hot spot in the QuickTime VR movie specified by the `qtvr` parameter that lies directly under the point specified by the `pt` parameter. If no hot spot lies under that point, the long integer is set to 0. this function also performs any other tasks that are typically performed when the user releases the mouse button after clicking it when the cursor is in a QuickTime VR movie.

Special Considerations

You need to call this function only if you have disabled mouse-down tracking for the specified QuickTime VR movie. Applications running on operating systems other than Mac OS should use the extended form of this function, `QTVRMouseUpExtended` (page 67).

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeVR.h`

QTVRMouseUpExtended

Handles the user's releasing the mouse button while the cursor is in a QuickTime VR movie for which mouse-down tracking is disabled.

```
OSErr QTVRMouseUpExtended (
   QTVRInstance qtvr,
   Point pt,
   UInt32 *hotSpotID,
   WindowRef w,
   UInt32 when,
   UInt16 modifiers
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

pt

The current location of the cursor, in the local coordinates of the graphics world specified by the `w` parameter.

hotSpotID

On entry, a pointer to a long integer. On return, that long integer contains the ID of the hot spot that lies beneath the specified point, or the value 0 if no hot spot lies beneath that point.

w

A pointer to a graphics world.

when

The time, in the number of ticks (sixtieths of a second) since system startup, when the mouse-up event was posted.

modifiers

A short integer, each bit of which is represented by a constant (see below) that provides information about the state of the modifier keys and the mouse button at the time the event was posted. More than one bit may be set. See these constants:

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function uses the same intercept as the `QTVRMouseUp` function but has two additional parameters. Applications that intercept this function should always check the `paramCount` field to make sure it is 5 before using the last two fields.

Special Considerations

You need to call `QTVRMouseUp` (page 66) or this function only if you have disabled mouse-down tracking for the specified QuickTime VR movie. Internally, QuickTime VR always uses the this function function instead of `QTVRMouseUp`. Developers implementing their own mouse down tracking don't need to use the extended version unless they also intercept the procedure and need the added parameters.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeVR.h`

QTVRMouseWithin

Handles the user's leaving the cursor in a QuickTime VR movie for which mouse-over tracking is disabled.

```
OSErr QTVRMouseWithin (
   QTVRInstance qtvr,
   Point pt,
   UInt32 *hotSpotID,
   WindowRef w
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

pt

The current location of the cursor, in the local coordinates of the graphics world specified by the `w` parameter.

hotSpotID

On entry, a pointer to a long integer. On return, that long integer contains the ID of the hot spot that lies beneath the specified point, or the value 0 if no hot spot lies beneath that point.

w

A pointer to a graphics world.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

You should call this function repeatedly for as long as the cursor remains in the specified QuickTime VR movie.

Special Considerations

You need to call `QTVRMouseWithin` only if you have disabled mouse-over tracking for the specified QuickTime VR movie.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeVR.h`

QTVRNudge

Turns one step in a particular direction and displays the new view.

```
OSErr QTVRNudge (
   QTVRInstance qtvr,
   QTVRNudgeControl direction
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

direction

The direction of the nudge (see below). Any value of the `direction` parameter that is not predefined is mapped to the closest defined value. See these constants:

`kQTVRRight`
`kQTVRUpRight`
`kQTVRUp`
`kQTVRUpLeft`
`kQTVRLeft`
`kQTVRDown`
`kQTVRDownRight`

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function adjusts the current view of the movie specified by the `qtvr` parameter as indicated by the `direction` parameter. In particular, it turns one step in the indicated direction and displays the new view. For example, to move to the next view that is right and up from the current view, set the `direction` parameter to `kQTVRUpRight` (that is, pi/4 radians, or 45 degrees). For objects, if no view is located at the adjacent object view defined by the nudge direction and wrapping is off in the desired direction, then this function remains at the current view and returns the result code `constraintReachedErr`. For objects, this function is useful for changing to an adjacent view without having to know the new pan and tilt angles. The direction of the nudge is affected by the current control settings.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrscript
vrscript.win

Declared In

`QuickTimeVR.h`

QTVRPanToColumn

Obtains the column number in the object image array that corresponds to a pan angle.

```
short QTVRPanToColumn (
   QTVRInstance qtvr,
   float panAngle
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

panAngle

A pan angle.

Return Value

The zero-based column number in the current object image array that corresponds to the pan angle specified by the `panAngle` parameter

Special Considerations

This function is valid only for object nodes.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`QuickTimeVR.h`

QTVRPtToAngles

Obtains the pan and tilt angles of a point.

```
OSErr QTVRPtToAngles (
   QTVRInstance qtvr,
   Point pt,
   float *panAngle,
   float *tiltAngle
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

pt

A point, in the local coordinates of the graphics world of the specified movie.

panAngle

On entry, a pointer to a floating-point value. On return, that value contains the pan angle of the specified point.

tiltAngle

On entry, a pointer to a floating-point value. On return, that value contains the tilt angle of the specified point.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
For a panorama, each point in the current view corresponds to a particular pan and tilt angle, with the point at the center of the view corresponding to the panorama's current pan and tilt angle. This function returns, in the floating-point values pointed to by the `panAngle` and `tiltAngle` parameters, the pan and tilt angles of the point specified by the `pt` parameter.

Special Considerations

This function is valid only for panoramic nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRPtToHotSpotID

Obtains the ID of the hot spot, if any, that lies beneath a point.

```
OSErr QTVRPtToHotSpotID (
   QTVRInstance qtvr,
   Point pt,
   UInt32 *hotSpotID
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

pt

A point, in the local coordinates of the graphics world of the specified movie.

hotSpotID

On entry, a pointer to a long integer. On return, that long integer contains the ID of the hot spot that lies beneath the specified point, or the value 0 if no hot spot lies beneath that point.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRRefreshBackBuffer

Refreshes the QTVR back buffer.

```
OSErr QTVRRefreshBackBuffer (
   QTVRInstance qtvr,
   UInt32 flags
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

flags

Unused. Set this parameter to 0.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function refreshes some or all of the back buffer associated with the QuickTime VR movie specified by the `qtvr` parameter by reloading the appropriate data from the diced frames in the panorama image track. You can call this function either in a back buffer imaging procedure or elsewhere in your application. If you call this function in a back buffer imaging procedure, only the current rectangle (that is, the rectangle specified by the procedure's `drawRect` parameter) is refreshed. If you call this function outside of a back buffer imaging procedure, all areas of interest specified in the most recent call to `QTVRSetBackBufferImagingProc` (page 77) are refreshed.

Special Considerations

This function is valid only for panoramic nodes.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrbackbuffer

vrmovies

vrmovies.win

vrscript

vrscript.win

Declared In

`QuickTimeVR.h`

QTVRReplaceCursor

Replaces any of the standard QuickTime VR cursors with your own custom cursor.

```
OSErr QTVRReplaceCursor (
   QTVRInstance qtvr,
   QTVRCursorRecord *cursRecord
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

cursRecord

A pointer to a `QTVRCursorRecord` structure.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function replaces one or more of the standard QuickTime VR cursors associated with the instance specified by the `qtvr` parameter with the cursors specified in the cursor record pointed to by the `cursRecord` parameter. If the `type` field of the specified cursor record is `kQTVRUseDefaultCursor`, the default cursor for the given resource ID is reloaded; in this case, the `handle` field of that record should be set to `NIL`.

Special Considerations

This function replaces the standard cursors only for the specified QuickTime VR movie instance. To replace the standard cursors for all QuickTime VR movie instances you create, you need to call this function for each such instance.

Version Notes

Introduced in QuickTime 3 or earlier. Note that QuickTime VR 2.1 makes a copy of the cursor handle specified in the cursor record. The application is responsible for disposing of its own cursor handle.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrcursors

vrcursors.win

vrscript

vrscript.win

Declared In

`QuickTimeVR.h`

QTVRRowToTilt

Obtains the tilt angle that corresponds to a row number in a QTVR object image array.

```
float QTVRRowToTilt (
   QTVRInstance qtvr,
   short row
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

row

A row number.

Return Value

The tilt angle that corresponds to the zero-based row number in the object image array specified by the row parameter.

Special Considerations

This function is valid only for object nodes.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

QuickTimeVR.h

QTVRSetAngularUnits

Sets the type of unit used when specifying QTVR angles.

```
OSErr QTVRSetAngularUnits (
   QTVRInstance qtvr,
   QTVRAngularUnits units
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

units

A constant (see below) that indicates the type of angular units to use. See these constants:

kQTVRDegrees

kQTVRRadians

Return Value

See Error Codes. Returns noErr if there is no error.

Discussion

This function sets the type of angular units to be used in all subsequent QuickTime VR Manager calls for the QuickTime VR movie specified by the qtvr parameter to the unit type specified by the units parameter.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrbackbuffer
vrbackbuffer.win
vrmakepano.win
vrmovies.win
vrspeech

Declared In
QuickTimeVR.h

QTVRSetAnimationSetting

Sets the state of an animation setting for an object node.

```
OSErr QTVRSetAnimationSetting (
   QTVRInstance qtvr,
   QTVRObjectAnimationSetting setting,
   Boolean enable
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

setting

An animation setting (see below). See these constants:

kQTVRPalindromeViewFrames
kQTVRDontLoopViewFrames
kQTVRPlayEveryViewFrame
kQTVRSyncViewToFrameRate
kQTVRPalindromeViews
kQTVRPlayStreamingViews

enable

A Boolean value that indicates whether the specified animation setting is to be enabled for the specified object node (TRUE) or disabled (FALSE).

Return Value
See Error Codes. Returns noErr if there is no error.

Special Considerations

This function is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

QTVRSetBackBufferImagingProc

Installs or removes a QTVR back buffer imaging procedure.

```
OSErr QTVRSetBackBufferImagingProc (
   QTVRInstance qtvr,
   QTVRBackBufferImagingUPP backBufferImagingProc,
   UInt16 numAreas,
   QTVRAreaOfInterest areasOfInterest[],
   SInt32 refCon
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

backBufferImagingProc

A Universal Procedure Pointer for a `QTVRBackBufferImagingProc` callback. To remove a previously installed back buffer imaging procedure, pass `NIL`.

numAreas

The number of area of interest structures in the array pointed to by the `areasOfInterest` parameter.

areasOfInterest[]

A pointer to an array of `QTVRAreaOfInterest` structures. Each structure defines a rectangular areas about which you want your back buffer imaging procedure to be notified. Your procedure is called for each area of interest as it becomes visible or not visible. You indicate when you want your procedure to be called for a particular area of interest by setting flags in the `flags` field in the corresponding area of interest structure. The width of the area of interest is limited by the size of the back buffer. If the back buffer is less than the full cache size, then the area of interest can be no wider than half the size of the back buffer. (For vertical cylinder geometries, limiting factor would be the height of the buffer.) For a full cache back buffer, the width of the area of interest can be the full size of the buffer. If the width limit is exceeded, this function returns `constraintReachedErr`.

refCon

A reference constant. This value is passed to the specified back buffer imaging callback. Use this parameter to point to a data structure containing any information your callback needs.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function installs the procedure specified by the `backBufferImagingProc` parameter as a back buffer imaging procedure for the panoramic node specified by the `qtvr` parameter. You can use that procedure to draw directly into the back buffer. Coordinates in the back buffer are dependent on the current correction

mode; as a result, you need to indicate the area you're interested in drawing into by specifying a pan angle and tilt angle to determine the upper-left corner of the area and a height and width relative to that corner. Specifying a height and width instead of a second pair of pan and tilt angles for the bottom-right coordinate allows the rectangle to wrap around the edge of the panorama.

Special Considerations

This function is valid only for panoramic nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrbackbuffer
vrmovies
vrmovies.win
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

QTVRSetBackBufferPrefs

Sets the resolution, pixel format, and size of the back buffer maintained internally by QuickTime VR for caching a panoramic image in a particular pixel format.

```
OSErr QTVRSetBackBufferPrefs (
   QTVRInstance qtvr,
   UInt32 geometry,
   UInt16 resolution,
   UInt32 cachePixelFormat,
   SInt16 cacheSize
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

geometry

The type and orientation of the panorama data (see below). See these constants:

`kQTVRUseMovieGeometry`
`kQTVRVerticalCylinder`

resolution

The desired image resolution (see below). See these constants:

`kQTVRDefaultRes`
`kQTVRFullRes`
`kQTVRHalfRes`
`kQTVRQuarterRes`

cachePixelFormat

The desired pixel format for the back buffer (see below). See these constants:

cacheSize

The desired size for the panorama back buffer (see below). See these constants:

```
kQTVRMinimumCache
kQTVRSuggestedCache
kQTVRFullCache
```

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function sets the resolution, pixel format, and size of the panorama back buffer for the movie specified by the `qtvr` parameter to the values specified by the `resolution` parameter an the `cachePixelFormat` and `cacheSize` parameters. You can specify a resolution that isn't contained in the movie file; if you do so, QuickTime VR takes the highest resolution image in the file and reduces it to fit into the specified buffer size. If you specify an unsupported pixel format, this function may return an error.

Special Considerations

This function is valid only for panoramic nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

QTVRSetConstraints

Sets the constraints of a VR movie.

```
OSErr QTVRSetConstraints (
   QTVRInstance qtvr,
   UInt16 kind,
   float minValue,
   float maxValue
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

kind

The type of constraint to set (see below). See these constants:

kQTVRPan

kQTVRTilt

kQTVRFieldOfView

minValue

A floating-point value that contains the desired minimum constraint of the specified type.

maxValue

A floating-point value that contains the desired maximum constraint of the specified type.

Return Value

See Error Codes. Returns noErr if there is no error.

Discussion

This function sets the minimum and maximum constraints of the type specified by the kind parameter to the values specified by the minValue and maxValue parameters. Note that when you want to specify a pan angle constraint, the minValue and maxValue parameters should be specified so that a clockwise sweep from minValue to maxValue selects the desired angular expanse. For example, to constrain panning in the 90-degree expanse that spreads out 45 degrees on each side of the pan angle 0 degrees, you should set the minValue parameter to 315 degrees and the maxValue parameter to 45 degrees. Similarly, to constrain panning in the remaining 270-degree expanse, you should set the minValue parameter to 45 degrees and the maxValue parameter to 315 degrees.

Special Considerations

The values passed to this function are unaffected by the current control settings.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrscript

vrscript.win

Declared In

QuickTimeVR.h

QTVRSetControlSetting

Sets the state of a control setting for a QTVR object node.

```
OSErr QTVRSetControlSetting (
   QTVRInstance qtvr,
   QTVRControlSetting setting,
   Boolean enable
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

setting

A control setting (see below). See these constants:

`kQTVRWrapPan`
`kQTVRWrapTilt`
`kQTVRCanZoom`
`kQTVRReverseHControl`
`kQTVRReverseVControl`
`kQTVRSwapHVControl`
`kQTVRTranslation`

enable

A Boolean value that indicates whether the specified control setting is to be enabled for the specified object node (TRUE) or disabled (FALSE).

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Special Considerations

This function is valid only for object nodes.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrscript

vrscript.win

Declared In

`QuickTimeVR.h`

QTVRSetEnteringNodeProc

Installs or removes a node-entering procedure.

```
OSErr QTVRSetEnteringNodeProc (
   QTVRInstance qtvr,
   QTVREnteringNodeUPP enteringNodeProc,
   SInt32 refCon,
   UInt32 flags
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

enteringNodeProc

A Universal Procedure Pointer for a `QTVREnteringNodeProc` callback. To remove a previously installed `QTVREnteringNodeProc` callback, set `enteringNodeProc` to `NIL`.

refCon

A reference constant. This value is passed to the specified node-entering callback. Use this parameter to point to a data structure containing any information your callback needs.

flags

Unused. Set this parameter to 0.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function installs the procedure specified by the `enteringNodeProc` parameter as a node-entering procedure for the QuickTime VR movie specified by the `qtvr` parameter. Your procedure is called whenever a node is entered (either in response to user actions or in response to QuickTime VR Manager functions that change nodes). The reference constant specified by the `refCon` parameter is passed unchanged to that node-entering procedure.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrscript

vrscript.win

vrspeech

Declared In

`QuickTimeVR.h`

QTVRSetFieldOfView

Sets the vertical field of view of a QuickTime VR movie.

```
OSErr QTVRSetFieldOfView (
   QTVRInstance qtvr,
   float fieldOfView
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

fieldOfView

The desired vertical field of view for the specified movie. This value is constrained by the maximum field of view of the movie. Values that lie outside that limit are clipped to the maximum. Pan and tilt angle values are also clipped if, when combined with the current field of view, they would cause an image to lie outside the current constraints.

Return Value

See `Error Codes`. Returns `noErr` if there is no error. If the control setting `kQTVRCanZoom` is disabled, the field of view is unchanged and this function returns the result code `constraintReachedErr`. You can use `QTVRSetControlSetting` (page 80) to control the setting of `kQTVRCanZoom`.

Special Considerations

The pan and tilt angles are subject to the current pan and tilt range constraints, as imposed by the viewing limits and the current field of view. Accordingly, if you want to change the field of view, you should do so before adjusting the pan or tilt angles. Otherwise, the pan and tilt angles are clipped against the current field of view, which may result in an incorrect view when you alter the field of view.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrscript

vrscript.win

vrspeech

Declared In

`QuickTimeVR.h`

QTVRSetFrameRate

Sets the frame rate of an object node.

```
OSErr QTVRSetFrameRate (
   QTVRInstance qtvr,
   float rate
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

rate

The desired frame rate of the specified movie. A frame rate is a floating-point value in the range from -100.0 to +100.0. Positive values indicate forward rates, and negative values indicate reverse rates. Set this parameter to 0 to stop the movie. If the value specified lies outside the valid range, this function returns the result code `constraintReachedErr` and sets the frame rate to the nearest constraint.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function is most useful when an object is being viewed with a looping animation. (The current view of the object may contain frames that are played in a loop, as specified by the file format.) You can use this function to change the frame rate of the loop.

Special Considerations

This function is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript

vrscript.win

Declared In
`QuickTimeVR.h`

QTVRSetImagingProperty

Sets the value of an imaging property of a movie.

```
OSErr QTVRSetImagingProperty (
   QTVRInstance qtvr,
   QTVRImagingMode imagingMode,
   UInt32 imagingProperty,
   SInt32 propertyValue
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

imagingMode

An imaging mode (see below). See these constants:

`kQTVRStatic`

`kQTVRMotion`

`kQTVRAllModes`

imagingProperty

An imaging property (see below). See these constants:

kQTVRImagingCorrection
kQTVRImagingQuality
kQTVRImagingDirectDraw
kQTVRImagingCurrentMode

propertyValue

The desired value for the specified imaging property for the specified mode.

Return Value

See Error Codes. Returns noErr if there is no error.

Discussion

Default values for all imaging properties can be contained in a QuickTime VR movie file. If no defaults are specified in a movie file, the QuickTime VR Manager uses these values: for static mode, the kQTVRImagingQuality property is codecHighQuality and kQTVRImagingDirectDraw is TRUE; for motion mode, the kQTVRImagingQuality property is codecLowQuality and kQTVRImagingDirectDraw is TRUE. The default correction mode is kQTVRFullCorrection for both static and motion imaging modes. Note that it would look strange to have one correction mode for static imaging and a different correction mode for motion imaging. As a result, you should typically set the imagingMode parameter to kQTVRAllModes when setting a property of type kQTVRImagingCorrection.

Special Considerations

This function is valid only for panoramic nodes.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrscript
vrscript.win

Declared In

QuickTimeVR.h

QTVRSetInteractionProperty

Sets the value of an interaction property.

```
OSErr QTVRSetInteractionProperty (
   QTVRInstance qtvr,
   UInt32 property,
   void *value
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

property

An interaction property type (see below). See these constants:

```
kQTVRInteractionMouseClickHysteresis
kQTVRInteractionMouseClickTimeout
kQTVRInteractionPanTiltSpeed
kQTVRInteractionZoomSpeed
kQTVRInteractionTranslateOnMouseDown
kQTVRInteractionMouseMotionScale
kQTVRInteractionNudgeMode
```

value

The desired value of the specified interaction property.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function sets the value of the interaction property of the type specified by the `property` parameter for the movie specified by the `qtvr` parameter to the value specified by the `value` parameter. For types that occupy 32 or fewer bits of memory, you pass the desired value itself (cast to a `void*` type) in the `value` parameter. For structures and floating-point values, you must pass a pointer to the desired value in the `value` parameter. Note that floating-point values are usually stored as 32-bit values, but compilers differ in how they pass floating-point values as parameters; as a result, this function demands that floating-point values always be passed by reference.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrscript

vrscript.win

Declared In

`QuickTimeVR.h`

QTVRSetLeavingNodeProc

Installs or removes a node-leaving procedure.

```
OSErr QTVRSetLeavingNodeProc (
   QTVRInstance qtvr,
   QTVRLeavingNodeUPP leavingNodeProc,
   SInt32 refCon,
   UInt32 flags
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

leavingNodeProc

A Universal Procedure Pointer for a QTVRLeavingNodeProc callback. To remove a previously installed QTVRLeavingNodeProc callback, set leavingNodeProc to NIL.

refCon

A reference constant. This value is passed to the specified node-leaving callback. Use this parameter to point to a data structure containing any information your callback needs.

flags

Unused. Set this parameter to 0.

Return Value
See Error Codes. Returns noErr if there is no error.

Discussion
This function installs the procedure specified by the leavingNodeProc parameter as a node-leaving procedure for the QuickTime VR movie specified by the qtvr parameter. Your procedure is called whenever a node is left (either in response to user actions or in response to QuickTime VR Manager functions that change nodes). The reference constant specified by the refCon parameter is passed unchanged to that node-leaving procedure.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
QuickTimeVR.h

QTVRSetMouseDownTracking

Sets the state of mouse-down tracking.

```
OSErr QTVRSetMouseDownTracking (
   QTVRInstance qtvr,
   Boolean enable
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

enable

A Boolean value that indicates whether QuickTime VR should handle mouse-down tracking for the specified movie (TRUE) or not (FALSE).

Return Value
See Error Codes. Returns noErr if there is no error.

Discussion
By default, QuickTime VR tracks mouse clicks in a QuickTime VR movie and triggers hot spots as appropriate. If you disable mouse-down tracking (by passing FALSE in the `enable` parameter), you must call `QTVRMouseDown` (page 61), `QTVRMouseStillDown` (page 64), and `QTVRMouseUp` (page 66) at the appropriate times to handle user actions.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRSetMouseOverHotSpotProc

Installs or removes a mouse over hot spot procedure.

```
OSErr QTVRSetMouseOverHotSpotProc (
   QTVRInstance qtvr,
   QTVRMouseOverHotSpotUPP mouseOverHotSpotProc,
   SInt32 refCon,
   UInt32 flags
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

mouseOverHotSpotProc

A Universal Procedure Pointer for a `QTVRMouseOverHotSpotProc` callback. To remove a previously installed `QTVRMouseOverHotSpotUPP` callback, set `mouseOverHotSpotProc` to `NIL`.

refCon

A reference constant. This value is passed to the specified mouse over hot spot callback. Use this parameter to point to a data structure containing any information your callback needs.

flags

Unused. Set this parameter to 0.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function installs the routine specified by the `mouseOverHotSpotProc` parameter as a mouse over hot spot procedure for the QuickTime VR movie specified by the `qtvr` parameter. Subsequent user actions (such as moving the cursor over an enabled hot spot in that movie) cause the callback routine to be executed. The reference constant specified by the `refCon` parameter is passed unchanged to your callback routine.

Special Considerations

Your mouse over hot spot procedure is called only for enabled hot spots.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrcursors
vrcursors.win
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

QTVRSetMouseOverTracking

Sets the state of mouse-over tracking.

```
OSErr QTVRSetMouseOverTracking (
   QTVRInstance qtvr,
   Boolean enable
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

enable

A Boolean value that indicates whether QuickTime VR should handle mouse-over tracking for the specified movie (TRUE) or not (FALSE).

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
By default, QuickTime VR tracks mouse movements in a QuickTime VR movie and changes the shape of the cursor as appropriate. If you disable mouse-over tracking (by passing FALSE in the `enable` parameter), you must call `QTVRMouseEnter` (page 63), `QTVRMouseWithin` (page 69), and `QTVRMouseLeave` (page 63) at the appropriate times to handle user actions.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRSetPanAngle

Sets the pan angle of a QuickTime VR movie.

```
OSErr QTVRSetPanAngle (
   QTVRInstance qtvr,
   float panAngle
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

panAngle

The desired pan angle of the specified movie. This value is constrained by the maximum and minimum pan angles of the movie. If the angle falls outside of those constraints and the control setting `kQTVRWrapPan` is disabled, the angle is set to the minimum or maximum, whichever is closer. If wrapping is enabled, the pan angle is clipped to fall within the constraints. Pan angle values are also clipped if the requested pan angle, when combined with the current tilt angle and field of view, would cause an image to lie outside the current constraints.

Return Value

See `Error Codes`. Returns `noErr` if there is no error. This function returns the result code `constraintReachedErr` if wrapping is off and the angle is set to the minimum or maximum constraint value.

Special Considerations

The pan and tilt angles are subject to the current pan and tilt range constraints, as imposed by the viewing limits and the current field of view. Accordingly, if you want to change the field of view, you should do so before adjusting the pan or tilt angles. Otherwise, the pan and tilt angles are clipped against the current field of view, which may result in an incorrect view when you alter the field of view.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrmovies

vrmovies.win

vrscript

vrscript.win

vrspeech

Declared In

`QuickTimeVR.h`

QTVRSetPrescreenImagingCompleteProc

Installs or removes a prescreen buffer imaging completion procedure.

```
OSErr QTVRSetPrescreenImagingCompleteProc (
   QTVRInstance qtvr,
   QTVRImagingCompleteUPP imagingCompleteProc,
   SInt32 refCon,
   UInt32 flags
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

imagingCompleteProc

A Universal Procedure Pointer for a `QTVRImagingCompleteProc` callback. To remove a previously installed `QTVRImagingCompleteProc` callback, set `imagingCompleteProc` to `NIL`.

refCon

A reference constant. This value is passed to the specified prescreen buffer imaging completion callback. Use this parameter to point to a data structure containing any information your callback needs.

flags

A constant (see below) that causes a draw attempt on every idle passed to the movie controller besides being called whenever QuickTime VR finishes drawing an image into the prescreen buffer. See these constants:

`kQTVRPreScreenEveryIdle`

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function installs the procedure specified by the `imagingCompleteProc` parameter as a prescreen buffer imaging completion procedure for the QuickTime VR movie specified by the `qtvr` parameter. Your procedure is called whenever QuickTime VR finishes drawing an image into the prescreen buffer. The reference constant specified by the `refCon` parameter is passed unchanged to that prescreen buffer imaging completion procedure.

Special Considerations

`QTVRSetPrescreenImagingCompleteProc` is valid only for panoramic nodes.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrscript

vrscript.win

Declared In

`QuickTimeVR.h`

QTVRSetTiltAngle

Sets the tilt angle of a QuickTime VR movie.

```
OSErr QTVRSetTiltAngle (
   QTVRInstance qtvr,
   float tiltAngle
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

tiltAngle

The desired tilt angle of the specified movie. This value is constrained by the maximum and minimum tilt angles of the movie. If the angle falls outside of those constraints and the control setting `kQTVRWrapTilt` is disabled, the angle is set to the minimum or maximum, whichever is closer. If wrapping is enabled, the tilt angle is clipped to fall within the constraints. Tilt angle values are also clipped if the requested tilt angle, when combined with the current pan angle and field of view, would cause an image to lie outside the current constraints.

Return Value

See `Error Codes`. Returns `noErr` if there is no error. This function returns the result code `constraintReachedErr` if wrapping is off and the angle is set to the minimum or maximum constraint value.

Discussion

When a cylindrical panorama is zoomed all the way out (to its maximum vertical field of view), it can no longer be tilted because its entire vertical surface is exposed. Attempting to set the tilt angle will result in a `ConstraintReachedErr` error (except for the degenerate case of setting the tilt angle to its current value).

The tilt angle may not be zero when the panorama is fully zoomed out; it may be tilted by one line of pixels. The tilt angle is small in this case, typically 0.006, but its exact magnitude depends on the height of the panorama; the taller the panorama, the smaller the error.

Do not test for a tilt angle of exactly 0.0 or attempt to adjust the tilt angle of a fully zoomed-out panorama.

Special Considerations

The pan and tilt angles are subject to the current pan and tilt range constraints, as imposed by the viewing limits and the current field of view. Accordingly, if you want to change the field of view, you should do so before adjusting the pan or tilt angles. Otherwise, the pan and tilt angles are clipped against the current field of view, which may result in an incorrect view when you alter the field of view.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrmovies

vrmovies.win

vrscript

vrscript.win

vrspeech

Declared In
QuickTimeVR.h

QTVRSetTransitionProperty

Sets the value of a transition property.

```
OSErr QTVRSetTransitionProperty (
   QTVRInstance qtvr,
   UInt32 transitionType,
   UInt32 transitionProperty,
   SInt32 transitionValue
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

transitionType

A type of transition (see below). See these constants:

kQTVRTransitionSwing

transitionProperty

A type of transition property (see below). See these constants:

kQTVRTransitionSpeed

kQTVRTransitionDirection

transitionValue

The desired value for the specified transition property.

Return Value
See Error Codes. Returns noErr if there is no error.

Discussion
This function sets the value of the transition property whose type is specified by the transitionType and transitionProperty parameters for the movie specified by the qtvr parameter to the value specified by the transitionValue parameter. Note that calling this function simply sets a transition property's value; you must still call QTVREnableTransition (page 26) to enable that transition effect.

Special Considerations

QTVRSetTransitionProperty is valid only for panoramic nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
QuickTimeVR.h

QTVRSetViewCenter

Sets the view center of a QuickTime VR movie.

```
OSErr QTVRSetViewCenter (
   QTVRInstance qtvr,
   const QTVRFloatPoint *viewCenter
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

viewCenter

A pointer to a QTVRFloatPoint structure that contains the desired view center of the specified movie. This point is constrained by the current field of view of the movie. The values you pass in the QTVRFloatPoint structure are adjusted so that the magnified area does not show anything outside the view.

Return Value
See Error Codes. Returns noErr if there is no error. If the kQTVRTranslation control setting is disabled, this function returns the result code constraintReachedErr and doesn't change the current view center. You can use QTVRSetControlSetting (page 80) to control the setting of kQTVRTranslation.

Special Considerations

QTVRSetViewCenter is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
QuickTimeVR.h

QTVRSetViewCurrentTime

Sets the time in the current QTVR view.

```
OSErr QTVRSetViewCurrentTime (
   QTVRInstance qtvr,
   TimeValue time
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

time

The desired time in the current view. This value should be greater than or equal to 0 and less than or equal to the value returned by `QTVRGetCurrentViewDuration` (page 37).

Return Value

See `Error Codes`. Returns `noErr` if there is no error. This function returns the result code `timeNotInViewErr` if the specified time value is greater than or equal to the view duration of the specified object node; in addition, it sets the current view time to 1 less than the view duration. Similarly, this function returns the result code `timeNotInViewErr` if the specified time value is less than 0; in that case, it sets the current view time to 0.

Special Considerations

`QTVRSetViewCurrentTime` is valid only for object nodes.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrscript

vrscript.win

Declared In

`QuickTimeVR.h`

QTVRSetViewParameter

Undocumented

```
OSErr QTVRSetViewParameter (
   QTVRInstance qtvr,
   UInt32 viewParameter,
   void *value,
   UInt32 flagsIn
);
```

Parameters

qtvr

Undocumented

viewParameter

Undocumented

value
Undocumented

flagsIn
Undocumented

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 5.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRSetViewRate

Sets the view rate of a QTVR object node.

```
OSErr QTVRSetViewRate (
   QTVRInstance qtvr,
   float rate
);
```

Parameters

qtvr
An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

rate
The desired view rate of the specified movie. A view rate is a floating-point value in the range from -100.0 to +100.0. Positive values indicate forward rates, and negative values indicate reverse rates. Set this parameter to 0 to stop the movie.

Return Value
See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function sets the view rate of the object node specified by the `qtvr` parameter to the rate specified by the rate parameter. A node's view rate might be modified by the current animation settings. If the value specified in the rate parameter lies outside the valid range, this function returns the result code `constraintReachedErr` and sets the view rate to the nearest constraint.

Special Considerations
`QTVRSetViewRate` is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
QuickTimeVR.h

QTVRSetViewState

Sets the value of a QTVR view state.

```
OSErr QTVRSetViewState (
   QTVRInstance qtvr,
   QTVRViewStateType viewStateType,
   UInt16 state
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling QTVRGetQTVRInstance (page 47).

viewStateType

A view state type (see below). See these constants:

kQTVRDefault
kQTVRCurrent
kQTVRMouseDown

state

The desired value of the specified type of view state.

Return Value
See Error Codes. Returns noErr if there is no error.

Special Considerations

QTVRSetViewState is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
QuickTimeVR.h

QTVRSetVisible

Sets a VR movie's visibility state.

```
OSErr QTVRSetVisible (
   QTVRInstance qtvr,
   Boolean visible
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

visible

A Boolean value that indicates whether the specified movie is to be visible (TRUE) or not (FALSE).

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

Setting the visibility state to FALSE is useful if you want to turn off imaging a QuickTime VR movie without purging the associated data from memory. When a panoramic node's visibility state is FALSE, the corrected image is still drawn to the prescreen buffer. You can access the data in that buffer by calling `QTVRSetPrescreenImagingCompleteProc` (page 90).

Special Considerations

This function is valid only for panoramic nodes.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrscript

vrscript.win

Declared In

`QuickTimeVR.h`

QTVRShowDefaultView

Displays the default view of a QTVR node.

```
OSErr QTVRShowDefaultView (
   QTVRInstance qtvr
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion
This function sets the default values of the pan angle, tilt angle, field of view, view center (for object nodes), default state, mouse-down state, and all applicable animation and control settings for the VR movie specified by the `qtvr` parameter. A VR movie's default view values are stored in the movie file.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
vrscript
vrscript.win

Declared In
`QuickTimeVR.h`

QTVRTiltToRow

Obtains the row number in the QTVR object image array that corresponds to a tilt angle.

```
short QTVRTiltToRow (
   QTVRInstance qtvr,
   float tiltAngle
);
```

Parameters

qtvr
An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

tiltAngle
A tilt angle.

Return Value
The zero-based row number in the current object image array that corresponds to the tilt angle specified by the `tiltAngle` parameter.

Special Considerations

This function is valid only for object nodes.

Version Notes
Introduced in QuickTime 3 or earlier.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRTriggerHotSpot

Triggers a QTVR hot spot.

```
OSErr QTVRTriggerHotSpot (
   QTVRInstance qtvr,
   UInt32 hotSpotID,
   QTAtomContainer nodeInfo,
   QTAtom selectedAtom
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

hotSpotID

A hot spot ID.

nodeInfo

A node information atom container, obtained from a previous call to `QTVRGetNodeInfo` (page 45). You can pass the `value` 0 in this parameter to have QuickTime determine the appropriate node information atom container.

selectedAtom

The atom of the hot spot to trigger. You can pass the `value` 0 in this parameter to have QuickTime determine the appropriate hot spot atom.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

One way you can use this function is to execute any hot spot without the user's having clicked it. Usually, you need only specify the `qtvr` instance and the hot spot ID. You can pass zero for the `nodeInfo` and `selectedAtom` parameters.

The more common use of this function is not in calling it directly, but in setting up an intercept procedure on it. This function is called internally by QuickTime whenever a user clicks a hot spot. You can intercept calls to trigger your custom hot spots, which allows you to perform any custom actions you desire.

When this function is called internally (and then intercepted by your intercept procedure), the `nodeInfo` and `selectedAtom` parameters have been properly set by QuickTime and are available for your use. For undefined hot spots that do not have an associated hot spot atom in the node info atom container, the `selectedAtom` parameter will be set to zero.

Special Considerations

You can call this function even on hot spots that are currently disabled.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrscript

vrscript.win

Declared In

`QuickTimeVR.h`

QTVRUpdate

Forces an immediate update of a QuickTime VR movie image.

```
OSErr QTVRUpdate (
   QTVRInstance qtvr,
   QTVRImagingMode imagingMode
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

imagingMode

An imaging mode. You can specify `kQTVRCurrentMode` (see below) to use the current imaging mode. See these constants:

`kQTVRCurrentMode`

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function immediately updates the image for the QuickTime VR movie specified by the `qtvr` parameter, without waiting for the next call to `MoviesTask` in your application's main event loop. If you plan to call this function repeatedly for a movie instance, then for improved performance you should bracket those calls with calls to `QTVRBeginUpdateStream` (page 21) and `QTVREndUpdateStream` (page 27).

Special Considerations

If you call this function after calling `QTVRBeginUpdateStream` (page 21) but before calling `QTVREndUpdateStream` (page 27), the `imagingMode` parameter passed to it must be the same as the `imagingMode` parameter passed to `QTVRBeginUpdateStream`. If you do not specify the same imaging mode to those two functions, an error will result.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrmovies

vrmovies.win

vrscript

vrscript.win

vrspeech

Declared In

`QuickTimeVR.h`

QTVRWrapAndConstrain

Preflights a change in the viewing or control characteristics of a QTVR object or panoramic node.

```
OSErr QTVRWrapAndConstrain (
   QTVRInstance qtvr,
   short kind,
   float value,
   float *result
);
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

kind

A constraint type (see below). See these constants:

`kQTVRPan`
`kQTVRTilt`
`kQTVRFieldOfView`
`kQTVRViewCenterH`
`kQTVRViewCenterV`

value

The desired value of the specified viewing characteristic.

result

On return, the value to which the specified viewing characteristic would be set if it were changed.

Return Value

See `Error Codes`. Returns `noErr` if there is no error.

Discussion

This function returns, in the `result` parameter, the constrained or wrapped value that would result from setting the viewing or control characteristic specified by the `kind` parameter to the value specified by the `value` parameter. For example, if the `kind` parameter is set to `kQTVRPan`, then this function returns the value that would result from calling `QTVRSetPanAngle` (page 89) with its `panAngle` parameter set to the `value` parameter. Similarly, you can use this function to find the current bounds of the view center. It takes into account the current constraints and wrapping modes of the node specified by the `qtvr` parameter. This function does not change the current view or other settings of the specified object or panorama.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

vrmovies
vrmovies.win
vrscript
vrscript.win

Declared In

`QuickTimeVR.h`

Callbacks

QTVRBackBufferImagingProc

An imaging procedure that draws directly into the back buffer for a QTVR panoramic node.

```
typedef OSErr (*QTVRBackBufferImagingProcPtr) (QTVRInstance qtvr, Rect *drawRect,
 UInt16 areaIndex, UInt32 flagsIn, UInt32 *flagsOut, SInt32 refCon);
```

If you name your function `MyQTVRBackBufferImagingProc`, you would declare it this way:

```
OSErr MyQTVRBackBufferImagingProc (
    QTVRInstance    qtvr,
    Rect            *drawRect,
    UInt16          areaIndex,
    UInt32          flagsIn,
    UInt32          *flagsOut,
    SInt32          refCon );
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

drawRect

Undocumented

areaIndex

Undocumented

flagsIn

Undocumented

flagsOut

Undocumented

refCon

A reference constant that the client code supplies to your callback. You can use this reference to point to a data structure containing any information your callback needs.

Return Value

See `Error Codes`. Your callback should return `noErr` if there is no error.

Declared In

`QuickTimeVR.h`

QTVREnteringNodeProc

A routine called whenever a QTVR node is entered, in response either to user actions or QuickTime VR Manager functions that change nodes.

```
typedef OSErr (*QTVREnteringNodeProcPtr) (QTVRInstance qtvr, UInt32 nodeID, SInt32
 refCon);
```

If you name your function `MyQTVREnteringNodeProc`, you would declare it this way:

```
OSErr MyQTVREnteringNodeProc (
    QTVRInstance    qtvr,
    UInt32          nodeID,
    SInt32          refCon );
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

nodeID

A node ID. Set this parameter to `kQTVRCurrentNode` to designate the current node.

refCon

A reference constant that the client code supplies to your callback. You can use this reference to point to a data structure containing any information your callback needs.

Return Value

See `Error Codes`. Your callback should return `noErr` if there is no error.

Declared In

`QuickTimeVR.h`

QTVRImagingCompleteProc

An imaging completion procedure for a QuickTime VR movie, called whenever QTVR finishes drawing an image into the prescreen buffer.

```
typedef OSErr (*QTVRImagingCompleteProcPtr) (QTVRInstance qtvr, SInt32 refCon);
```

If you name your function `MyQTVRImagingCompleteProc`, you would declare it this way:

```
OSErr MyQTVRImagingCompleteProc (
    QTVRInstance    qtvr,
    SInt32          refCon );
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

refCon

A reference constant that the client code supplies to your callback. You can use this reference to point to a data structure containing any information your callback needs.

Return Value

See `Error Codes`. Your callback should return `noErr` if there is no error.

Declared In

`QuickTimeVR.h`

QTVRInterceptProc

A routine that is called when certain QTVR functions are intercepted.

```
typedef void (*QTVRInterceptProcPtr) (QTVRInstance qtvr, QTVRInterceptPtr qtvrMsg,
 SInt32 refCon, Boolean *cancel);
```

If you name your function `MyQTVRInterceptProc`, you would declare it this way:

```
void MyQTVRInterceptProc (
    QTVRInstance        qtvr,
    QTVRInterceptPtr    qtvrMsg,
    SInt32              refCon,
    Boolean             *cancel );
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

qtvrMsg

A pointer to a `QTVRInterceptRecord` structure that determines which functions are intercepted.

refCon

A reference constant that the client code supplies to your callback. You can use this reference to point to a data structure containing any information your callback needs.

cancel

A pointer to a Boolean; set to TRUE if the intercept is cancelled.

Declared In

`QuickTimeVR.h`

QTVRLeavingNodeProc

A routine called whenever a QTVR node is left, in response either to user actions or QuickTime VR Manager functions that change nodes.

```
typedef OSErr (*QTVRLeavingNodeProcPtr) (QTVRInstance qtvr, UInt32 fromNodeID,
UInt32 toNodeID, Boolean *cancel, SInt32 refCon);
```

If you name your function `MyQTVRLeavingNodeProc`, you would declare it this way:

```
OSErr MyQTVRLeavingNodeProc (
    QTVRInstance    qtvr,
    UInt32          fromNodeID,
    UInt32          toNodeID,
    Boolean         *cancel,
    SInt32          refCon );
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

fromNodeID

The ID of the node being left. Set this parameter to `kQTVRCurrentNode` to designate the current node.

toNodeID

The ID of the node being entered. Set this parameter to `kQTVRCurrentNode` to designate the current node.

cancel

A pointer to a Boolean; set to TRUE if the callback is cancelled.

refCon

A reference constant that the client code supplies to your callback. You can use this reference to point to a data structure containing any information your callback needs.

Return Value

See `Error Codes`. Your callback should return `noErr` if there is no error.

Declared In

`QuickTimeVR.h`

QTVRMouseOverHotSpotProc

A routine that is called when the mouse is over a hot spot in a QTVR movie.

```
typedef OSErr (*QTVRMouseOverHotSpotProcPtr) (QTVRInstance qtvr, UInt32 hotSpotID,
 UInt32 flags, SInt32 refCon);
```

If you name your function `MyQTVRMouseOverHotSpotProc`, you would declare it this way:

```
OSErr MyQTVRMouseOverHotSpotProc (
    QTVRInstance    qtvr,
    UInt32          hotSpotID,
    UInt32          flags,
    SInt32          refCon );
```

Parameters

qtvr

An instance of a QuickTime VR movie. You can get this value by calling `QTVRGetQTVRInstance` (page 47).

hotSpotID

A hot spot ID.

flags

Undocumented

refCon

A reference constant that the client code supplies to your callback. You can use this reference to point to a data structure containing any information your callback needs.

Return Value

See `Error Codes`. Your callback should return `noErr` if there is no error.

Declared In

`QuickTimeVR.h`

Data Types

QTVRAngularUnits

Represents a type used by the Virtual Reality API.

```
typedef UInt32 QTVRAngularUnits;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRAreaOfInterest

Contains information passed to QTVRSetBackBufferImagingProc.

```
struct QTVRAreaOfInterest {
     float     panAngle;
     float     tiltAngle;
     float     width;
     float     height;
     UInt32    flags;
 };
```

Fields
`panAngle`

Discussion
The pan angle of the upper-left coordinate (in panorama space) of the area of interest.

`tiltAngle`

Discussion
The tilt angle of the upper-left coordinate (in panorama space) of the area of interest.

`width`

Discussion
The width of the area of interest.

`height`

Discussion
The height of the area of interest.

`flags`

Discussion
A set of bit flags (see below) that indicate when to call the back buffer imaging procedure for this area of interest. See these constants:

`kQTVRBackBufferEveryUpdate`

`kQTVRBackBufferEveryIdle`

`kQTVRBackBufferAlwaysRefresh`

Discussion
The `areasOfInterest` parameter to `QTVRSetBackBufferImagingProc` (page 77) specifies an array of `QTVRAreaOfInterest` structures, each one of which indicates a rectangular area in the QTVR back buffer.

Related Functions
`QTVRSetBackBufferImagingProc` (page 77)

Declared In
`QuickTimeVR.h`

QTVRBackBufferImagingUPP

Represents a type used by the Virtual Reality API.

```
typedef STACK_UPP_TYPE(QTVRBackBufferImagingProcPtr) QTVRBackBufferImagingUPP;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRControlSetting

Represents a type used by the Virtual Reality API.

```
typedef UInt32 QTVRControlSetting;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRCursorRecord

Contains information passed to QTVRReplaceCursor.

```
struct QTVRCursorRecord {
     UInt16    theType;
     SInt16    rsrcID;
     Handle    handle;
 };
```

Fields
`theType`

Discussion
A constant (see below) that defines the type of cursor to replace. See these constants:

`kQTVRUseDefaultCursor`

`kQTVRStdCursorType`

`kQTVRColorCursorType`

`rsrcID`

Discussion
The resource ID of the cursor to replace; see `QTVR Cursors`.

`handle`

Discussion
A handle to the cursor data that is to replace the specified cursor. If `theType` is `kQTVRUseDefaultCursor`, then this field should contain `NIL`.

Discussion
The `cursRecord` parameter to `QTVRReplaceCursor` (page 73) specifies a `QTVRCursorRecord` structure, which indicates the cursor to replace and its replacement cursor.

Version Notes
In earlier versions of QuickTime, `theType` was called the `type` field.

Related Functions
`QTVRReplaceCursor` (page 73)

Declared In
`QuickTimeVR.h`

QTVREnteringNodeUPP

Represents a type used by the Virtual Reality API.

```
typedef STACK_UPP_TYPE(QTVREnteringNodeProcPtr) QTVREnteringNodeUPP;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRFloatPoint

Specifies a point in a QTVR panorama or object.

```
struct QTVRFloatPoint {
     float    x;
     float    y;
 };
```

Fields
`x`

Discussion
The horizontal coordinate of the point.

`y`

Discussion
The vertical coordinate of the point.

Related Functions
QTVRAnglesToCoord (page 20)
QTVRCoordToAngles (page 23)
QTVRGetViewCenter (page 51)

Declared In
QuickTimeVR.h

QTVRImagingCompleteUPP

Represents a type used by the Virtual Reality API.

```
typedef STACK_UPP_TYPE(QTVRImagingCompleteProcPtr) QTVRImagingCompleteUPP;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeVR.h

QTVRImagingMode

Represents a type used by the Virtual Reality API.

```
typedef UInt32 QTVRImagingMode;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeVR.h

QTVRInstance

Represents a type used by the Virtual Reality API.

```
typedef struct OpaqueQTVRInstance * QTVRInstance;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeVR.h

QTVRInterceptRecord

Contains information about a QTVRInterceptProc being intercepted by QTVR.

```
struct QTVRInterceptRecord {
     SInt32    reserved1;
     SInt32    selector;
     SInt32    reserved2;
     SInt32    reserved3;
     SInt32    paramCount;
     void *    parameter[6];
 };
```

Fields

`reserved1`

Discussion
Reserved; do not use.

`selector`

Discussion
A constant that indicates which routine has triggered your intercept procedure (see below). See these constants:

- `kQTVRSetPanAngleSelector`
- `kQTVRSetTiltAngleSelector`
- `kQTVRSetFieldOfViewSelector`
- `kQTVRSetViewCenterSelector`
- `kQTVRMouseEnterSelector`
- `kQTVRMouseWithinSelector`
- `kQTVRMouseLeaveSelector`
- `kQTVRMouseDownSelector`
- `kQTVRMouseStillDownSelector`
- `kQTVRMouseUpSelector`
- `kQTVRTriggerHotSpotSelector`
- `kQTVRGetHotSpotTypeSelector`

`reserved2`

Discussion
Reserved; do not use.

`reserved3`

Discussion
Reserved; do not use.

`paramCount`

Discussion
The number of items in the `parameter` field.

`parameter`

Discussion
An array that holds, in order, the parameters that were passed to the intercepted function, minus the QTVR instance parameter. For example, if you intercept the `QTVRSetPanAngle` function, the `parameter` field contains a single item, a pointer to a floating-point value that is the new pan angle. You can determine how many items the `parameter` field contains by inspecting the `paramCount` field.

Related Functions

`QTVRCallInterceptedProc` (page 22)

Declared In
QuickTimeVR.h

QTVRInterceptUPP

Represents a type used by the Virtual Reality API.

```
typedef STACK_UPP_TYPE(QTVRInterceptProcPtr) QTVRInterceptUPP;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeVR.h

QTVRLeavingNodeUPP

Represents a type used by the Virtual Reality API.

```
typedef STACK_UPP_TYPE(QTVRLeavingNodeProcPtr) QTVRLeavingNodeUPP;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeVR.h

QTVRMouseOverHotSpotUPP

Represents a type used by the Virtual Reality API.

```
typedef STACK_UPP_TYPE(QTVRMouseOverHotSpotProcPtr) QTVRMouseOverHotSpotUPP;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeVR.h

QTVRNudgeControl

Represents a type used by the Virtual Reality API.

```
typedef UInt32 QTVRNudgeControl;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
QuickTimeVR.h

QTVRObjectAnimationSetting

Represents a type used by the Virtual Reality API.

```
typedef UInt32 QTVRObjectAnimationSetting;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRProcSelector

Represents a type used by the Virtual Reality API.

```
typedef UInt32 QTVRProcSelector;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

QTVRViewStateType

Represents a type used by the Virtual Reality API.

```
typedef UInt32 QTVRViewStateType;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`QuickTimeVR.h`

Constants

kQTVRBackBufferAlwaysRefresh

Constants grouped with kQTVRBackBufferAlwaysRefresh.

```
enum {
  kQTVRBackBufferEveryUpdate    = 1L << 0,
  kQTVRBackBufferEveryIdle      = 1L << 1,
  kQTVRBackBufferAlwaysRefresh  = 1L << 2,
  kQTVRBackBufferHorizontal     = 1L << 3 /* Requires that backbuffer proc be
long-rowBytes aware (gestaltQDHasLongRowBytes)*/
};
```

Constants

`kQTVRBackBufferEveryUpdate`

If this bit is set, the back buffer imaging procedure is to be called whenever QuickTime is about to update the window containing the specified QuickTime VR movie instance. That is, the procedure is called just before QuickTime unwarps the back buffer image into the prescreen buffer and redraws the screen image.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

`kQTVRBackBufferEveryIdle`

If this bit is set, the back buffer imaging procedure is to be called when either `MCIsPlayerEvent` is called with the idle parameter, or when `MCIdle` is called. Its purpose is to cause the software to draw as often as possible.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

`kQTVRBackBufferAlwaysRefresh`

If this bit is set, the back buffer is always refreshed to the proper movie data just before your back buffer imaging procedure is called. If your back buffer imaging procedure completely overwrites the rectangle passed to it, you should not set this bit.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

Declared In

`QuickTimeVR.h`

QTVRGoToNodeID Values

Constants passed to QTVRGoToNodeID.

```
enum {
  kQTVRCurrentNode              = 0,
  kQTVRPreviousNode             = (long)0x80000000,
  kQTVRDefaultNode              = (long)0x80000001
};
```

Declared In

`QuickTimeVR.h`

QTVRSetViewState Values

Constants passed to QTVRSetViewState.

```
enum {
  kQTVRDefault                  = 0,
  kQTVRCurrent                  = 2,
  kQTVRMouseDown                = 3
};
```

Declared In
QuickTimeVR.h

QTVRSetBackBufferPrefs Values

Constants passed to QTVRSetBackBufferPrefs.

```
enum {
  kQTVRDefaultRes               = 0,
  kQTVRFullRes                  = 1L << 0,
  kQTVRHalfRes                  = 1L << 1,
  kQTVRQuarterRes               = 1L << 2
};
enum {
  kQTVRMinimumCache             = -1,
  kQTVRSuggestedCache           = 0,
  kQTVRFullCache                = 1
};
```

Constants

`kQTVRQuarterRes`

One-quarter the full resolution of the image.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

`kQTVRMinimumCache`

The minimum cache size required to display the specified VR movie.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

`kQTVRSuggestedCache`

The suggested cache size, a cache large enough to allow full zooming out of the panorama.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

Declared In
QuickTimeVR.h

QTVRSetAngularUnits Values

Constants passed to QTVRSetAngularUnits.

```
enum {
  kQTVRDegrees                  = 0,
  kQTVRRadians                  = 1
};
```

Declared In
QuickTimeVR.h

QTVREnableHotSpot Values

Constants passed to QTVREnableHotSpot.

```
enum {
  kQTVRHotSpotID                = 0,
  kQTVRHotSpotType              = 1,
  kQTVRAllHotSpots              = 2
};
```

Declared In
QuickTimeVR.h

kQTVRImagingCorrection

Constants grouped with kQTVRImagingCorrection.

```
enum {
  kQTVRImagingCorrection        = 1,
  kQTVRImagingQuality           = 2,
  kQTVRImagingDirectDraw        = 3,
  kQTVRImagingCurrentMode       = 100   /* Get Only*/
};
```

Declared In
QuickTimeVR.h

QTVRSetInteractionProperty Values

Constants passed to QTVRSetInteractionProperty.

```
enum {
  kQTVRInteractionMouseClickHysteresis = 1, /* pixels within which the mouse is
considered not to have moved (UInt16)*/
  kQTVRInteractionMouseClickTimeout = 2, /* ticks after which a mouse click times
 out and turns into panning (UInt32)*/
  kQTVRInteractionPanTiltSpeed  = 3,    /* control the relative pan/tilt speed from
 1 (slowest) to 10 (fastest). (UInt32) Default is 5;*/
  kQTVRInteractionZoomSpeed     = 4,    /* control the relative zooming speed from
 1 (slowest) to 10 (fastest). (UInt32) Default is 5;*/
  kQTVRInteractionTranslateOnMouseDown = 101, /* Holding MouseDown with this setting
 translates zoomed object movies (Boolean)*/
  kQTVRInteractionMouseMotionScale = 102, /* The maximum angle of rotation caused
 by dragging across the display window. (* float)*/
  kQTVRInteractionNudgeMode     = 103   /* A QTVRNudgeMode: rotate, translate, or
 the same as the current mouse mode. Requires QTVR 2.1*/
};
```

Declared In
QuickTimeVR.h

kQTVRDontLoopViewFrames

Constants grouped with kQTVRDontLoopViewFrames.

```
enum {
                                        /* View Frame Animation Settings*/
  kQTVRPalindromeViewFrames     = 1,
  kQTVRStartFirstViewFrame      = 2,
  kQTVRDontLoopViewFrames       = 3,
  kQTVRPlayEveryViewFrame      = 4,    /* Requires QTVR 2.1 (kQTVRAPIMajorVersion02
 + kQTVRAPIMinorVersion10)*/
                                        /* View Animation Settings*/
  kQTVRSyncViewToFrameRate      = 16,
  kQTVRPalindromeViews          = 17,
  kQTVRPlayStreamingViews      = 18    /* Requires QTVR 2.1 (kQTVRAPIMajorVersion02
 + kQTVRAPIMinorVersion10)*/
};
```

Declared In
QuickTimeVR.h

QTVRWrapAndConstrain Values

Constants passed to QTVRWrapAndConstrain.

```
enum {
  kQTVRPan                      = 0,
  kQTVRTilt                     = 1,
  kQTVRFieldOfView              = 2,
  kQTVRViewCenterH              = 4,    /* WrapAndConstrain only*/
  kQTVRViewCenterV              = 5     /* WrapAndConstrain only*/
};
```

Declared In
QuickTimeVR.h

QTVRSetPrescreenImagingCompleteProc Values

Constants passed to QTVRSetPrescreenImagingCompleteProc.

```
enum {
  kQTVRPreScreenEveryIdle       = 1L << 0 /* Requires QTVR 2.1
(kQTVRAPIMajorVersion02 + kQTVRAPIMinorVersion10)*/
};
```

Declared In
`QuickTimeVR.h`

kQTVRDown

Constants grouped with kQTVRDown.

```
enum {
  kQTVRRight                    = 0,
  kQTVRUpRight                  = 45,
  kQTVRUp                       = 90,
  kQTVRUpLeft                   = 135,
  kQTVRLeft                     = 180,
  kQTVRDownLeft                 = 225,
  kQTVRDown                     = 270,
  kQTVRDownRight                = 315
};
```

Constants

`kQTVRLeft`

Swing the view to the left.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

`kQTVRDown`

Swing the view down.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

Declared In
`QuickTimeVR.h`

kQTVRGetHotSpotTypeSelector

Constants grouped with kQTVRGetHotSpotTypeSelector.

```
enum {
  kQTVRSetPanAngleSelector      = 0x2000,
  kQTVRSetTiltAngleSelector     = 0x2001,
  kQTVRSetFieldOfViewSelector   = 0x2002,
  kQTVRSetViewCenterSelector    = 0x2003,
  kQTVRMouseEnterSelector       = 0x2004,
  kQTVRMouseWithinSelector      = 0x2005,
  kQTVRMouseLeaveSelector       = 0x2006,
  kQTVRMouseDownSelector        = 0x2007,
  kQTVRMouseStillDownSelector   = 0x2008,
  kQTVRMouseUpSelector          = 0x2009,
  kQTVRTriggerHotSpotSelector   = 0x200A,
  kQTVRGetHotSpotTypeSelector   = 0x200B, /* Requires QTVR 2.1
(kQTVRAPIMajorVersion02 + kQTVRAPIMinorVersion10)*/
  kQTVRSetViewParameterSelector = 0x200C, /* Requires QTVR 5.0
(kQTVRAPIMajorVersion05 + kQTVRAPIMinorVersion00)*/
  kQTVRGetViewParameterSelector = 0x200D /* Requires QTVR 5.0 (kQTVRAPIMajorVersion05
 + kQTVRAPIMinorVersion00)*/
};
```

Constants

`kQTVRSetPanAngleSelector`

Value is 0x2000.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

`kQTVRSetTiltAngleSelector`

Value is 0x2000.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

`kQTVRSetFieldOfViewSelector`

Value is 0x2002.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

`kQTVRSetViewCenterSelector`

Value is 0x2003.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

`kQTVRMouseEnterSelector`

Value is 0x2004.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

`kQTVRMouseWithinSelector`

Value is 0x2005.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

`kQTVRMouseLeaveSelector`
Value is 0x2006.
Available in Mac OS X v10.0 and later.
Declared in `QuickTimeVR.h`.

`kQTVRMouseDownSelector`
Value is 0x2007.
Available in Mac OS X v10.0 and later.
Declared in `QuickTimeVR.h`.

`kQTVRMouseStillDownSelector`
Value is 0x2008.
Available in Mac OS X v10.0 and later.
Declared in `QuickTimeVR.h`.

`kQTVRMouseUpSelector`
Value is 0x2009.
Available in Mac OS X v10.0 and later.
Declared in `QuickTimeVR.h`.

`kQTVRTriggerHotSpotSelector`
Value is 0x200A.
Available in Mac OS X v10.0 and later.
Declared in `QuickTimeVR.h`.

`kQTVRGetHotSpotTypeSelector`
Value is 0x200B.
Available in Mac OS X v10.0 and later.
Declared in `QuickTimeVR.h`.

Declared In
`QuickTimeVR.h`

kQTVRAllModes

Constants grouped with kQTVRAllModes.

```
enum {
  kQTVRStatic                   = 1,
  kQTVRMotion                   = 2,
  kQTVRCurrentMode              = 0,    /* Special Value for QTVRUpdate*/
  kQTVRAllModes                 = 100   /* Special value for QTVRSetProperty*/
};
```

Declared In
`QuickTimeVR.h`

QTVRSetTransitionProperty Values

Constants passed to QTVRSetTransitionProperty.

```
enum {
  kQTVRTransitionSpeed          = 1,
  kQTVRTransitionDirection      = 2
};
enum {
  kQTVRTransitionSwing          = 1
};
```

Declared In
`QuickTimeVR.h`

QTVRCursorRecord Values

Constants passed to QTVRCursorRecord.

```
enum {
  kQTVRUseDefaultCursor         = 0,
  kQTVRStdCursorType            = 1,
  kQTVRColorCursorType          = 2
};
```

Constants

`kQTVRUseDefaultCursor`
Restore the default cursor. In this case, the `handle` field of the cursor record should contain `NIL`.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

`kQTVRStdCursorType`
The cursor is a standard black-and-white cursor.

Available in Mac OS X v10.0 and later.

Declared in `QuickTimeVR.h`.

Declared In
`QuickTimeVR.h`

kQTVRCube

Constants grouped with kQTVRCube.

```
enum {
  kQTVRUseMovieGeometry         = 0,
  kQTVRVerticalCylinder         = 'vcyl',
  kQTVRHorizontalCylinder       = 'hcyl',
  kQTVRCube                     = 'cube'
};
```

Declared In
`QuickTimeVR.h`

QTVRSetControlSetting Values

Constants passed to QTVRSetControlSetting.

```
enum {
  kQTVRWrapPan                  = 1,
  kQTVRWrapTilt                 = 2,
  kQTVRCanZoom                  = 3,
  kQTVRReverseHControl          = 4,
  kQTVRReverseVControl          = 5,
  kQTVRSwapHVControl            = 6,
  kQTVRTranslation              = 7
};
```

Declared In

QuickTimeVR.h

Document Revision History

This table describes the changes to *QuickTime Virtual Reality Reference*.

Date	Notes
2006-11-10	Clarify panorama tilt angle restraints.
2006-05-23	New document, based on previously published material, that describes the API for QuickTime Virtual Reality.

Document Revision History

Index

D

K

N

Q