Import and Export Reference for QuickTime

QuickTime > Import & Export

2006-05-23

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

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

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Import and Export Reference for QuickTime

Framework:	Frameworks/QuickTime.framework
Declared in	Dialogs.h ImageCompression.h

Overview

Image importers and exporters manage the import and export of graphic images, such as JPEG, TIFF, Photoshop, and PNG. Movie data exchange components support the import and export of other multimedia formats, such as AIFF, WAVE, AVI, MPEG-1, MIDI, MPEG-4, 3GPP, MP3, MPEG-2, H.263, and OpenDML. QuickTime can open any format file for which it has an importer and create any for which it has an exporter.

Functions by Task

Accessing a Graphics Exporter's Input Image

`GraphicsExportDrawInputImage` (page 22)
Draws a rectangular portion of the input image in a graphics export operation.

`GraphicsExportGetInputImageDepth` (page 33)
Returns the depth of the input image for a graphics export operation.

`GraphicsExportGetInputImageDescription` (page 33)
Returns an image description describing the input image in a graphics export operation.

`GraphicsExportGetInputImageDimensions` (page 34)
Returns the dimensions of the input image in a graphics export operation.

Accessing Graphics Exporter Settings

`GraphicsExportGetColorSyncProfile` (page 23)
Gets the current value of the ColorSync profile for a graphics export operation.

`GraphicsExportGetCompressionMethod` (page 24)
Returns the compression method for a graphics export operation.

`GraphicsExportGetCompressionQuality` (page 24)
Returns the compression quality value for a graphics export operation.

`GraphicsExportGetDepth` (page 26)
Returns the current depth setting for a graphics export operation.

GraphicsExportGetDontRecompress (page 27)
Determines whether the original compressed data for a graphics export operation will not be decompressed and recompressed, but be copied through to the output file.

GraphicsExportGetInterlaceStyle (page 38)
Returns the interlace style in a graphics export operation.

GraphicsExportGetMetaData (page 38)
Returns the current user data setting in a graphics export operation.

GraphicsExportGetResolution (page 44)
Determines the resolution of a graphics exporter component.

GraphicsExportGetTargetDataSize (page 45)
Returns the current desired maximum data size for a graphics export operation.

GraphicsExportSetColorSyncProfile (page 50)
Sets the ColorSync profile to embed in the image file for a graphics export operation.

GraphicsExportSetCompressionMethod (page 50)
Defines the compression method to use in a graphics export operation.

GraphicsExportSetCompressionQuality (page 51)
Defines the compression quality for a graphics export operation.

GraphicsExportSetDepth (page 52)
Defines the depth to use in a graphics export operation.

GraphicsExportSetDontRecompress (page 52)
Requests that the original compressed data for a graphics export operation not be decompressed and recompressed, but be copied through to the output file.

GraphicsExportSetInterlaceStyle (page 62)
Defines the interlace style for a graphics export operation.

GraphicsExportSetMetaData (page 62)
Defines supplemental data for a graphics export operation, such as copyright text.

GraphicsExportSetResolution (page 67)
Defines the resolution to store in the image file for a graphics export operation.

GraphicsExportSetTargetDataSize (page 69)
Defines a desired maximum data size for a graphics export operation and asks for a quality that does not exceed that size.

Drawing Imported Images

GraphicsImportDraw (page 74)
Draws an imported image.

GraphicsImportGetGWorld (page 94)
Returns the current graphics port and device for drawing an imported image.

GraphicsImportSetGWorld (page 116)
Sets the graphics port and device for drawing an imported image.

Finding Out About Graphics Export Image Formats

GraphicsExportGetDefaultFileNameExtension (page 25)
Returns the suggested file name extension for a graphics export operation.

GraphicsExportGetDefaultFileTypeAndCreator (page 26)
Returns the suggested file type and creator for a graphics export operation.

GraphicsExportGetMIMETypeList (page 39)
Returns MIME types and other information about the graphics format in a graphics export operation.

Getting and Setting Progress Procs

GraphicsExportGetProgressProc (page 43)
Returns the current progress function for a graphics export operation.

GraphicsExportSetProgressProc (page 67)
Installs a progress function in a graphics export operation.

Getting Image Characteristics

GraphicsImportDoesDrawAllPixels (page 71)
Asks whether the graphics importer expects to draw every pixel.

GraphicsImportGetImageDescription (page 96)
Returns image description information for an imported image.

GraphicsImportGetMetaData (page 98)
Extracts user data from an imported image file.

GraphicsImportGetNaturalBounds (page 99)
Returns the bounding rectangle of an imported image.

GraphicsImportValidate (page 122)
Validates image data for a data reference to an imported image.

Getting MIME Types

GraphicsImportGetMIMETypeList (page 99)
Returns a list of MIME types supported by the graphics importer component.

Internal Graphics Export Routines

GraphicsExportCanTranscode (page 18)
Asks whether the current graphics export operation should be performed by transcoding.

GraphicsExportCanUseCompressor (page 19)
Asks whether to use a compressor in a graphics export operation.

GraphicsExportDoStandaloneExport (page 20)
Performs a standalone graphics export operation.

`GraphicsExportDoTranscode` (page 21)
: Performs a graphics export operation by transcoding.

`GraphicsExportDoUseCompressor` (page 21)
: Performs a graphics export operation with compression.

Managing Graphics Importers

`GraphicsImportGetColorSyncProfile` (page 81)
: Returns a ColorSync profile for an imported image, if one is embedded in the image file.

`GraphicsImportGetDataOffsetAndSize` (page 83)
: Returns the offset and size of the compressed image data within an imported image file.

`GraphicsImportGetDataOffsetAndSize64` (page 84)
: Provides a 64-bit version of GraphicsImportGetDataOffsetAndSize.

`GraphicsImportGetDataReferenceOffsetAndLimit` (page 85)
: Returns the data reference starting offset and data size limit for an imported image.

`GraphicsImportGetDataReferenceOffsetAndLimit64` (page 86)
: Provides a 64-bit version of GraphicsImportGetDataReferenceOffsetAndLimit.

`GraphicsImportGetDefaultClip` (page 87)
: Returns the default clipping region for an imported image, if one is stored there.

`GraphicsImportGetDefaultGraphicsMode` (page 88)
: Returns the default graphics mode for an imported image, if one is stored there.

`GraphicsImportGetDefaultMatrix` (page 89)
: Returns the default matrix for an imported image, if one is stored there.

`GraphicsImportGetDefaultSourceRect` (page 89)
: Returns the default source rectangle for an imported image, if one is stored there.

`GraphicsImportGetDestRect` (page 90)
: Returns the destination rectangle for an imported image.

`GraphicsImportGetFlags` (page 93)
: Returns the current flags of a graphics importer component.

`GraphicsImportGetImageCount` (page 95)
: Returns the number of images in an imported image file.

`GraphicsImportGetImageIndex` (page 97)
: Returns the current image index for an imported image.

`GraphicsImportReadData` (page 103)
: Reads imported image data.

`GraphicsImportReadData64` (page 104)
: Provides a 64-bit version of GraphicsImportReadData.

`GraphicsImportSetDataReferenceOffsetAndLimit` (page 112)
: Specifies the data reference starting offset and data size limit for an imported image.

`GraphicsImportSetDataReferenceOffsetAndLimit64` (page 113)
: Provides a 64-bit version of GraphicsImportSetDataReferenceOffsetAndLimit.

`GraphicsImportSetDestRect` (page 114)
: Sets the destination rectangle for a graphics import operation.

GraphicsImportSetFlags (page 115)
Sets the flags for a graphics importer component.

GraphicsImportSetImageIndex (page 117)
Specifies the image index for an imported image.

GraphicsImportSetImageIndexToThumbnail (page 118)
Looks for a graphics subimage that contains a thumbnail.

Obtaining Graphics Exporter Settings

GraphicsExportGetSettingsAsAtomContainer (page 44)
Retrieves the current settings from a graphics exporter component.

GraphicsExportGetSettingsAsText (page 45)
Retrieves the current settings from the graphics export component in a user-readable format.

GraphicsExportRequestSettings (page 49)
Displays a dialog for the user to configure graphics exporter settings, if applicable.

GraphicsExportSetSettingsFromAtomContainer (page 68)
Sets the graphics exporter component's current configuration to match the settings in a passed atom container.

Reading Graphics Exporter Input Data

GraphicsExportGetInputDataSize (page 30)
Returns the number of bytes of original image data that can be read in a graphics export operation.

GraphicsExportMayExporterReadInputData (page 46)
Asks whether the image source for a graphics export operation is in a form that the exporter can read.

GraphicsExportReadInputData (page 47)
Reads the original image data in a graphics export operation.

Restricting the Range of an Input Image's Source

GraphicsExportGetInputOffsetAndLimit (page 35)
Retrieves the current input offset and limit in a graphics export operation.

GraphicsExportSetInputOffsetAndLimit (page 59)
Specifies the portion of an input data reference, file, handle or pointer that a graphics exporter is permitted to read.

Saving Image Files

GraphicsImportDoExportImageFileDialog (page 72)
Presents a dialog box letting the user save an imported image in a foreign file format.

GraphicsImportExportImageFile (page 75)
Saves an imported image in a foreign file format.

`GraphicsImportGetAsPicture` (page 78)
: Creates a QuickDraw picture handle to an imported image.

`GraphicsImportGetExportImageTypeList` (page 91)
: Returns information about available export formats for a graphics importer.

`GraphicsImportGetExportSettingsAsAtomContainer` (page 92)
: Retrieves settings for image files exported by the graphics importer.

`GraphicsImportSaveAsPicture` (page 104)
: Creates a QuickDraw picture file for an imported image.

`GraphicsImportSaveAsQuickTimeImageFile` (page 106)
: Creates a QuickTime Image file of an imported image.

`GraphicsImportSetExportSettingsFromAtomContainer` (page 114)
: Determines settings for the export of imported image files.

Setting Drawing Parameters

`GraphicsImportGetBoundsRect` (page 79)
: Returns the bounding rectangle for drawing an imported image.

`GraphicsImportGetClip` (page 80)
: Returns the current clipping region for an imported image.

`GraphicsImportGetGraphicsMode` (page 94)
: Returns the graphics transfer mode for an imported image.

`GraphicsImportGetMatrix` (page 97)
: Returns the transformation matrix to be used for drawing an imported image.

`GraphicsImportGetProgressProc` (page 101)
: Returns the current progress function for a graphics import operation.

`GraphicsImportGetQuality` (page 101)
: Returns the image quality value for an imported image.

`GraphicsImportGetSourceRect` (page 102)
: Returns the current source rectangle for an imported image.

`GraphicsImportSetBoundsRect` (page 107)
: Defines the rectangle in which to draw an imported image.

`GraphicsImportSetClip` (page 108)
: Defines the clipping region for drawing an imported image.

`GraphicsImportSetGraphicsMode` (page 116)
: Sets the graphics transfer mode for an imported image.

`GraphicsImportSetMatrix` (page 119)
: Defines the transformation matrix to use for drawing an imported image.

`GraphicsImportSetProgressProc` (page 120)
: Installs a progress procedure to call while drawing an imported image.

`GraphicsImportSetQuality` (page 121)
: Sets the image quality value for an imported image.

`GraphicsImportSetSourceRect` (page 122)
: Sets the source rectangle to use for an imported image.

Specifying a Graphics Import Data Source

`GraphicsImportGetDataFile` (page 81)
Returns the file containing the graphics data for an imported image.

`GraphicsImportGetDataHandle` (page 82)
Returns a handle to imported graphics data.

`GraphicsImportGetDataReference` (page 84)
Returns a data reference to imported graphics data.

`GraphicsImportSetDataFile` (page 109)
Specifies the file that contains imported graphics data.

`GraphicsImportSetDataHandle` (page 110)
Specifies the handle that references imported graphics data.

`GraphicsImportSetDataReference` (page 111)
Specifies the data reference for imported graphics data.

Specifying Destinations for Output Images

`GraphicsExportGetOutputDataReference` (page 39)
Gets the output data reference handle in a graphics export operation.

`GraphicsExportGetOutputFile` (page 40)
Returns the current output file for a graphics export operation.

`GraphicsExportGetOutputFileTypeAndCreator` (page 41)
Gets the type and creator codes for the output file in a graphics export operation.

`GraphicsExportGetOutputHandle` (page 41)
Returns the current output handle for a graphics export operation.

`GraphicsExportGetOutputOffsetAndMaxSize` (page 42)
Returns the output starting offset and maximum size limit for a graphics export operation.

`GraphicsExportSetOutputDataReference` (page 63)
Returns the current output data reference for a graphics export operation.

`GraphicsExportSetOutputFile` (page 64)
Defines the output file for a graphics export operation.

`GraphicsExportSetOutputFileTypeAndCreator` (page 64)
Sets the file type and creator codes for the output file of a graphics export operation.

`GraphicsExportSetOutputHandle` (page 65)
Sets a handle to the output of a graphics export operation.

`GraphicsExportSetOutputOffsetAndMaxSize` (page 66)
Specifies the output starting offset and maximum size limit for a graphics export operation.

Specifying Sources for Graphics Exporter Input Images

`GraphicsExportGetInputDataReference` (page 29)
Returns the current input data reference for a graphics export operation.

GraphicsExportGetInputFile (page 30)
: Returns the current input file for a graphics export operation.

GraphicsExportGetInputGraphicsImporter (page 31)
: Returns the current input graphics importer instance for a graphics export operation.

GraphicsExportGetInputGWorld (page 32)
: Returns the current input graphics world for a graphics export operation.

GraphicsExportGetInputHandle (page 32)
: Returns the current input handle for a graphics export operation.

GraphicsExportGetInputPicture (page 35)
: Returns the current input picture in a graphics export operation.

GraphicsExportGetInputPixmap (page 36)
: Returns the current input pixmap in a graphics export operation.

GraphicsExportGetInputPtr (page 37)
: Returns the current input pointer in a graphics export operation.

GraphicsExportSetInputDataReference (page 55)
: Specifies that the source image for a graphics export operation is a compressed image stored in a data reference.

GraphicsExportSetInputFile (page 55)
: Specifies that the source image for a graphics export operation is a compressed image stored in a file.

GraphicsExportSetInputGraphicsImporter (page 56)
: Specifies that the source image for a graphics export operation is to be drawn by a graphics importer instance.

GraphicsExportSetInputGWorld (page 57)
: Specifies that the source image for a graphics export operation is a graphics world.

GraphicsExportSetInputHandle (page 58)
: Specifies that the source image for a graphics export operation is a compressed image referenced by a handle.

GraphicsExportSetInputPicture (page 59)
: Specifies that the source image for a graphics export operation is a picture.

GraphicsExportSetInputPixmap (page 60)
: Specifies that the source image for a graphics export operation is a pixmap.

GraphicsExportSetInputPtr (page 61)
: Specifies that the source image for a graphics export operation is a compressed image stored at a fixed address in memory.

Working With Exif Files

GraphicsExportGetExifEnabled (page 27)
: Returns the graphics exporter's current Exif export setting.

GraphicsExportGetThumbnailEnabled (page 46)
: Returns the current thumbnail creation setting for the graphics exporter when exporting Exif files.

GraphicsExportSetExifEnabled (page 53)
: Determines whether or not the graphics exporter component should create Exif files.

GraphicsExportSetThumbnailEnabled (page 69)
Determines whether or not the graphics exporter component should create an embedded thumbnail inside an exported Exif file.

Writing Graphics Exporter Output Data

GraphicsExportGetOutputMark (page 42)
Returns the current file position for a graphics export operation.

GraphicsExportReadOutputData (page 48)
Reads output image data in a graphics export operation.

GraphicsExportSetOutputMark (page 65)
Seeks to the specified file position in a graphics export operation.

GraphicsExportWriteOutputData (page 70)
Writes output image data in a graphics export operation.

Supporting Functions

GraphicsExportDoExport (page 19)
Performs a graphics export operation.

GraphicsExportGetInputCGBitmapContext (page 28)
Retrieves the CGBitmapContext that the graphics exporter is using as its input image.

GraphicsExportGetInputCGImage (page 28)
Determines which Core Graphics CGImage is the source for a graphics export operation.

GraphicsExportSetInputCGBitmapContext (page 54)
Sets the CGBitmapContext that the graphics exporter will use as its input image.

GraphicsExportSetInputCGImage (page 54)
Specifies a Core Graphics CGImage as the source for a graphics export operation.

GraphicsImportCreateCGImage (page 71)
Imports an image as a Core Graphics CGImage.

GraphicsImportDoExportImageFileToDataRefDialog (page 73)
Presents a dialog box that lets the user save an imported image in a foreign file format.

GraphicsImportExportImageFileToDataRef (page 77)
Saves an imported image in a foreign file format.

GraphicsImportGetAliasedDataReference (page 77)
Deprecated.

GraphicsImportGetBaseDataOffsetAndSize64 (page 79)
Undocumented

GraphicsImportGetDestinationColorSyncProfileRef (page 90)
Retrieves a ColorSync profile from a graphics importer component.

GraphicsImportGetGenericColorSyncProfile (page 93)
Retrieves the generic colorsync profile for a graphics importer component.

GraphicsImportGetOverrideSourceColorSyncProfileRef (page 100)
Retrieves the override ColorSync profile for a graphics importer component.

`GraphicsImportSaveAsPictureToDataRef` (page 105)
Creates a storage location that contains a QuickDraw picture for an imported image.

`GraphicsImportSaveAsQuickTimeImageFileToDataRef` (page 107)
Creates a storage location that contains a QuickTime image of an imported image.

`GraphicsImportSetDestinationColorSyncProfileRef` (page 113)
Sets the ColorSync profile for a graphics importer component.

`GraphicsImportSetOverrideSourceColorSyncProfileRef` (page 119)
Sets the override ColorSync profile for a graphics importer component.

`GraphicsImportWillUseColorMatching` (page 123)
Asks whether GraphicsImportDraw will use color matching if called with the current importer settings.

Functions

GraphicsExportCanTranscode

Asks whether the current graphics export operation should be performed by transcoding.

```
ComponentResult GraphicsExportCanTranscode (
   GraphicsExportComponent ci,
   Boolean *canTranscode
);
```

Parameters

ci
The component instance that identifies your connection to the graphics exporter component.

canTranscode
Points to a Boolean to receive the answer. TRUE means that the current export operation should be performed by transcoding, FALSE that it should not.

Return Value

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

Discussion

Graphics exporters may be able to transcode from some inputs and not from others. For instance, the JPEG graphics exporter is able to transcode compressed JPEG streams, but not other kinds of compressed data. The base graphics exporter makes this call to the format-specific graphics exporter to ask whether the current export operation should be done by transcoding. If the format-specific exporter replies that it should, the base exporter calls `GraphicsExportDoTranscode` (page 21) to do so. If the answer is no, then the format-specific exporter will not be able to transcode.

Special Considerations

This function is used for internal communication between the base and format-specific graphics exporter. Applications will not usually need to call it. Format-specific exporters may delegate this call, in which case the base graphics exporter's implementation gives a reply of FALSE.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportCanUseCompressor

Asks whether to use a compressor in a graphics export operation.

```
ComponentResult GraphicsExportCanUseCompressor (
   GraphicsExportComponent ci,
   Boolean *canUseCompressor,
   void *codecSettingsAtomContainerPtr
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

canUseCompressor

A Boolean variable to receive the answer.

codecSettingsAtomContainerPtr

A pointer to a `QTAtomContainer` variable. If the `canUseCompressor` parameter returns TRUE, the format-specific exporter should create a new QuickTime atom container with information about the compression operation and return it here.

Return Value

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

Discussion

The base graphics exporter makes this call of the format-specific graphics exporter to ask whether the current export operation should be done by using an image compressor. If the answer is TRUE, the format-specific exporter must also create and return an atom container. This atom container must contain a big-endian `'vide'` atom with at least a child atom of type `'sptl'` containing a `SCSpatialSettings` record specifying which compressor to use, the depth, and the spatial quality.

Special Considerations

This function is used for internal communication between the base and format-specific graphics exporter. Applications will not usually need to call it. Format-specific exporters may delegate this call, in which case the base graphics exporter's implementation gives a reply of FALSE.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportDoExport

Performs a graphics export operation.

```
ComponentResult GraphicsExportDoExport (
   GraphicsExportComponent ci,
   unsigned long *actualSizeWritten
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

actualSizeWritten

Points to a variable to receive the number of bytes written. If you are not interested in this information, pass `NIL`.

Return Value

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

Discussion

Before calling this function , you must specify an input image, using one of the `GraphicsExportSetInput...` functions, and a destination for the output image file, using one of the `GraphicsExportSetOutput...` functions.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Carbon GLSnapshot

Graphic Import-Export

ImproveYourImage

qtgraphics.win

TextNameTool

Declared In

`ImageCompression.h`

GraphicsExportDoStandaloneExport

Performs a standalone graphics export operation.

```
ComponentResult GraphicsExportDoStandaloneExport (
   GraphicsExportComponent ci
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

Return Value

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

Discussion

If both `GraphicsExportCanTranscode` (page 18) and `GraphicsExportCanUseCompressor` (page 19) reply FALSE, the base graphics exporter makes this call of the format-specific exporter to perform the export.

Special Considerations

This function is used for internal communication between the base and format-specific graphics exporter. Applications will not usually need to call it.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportDoTranscode

Performs a graphics export operation by transcoding.

```
ComponentResult GraphicsExportDoTranscode (
   GraphicsExportComponent ci
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics exporter component.

Return Value

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

Discussion

The base graphics exporter makes this call of the format-specific graphics exporter to perform a transcoding export. This function should call `GraphicsExportGetInputDataSize` (page 30) and `GraphicsExportReadInputData` (page 47) to measure and read the input image data, and `GraphicsExportWriteOutputData` (page 70) to write the output image file.

Special Considerations

This function is used for internal communication between the base and format-specific graphics exporter. Applications will not usually need to call it.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportDoUseCompressor

Performs a graphics export operation with compression.

```
ComponentResult GraphicsExportDoUseCompressor (
   GraphicsExportComponent ci,
   void *codecSettingsAtomContainer,
   ImageDescriptionHandle *outDesc
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

codecSettingsAtomContainer

An atom container returned by `GraphicsExportCanUseCompressor` (page 19).

outDesc

Points to an image description handle to receive an `ImageDescription` structure describing the compressed image.

Return Value

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

Discussion

The base graphics exporter makes this call to perform a compressing export.

Special Considerations

This function is used for internal communication between the base and format-specific graphics exporter. Applications will not usually need to call it. Format-specific exporters will normally delegate this call, unless they implement export to a container format like PICT or QuickTime Image. In that case, they will wrap the base exporter's implementation in one that forms the container about the compressed data.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportDrawInputImage

Draws a rectangular portion of the input image in a graphics export operation.

```
ComponentResult GraphicsExportDrawInputImage (
   GraphicsExportComponent ci,
   CGrafPtr gw,
   GDHandle gd,
   const Rect *srcRect,
   const Rect *dstRect
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

gw

A pointer to an offscreen graphics world, color graphics port, or basic graphics port.

gd

A handle to a `GDevice` record. If you pass a pointer to an offscreen graphics world in the `gw` parameter, set this parameter to `NIL` because `GraphicsExportDrawInputImage` ignores this parameter and sets the current device to the device attached to the offscreen graphics world.

srcRect

Specifies a portion of the input image.

dstRect

Specifies where in the drawing environment to draw that portion of the input image.

Return Value

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

Discussion

The `gw` and `gd` parameters specify a drawing environment such as you might pass to `GraphicsExportSetInputGWorld` (page 57). The `srcRect` and `dstRect` boundaries need not be the same width and height; you can use this function to scale the `srcRect` image portion. This would be useful, for example, if you were writing a graphics exporter for a multiple-resolution format.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

ElectricImageComponent

ElectricImageComponent.win

Declared In

`ImageCompression.h`

GraphicsExportGetColorSyncProfile

Gets the current value of the ColorSync profile for a graphics export operation.

```
ComponentResult GraphicsExportGetColorSyncProfile (
   GraphicsExportComponent ci,
   Handle *colorSyncProfile
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

colorSyncProfile

Points to a variable to receive the ColorSync profile as a newly allocated handle.

Return Value

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

Special Considerations

The caller is responsible for disposing of the returned handle.

Version Notes

Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportGetCompressionMethod

Returns the compression method for a graphics export operation.

```
ComponentResult GraphicsExportGetCompressionMethod (
   GraphicsExportComponent ci,
   long *compressionMethod
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

compressionMethod

Points to a value to receive the compression method.

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

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportGetCompressionQuality

Returns the compression quality value for a graphics export operation.

```
ComponentResult GraphicsExportGetCompressionQuality (
   GraphicsExportComponent ci,
   CodecQ *spatialQuality
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

spatialQuality

Points to a variable to receive a quality constant (see below). See these constants:

`codecMinQuality`

`codecLowQuality`

`codecNormalQuality`

`codecHighQuality`

`codecMaxQuality`

`codecLosslessQuality`

Return Value

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

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportGetDefaultFileNameExtension

Returns the suggested file name extension for a graphics export operation.

```
ComponentResult GraphicsExportGetDefaultFileNameExtension (
   GraphicsExportComponent ci,
   OSType *fileNameExtension
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

fileNameExtension

Points to a location to receive the file name extension.

Return Value

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

Discussion

File name extensions are returned as upper-case big-endian four-character codes. For example, the extension `.png` would be returned as `'PNG '` (0x504E4720).

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

ThreadsExporter

Declared In

`ImageCompression.h`

GraphicsExportGetDefaultFileTypeAndCreator

Returns the suggested file type and creator for a graphics export operation.

```
ComponentResult GraphicsExportGetDefaultFileTypeAndCreator (
   GraphicsExportComponent ci,
   OSType *fileType,
   OSType *fileCreator
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

fileType

Points to a location to receive the suggested file type for the image file format. If you don't need this information, pass NIL.

fileCreator

Points to a location to receive the suggested file creator for the new image file format. If you don't need this information, pass NIL.

Return Value

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

Discussion

This function, along with GraphicsExportGetDefaultFileNameExtension (page 25) and GraphicsExportGetMIMETypeList (page 39), returns information about the image format supported by a graphics exporter. Format-specific exporters must implement all three of these calls.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

ImageCompression.h

GraphicsExportGetDepth

Returns the current depth setting for a graphics export operation.

```
ComponentResult GraphicsExportGetDepth (
   GraphicsExportComponent ci,
   long *depth
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

depth

Points to a variable to receive the depth.

Return Value

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

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportGetDontRecompress

Determines whether the original compressed data for a graphics export operation will not be decompressed and recompressed, but be copied through to the output file.

```
ComponentResult GraphicsExportGetDontRecompress (
   GraphicsExportComponent ci,
   Boolean *dontRecompress
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

dontRecompress

Points to a Boolean to receive the recompression setting.

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

Discussion
Even though it is not decompressed and recompressed, graphics data may be modified when it is copied through.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportGetExifEnabled

Returns the graphics exporter's current Exif export setting.

```
ComponentResult GraphicsExportGetExifEnabled (
   GraphicsExportComponent ci,
   Boolean *exifEnabled
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component. This function is supported only by the TIFF and JPEG graphics exporters.

exifEnabled

Pass a pointer to a variable that will be set to TRUE if `Exif` export is enabled.

Return Value

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

Version Notes

Introduced in QuickTime 6.

Availability

Available in Mac OS X v10.1 and later.

Declared In

`ImageCompression.h`

GraphicsExportGetInputCGBitmapContext

Retrieves the CGBitmapContext that the graphics exporter is using as its input image.

```
ComponentResult GraphicsExportGetInputCGBitmapContext (
   GraphicsExportComponent ci,
   CGContextRef *bitmapContextRefOut
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

bitmapContextRef

A reference to the Core Graphics context.

Return Value

See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 6.4.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ImageCompression.h`

GraphicsExportGetInputCGImage

Determines which Core Graphics CGImage is the source for a graphics export operation.

```
ComponentResult GraphicsExportGetInputCGImage (
   GraphicsExportComponent ci,
   CGImageRef *imageRefOut
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

imageRef

A reference to a Core Graphics image.

Return Value

See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 6.4.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ImageCompression.h`

GraphicsExportGetInputDataReference

Returns the current input data reference for a graphics export operation.

```
ComponentResult GraphicsExportGetInputDataReference (
   GraphicsExportComponent ci,
   Handle *dataRef,
   OSType *dataRefType
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

dataRef

Points to a variable to receive the data reference handle.

dataRefType

Points to a variable to receive the `data` reference type.

Return Value

See `Error Codes`. If the current source is not a data reference, the function returns `paramErr`. The function returns `noErr` if there is no error.

Discussion

You can use this function to get the source of a graphics export operation. The source can be a QuickTime graphics importer component instance, a QuickDraw `Picture`, a graphics world, a `PixMap` structure, or a piece of compressed data described by an `ImageDescription` structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Special Considerations

The caller is responsible for disposing of the returned data reference handle.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In
`ImageCompression.h`

GraphicsExportGetInputDataSize

Returns the number of bytes of original image data that can be read in a graphics export operation.

```
ComponentResult GraphicsExportGetInputDataSize (
   GraphicsExportComponent ci,
   unsigned long *size
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

size

Points to a variable to receive the size in bytes.

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

Discussion
This function is used by format-specific graphics exporters when transcoding. Applications will not normally need to call this function.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportGetInputFile

Returns the current input file for a graphics export operation.

```
ComponentResult GraphicsExportGetInputFile (
   GraphicsExportComponent ci,
   FSSpec *theFile
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

theFile

A pointer to the file specification of the file containing the graphics data.

Return Value
See `Error Codes`. Returns `noErr` if there is no error. If the current source is not a file, the function returns `paramErr`.

Discussion
You can use this function to get the source of a graphics export operation. The source can be a QuickTime graphics importer component instance, a QuickDraw `Picture`, a graphics world, a `PixMap` structure, or a piece of compressed data described by an `ImageDescription` structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportGetInputGraphicsImporter

Returns the current input graphics importer instance for a graphics export operation.

```
ComponentResult GraphicsExportGetInputGraphicsImporter (
   GraphicsExportComponent ci,
   GraphicsImportComponent *grip
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

grip

Points to a variable to receive the source graphics importer.

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

Discussion
You must get the source of a graphics export operation. The source can be a QuickTime graphics importer component instance, a QuickDraw `Picture`, a graphics world, a `PixMap` structure, or a piece of compressed data described by an `ImageDescription` structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportGetInputGWorld

Returns the current input graphics world for a graphics export operation.

```
ComponentResult GraphicsExportGetInputGWorld (
   GraphicsExportComponent ci,
   GWorldPtr *gworld
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

gworld

Points to a variable to receive the source graphics world.

Return Value

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

Discussion

You can use this function to get the source of a graphics export operation. The source can be a QuickTime graphics importer component instance, a QuickDraw `Picture`, a graphics world, a `PixMap` structure, or a piece of compressed data described by an `ImageDescription` structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportGetInputHandle

Returns the current input handle for a graphics export operation.

```
ComponentResult GraphicsExportGetInputHandle (
   GraphicsExportComponent ci,
   Handle *h
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

h

A pointer to receive the handle.

Return Value

See `Error Codes`. Returns `noErr` if there is no error. If the current source is not a handle, the function returns `paramErr`.

Discussion

You can use this function to get the source of a graphics export operation. The source can be a QuickTime graphics importer component instance, a QuickDraw `Picture`, a graphics world, a `PixMap` structure, or a piece of compressed data described by an `ImageDescription` structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportGetInputImageDepth

Returns the depth of the input image for a graphics export operation.

```
ComponentResult GraphicsExportGetInputImageDepth (
   GraphicsExportComponent ci,
   long *inputDepth
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics exporter component.

`inputDepth`

Points to a variable to receive the input image depth.

Return Value

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

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportGetInputImageDescription

Returns an image description describing the input image in a graphics export operation.

```
ComponentResult GraphicsExportGetInputImageDescription (
   GraphicsExportComponent ci,
   ImageDescriptionHandle *desc
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

desc

Points to a variable to receive a handle to an `ImageDescription` structure.

Return Value

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

Discussion

This function returns an `ImageDescription` structure containing information such as the format of the compressed data, its bit depth, natural bounds, and resolution.

Special Considerations

The caller is responsible for disposing of the returned image description handle.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

ElectricImageComponent

ElectricImageComponent.win

Declared In

`ImageCompression.h`

GraphicsExportGetInputImageDimensions

Returns the dimensions of the input image in a graphics export operation.

```
ComponentResult GraphicsExportGetInputImageDimensions (
   GraphicsExportComponent ci,
   Rect *dimensions
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

dimensions

Points to a rectangle to receive the dimensions of the input image.

Return Value

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

Version Notes

Introduced in QuickTime 4.

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

Related Sample Code
ElectricImageComponent

ElectricImageComponent.win

Declared In
`ImageCompression.h`

GraphicsExportGetInputOffsetAndLimit

Retrieves the current input offset and limit in a graphics export operation.

```
ComponentResult GraphicsExportGetInputOffsetAndLimit (
   GraphicsExportComponent ci,
   unsigned long *offset,
   unsigned long *limit
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

offset

Points to a variable to receive the offset. If you don't need this information, pass `NIL`.

limit

Points to a variable to receive the limit. If you don't need this information, pass `NIL`.

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

Discussion
This function is only applicable when the input is a data reference, file, handle or pointer.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportGetInputPicture

Returns the current input picture in a graphics export operation.

```
ComponentResult GraphicsExportGetInputPicture (
   GraphicsExportComponent ci,
   PicHandle *picture
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

picture

Points to a variable to receive the source picture.

Return Value

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

Discussion

You can use this function to get the source of a graphics export operation. The source can be a QuickTime graphics importer component instance, a QuickDraw `Picture`, a graphics world, a `PixMap` structure, or a piece of compressed data described by an `ImageDescription` structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportGetInputPixmap

Returns the current input pixmap in a graphics export operation.

```
ComponentResult GraphicsExportGetInputPixmap (
   GraphicsExportComponent ci,
   PixMapHandle *pixmap
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

pixmap

Points to a variable to receive the source `PixMap` structure.

Return Value

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

Discussion

You can use this function to get the source of a graphics export operation. The source can be a QuickTime graphics importer component instance, a QuickDraw `Picture`, a graphics world, a `PixMap` structure, or a piece of compressed data described by an `ImageDescription` structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportGetInputPtr

Returns the current input pointer in a graphics export operation.

```
ComponentResult GraphicsExportGetInputPtr (
   GraphicsExportComponent ci,
   Ptr *p,
   unsigned long *size
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

p

A pointer to receive a pointer containing the graphics data.

size

A pointer to a value describing the size of the image data in bytes.

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

Discussion
You can use this function to get the source of a graphics export operation. The source can be a QuickTime graphics importer component instance, a QuickDraw `Picture`, a graphics world, a `PixMap` structure, or a piece of compressed data described by an `ImageDescription` structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportGetInterlaceStyle

Returns the interlace style in a graphics export operation.

```
ComponentResult GraphicsExportGetInterlaceStyle (
   GraphicsExportComponent ci,
   unsigned long *interlaceStyle
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

interlaceStyle

Points to a variable to receive the interlace style. Valid values and interpretations are defined by individual exporters. In QuickTime 4, the PNG graphics exporter supports the interlaceStyle settings shown below See these constants:

kQTPNGInterlaceNone

kQTPNGInterlaceAdam7

Return Value

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

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

ImageCompression.h

GraphicsExportGetMetaData

Returns the current user data setting in a graphics export operation.

```
ComponentResult GraphicsExportGetMetaData (
   GraphicsExportComponent ci,
   void *userData
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

userData

A pointer to a UserDataRecord structure.

Return Value

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

Version Notes

Introduced in QuickTime 4. In QuickTime 4, none of the supplied graphics exporters support setting user data.

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

Declared In
`ImageCompression.h`

GraphicsExportGetMIMETypeList

Returns MIME types and other information about the graphics format in a graphics export operation.

```
ComponentResult GraphicsExportGetMIMETypeList (
   GraphicsExportComponent ci,
   void *qtAtomContainerPtr
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

qtAtomContainerPtr

Receives a newly-created QuickTime atom container that contains information about the graphics format.

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

Discussion
This function creates and returns a QuickTime atom container that contains the format's name, as a string in an atom of type `'desc'` (`kMimeInfoDescriptionTag`), and optionally the MIME type as a string in an atom of type `'mime'[atom]` (`kMimeInfoMimeTypeTag`).

Version Notes
Introduced in QuickTime 4.

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

Related Sample Code
ThreadsExporter

Declared In
`ImageCompression.h`

GraphicsExportGetOutputDataReference

Gets the output data reference handle in a graphics export operation.

```
ComponentResult GraphicsExportGetOutputDataReference (
   GraphicsExportComponent ci,
   Handle *dataRef,
   OSType *dataRefType
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

dataRef

Points to a variable to receive the data reference handle.

dataRefType

Points to a variable to receive a constant that identifies the `data` reference type. See `Data References`.

Return Value

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

Special Considerations

The caller is responsible for disposing of the returned data reference handle.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportGetOutputFile

Returns the current output file for a graphics export operation.

```
ComponentResult GraphicsExportGetOutputFile (
   GraphicsExportComponent ci,
   FSSpec *theFile
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

theFile

Points to a variable to receive the `FSSpec`.

Return Value

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

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In
`ImageCompression.h`

GraphicsExportGetOutputFileTypeAndCreator

Gets the type and creator codes for the output file in a graphics export operation.

```
ComponentResult GraphicsExportGetOutputFileTypeAndCreator (
   GraphicsExportComponent ci,
   OSType *fileType,
   OSType *fileCreator
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

fileType

Receives the file type for the new image file. See `File Types and Creators`.

fileCreator

Receives the file creator for the new image file. See `File Types and Creators`.

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

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportGetOutputHandle

Returns the current output handle for a graphics export operation.

```
ComponentResult GraphicsExportGetOutputHandle (
   GraphicsExportComponent ci,
   Handle *h
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

h

Points to a variable to receive the handle.

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

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportGetOutputMark

Returns the current file position for a graphics export operation.

```
ComponentResult GraphicsExportGetOutputMark (
   GraphicsExportComponent ci,
   unsigned long *mark
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

mark

Receives the current file position, as a byte offset from the beginning of the output data reference.

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

Special Considerations

Not all output data types support the current file position feature.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportGetOutputOffsetAndMaxSize

Returns the output starting offset and maximum size limit for a graphics export operation.

```
ComponentResult GraphicsExportGetOutputOffsetAndMaxSize (
   GraphicsExportComponent ci,
   unsigned long *offset,
   unsigned long *maxSize,
   Boolean *truncateFile
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

offset

On return, a value describing the byte offset of the image data from the beginning of the data reference. If you are not interested in this information, you may pass `NIL`.

maxSize

On return, a value describing the maximum size limit. If you are not interested in this information, you may pass `NIL`.

truncateFile

A Boolean value; TRUE means to truncate the file, FALSE means not.

Return Value

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

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportGetProgressProc

Returns the current progress function for a graphics export operation.

```
ComponentResult GraphicsExportGetProgressProc (
   GraphicsExportComponent ci,
   ICMProgressProcRecordPtr progressProc
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

progressProc

A pointer to an `ICMProgressProc` callback.

Return Value

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

Discussion

By default, graphics export components have no progress functions.

Special Considerations

This function is always implemented by the base graphics exporter.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportGetResolution

Determines the resolution of a graphics exporter component.

```
ComponentResult GraphicsExportGetResolution (
   GraphicsExportComponent ci,
   Fixed *horizontalResolution,
   Fixed *verticalResolution
);
```

Parameters

ci

A component instance. Your software obtains this reference from `OpenComponent` or `OpenDefaultComponent`.

horizontalResolution

Points to a variable to receive the horizontal resolution.

verticalResolution

Points to a variable to receive the vertical resolution.

Return Value

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

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportGetSettingsAsAtomContainer

Retrieves the current settings from a graphics exporter component.

```
ComponentResult GraphicsExportGetSettingsAsAtomContainer (
   GraphicsExportComponent ci,
   void *qtAtomContainerPtr
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

qtAtomContainerPtr

Points to a variable to receive a new QuickTime atom container containing the current graphics exporter component settings.

Return Value

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

Special Considerations

The caller is responsible for disposing of the returned atom container.

Version Notes

Introduced in QuickTime 4.

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

Related Sample Code
Fiendishthngs

Declared In
`ImageCompression.h`

GraphicsExportGetSettingsAsText

Retrieves the current settings from the graphics export component in a user-readable format.

```
ComponentResult GraphicsExportGetSettingsAsText (
   GraphicsExportComponent ci,
   Handle *theText
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

theText

Points to a variable to receive a newly-allocated handle containing text.

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

Special Considerations

The caller is responsible for disposing of the returned handle.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportGetTargetDataSize

Returns the current desired maximum data size for a graphics export operation.

```
ComponentResult GraphicsExportGetTargetDataSize (
   GraphicsExportComponent ci,
   unsigned long *targetDataSize
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

targetDataSize

Points to a variable to receive the desired maximum data size in bytes.

Return Value

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

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportGetThumbnailEnabled

Returns the current thumbnail creation setting for the graphics exporter when exporting Exif files.

```
ComponentResult GraphicsExportGetThumbnailEnabled (
   GraphicsExportComponent ci,
   Boolean *thumbnailEnabled,
   long *maxThumbnailWidth,
   long *maxThumbnailHeight
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component. This function is supported only by the TIFF and JPEG graphics exporters.

thumbnailEnabled

Points to a variable to receive the current thumbnail setting. Pass `NIL` if you do not want to receive this information.

maxThumbnailWidth

Points to a variable to receive the current maximum thumbnail width. Pass `NIL` if you do not want to receive this information.

maxThumbnailHeight

Points to a variable to receive the current maximum thumbnail height. Pass `NIL` if you do not want to receive this information.

Return Value

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

Version Notes

Introduced in QuickTime 6.

Availability

Available in Mac OS X v10.1 and later.

Declared In

`ImageCompression.h`

GraphicsExportMayExporterReadInputData

Asks whether the image source for a graphics export operation is in a form that the exporter can read.

```
ComponentResult GraphicsExportMayExporterReadInputData (
   GraphicsExportComponent ci,
   Boolean *mayReadInputData
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

mayReadInputData

Points to a Boolean; TRUE means that the image source is in a form that the exporter can read, FALSE means it is not.

Return Value

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

Discussion

Some kinds of image source, such as files and handles, form a stream of bytes that can be read directly. Others, such as pictures and pixmaps, do not. Format-specific graphics exporters usually cannot transcode if they cannot read the original data, so those exporters which implement `GraphicsExportCanTranscode` (page 18) will usually first call `GraphicsExportMayExporterReadInputData`.

Special Considerations

This function is used by format-specific graphics exporters when transcoding. Applications will not normally need to call this function.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportReadInputData

Reads the original image data in a graphics export operation.

```
ComponentResult GraphicsExportReadInputData (
   GraphicsExportComponent ci,
   void *dataPtr,
   unsigned long dataOffset,
   unsigned long dataSize
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

dataPtr

A pointer to a memory block to receive the data.

dataOffset

The offset of the image data within the source image data. The function begins reading image data from this offset.

dataSize

The number of bytes of image data to read.

Return Value

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

Discussion

This function communicates with the appropriate data handler to retrieve image data.

Special Considerations

This function is used by format-specific graphics exporters when transcoding. Applications will not normally need to call this function.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportReadOutputData

Reads output image data in a graphics export operation.

```
ComponentResult GraphicsExportReadOutputData (
   GraphicsExportComponent ci,
   void *dataPtr,
   unsigned long dataOffset,
   unsigned long dataSize
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

dataPtr

A pointer to a memory block to receive the data.

dataOffset

The offset of the image data within the data reference. The function begins reading image data from this offset.

dataSize

The number of bytes of image data to read.

Return Value

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

Special Considerations

Not all output data types support this function.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportRequestSettings

Displays a dialog for the user to configure graphics exporter settings, if applicable.

```
ComponentResult GraphicsExportRequestSettings (
   GraphicsExportComponent ci,
   ModalFilterYDUPP filterProc,
   void *yourDataProc
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics exporter component.

`filterProc`

A `ModalFilterYDProc` callback. If you don't need one, pass `NIL`.

`yourDataProc`

An extra parameter that will be passed to the `ModalFilterProc` callback when it is called. If you don't need one, pass `NIL`.

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

Discussion
Some graphics exporters don't support settings dialogs, and so don't implement this call. To find out whether a graphics exporter implements this call, you can use this code:

```
CallComponentCanDo( myGraphicsExporter,
                    kGraphicsExportRequestSettingsSelect);
```

Version Notes
Introduced in QuickTime 4.

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

Related Sample Code
Graphic Import-Export

ImproveYourImage

Declared In
`ImageCompression.h`

GraphicsExportSetColorSyncProfile

Sets the ColorSync profile to embed in the image file for a graphics export operation.

```
ComponentResult GraphicsExportSetColorSyncProfile (
   GraphicsExportComponent ci,
   Handle colorSyncProfile
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

colorSyncProfile

A handle to the ColorSync profile.

Return Value

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

Discussion

ColorSync profiles allow image files to describe their native colorspace in a self-contained manner. They can be stored in atoms of type `'iicc'`.

Version Notes

Introduced in QuickTime 4. Starting with QuickTime 4, the JPEG, PNG, PICT, QuickTime Image and TIFF graphics exporters support embedded ColorSync profiles.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportSetCompressionMethod

Defines the compression method to use in a graphics export operation.

```
ComponentResult GraphicsExportSetCompressionMethod (
   GraphicsExportComponent ci,
   long compressionMethod
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

compressionMethod

A value (see below) describing the compression algorithm to be used by the graphics exporter. See these constants:

`kQTTIFFCompression_None`

`kQTTIFFCompression_PackBits`

Return Value

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

Discussion
In QuickTime 4, the TIFF graphics exporter supports the `compressionMethod` settings `kQTTIFFCompression_None` and `kQTTIFFCompression_PackBits`. Some image formats, such as TIFF, support several compression methods.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportSetCompressionQuality

Defines the compression quality for a graphics export operation.

```
ComponentResult GraphicsExportSetCompressionQuality (
   GraphicsExportComponent ci,
   CodecQ spatialQuality
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

spatialQuality

A constant (see below) that defines the currently specified quality value. See these constants:

`codecMinQuality`

`codecLowQuality`

`codecNormalQuality`

`codecHighQuality`

`codecMaxQuality`

`codecLosslessQuality`

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

Discussion
This setting is only supported by lossy compression methods.

Version Notes
Introduced in QuickTime 4.

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

Related Sample Code
Carbon GLSnapshot

qtgraphics

qtgraphics.win

TextNameTool

ThreadsExporter

Declared In
`ImageCompression.h`

GraphicsExportSetDepth

Defines the depth to use in a graphics export operation.

```
ComponentResult GraphicsExportSetDepth (
   GraphicsExportComponent ci,
   long depth
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics exporter component.

`depth`

A value describing the depth of the image data. Some image file formats support more than one pixel depth.

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

Discussion
The BMP, JPEG, Photoshop, PNG, PICT, QuickTime Image, TGA and TIFF graphics exporters support the depth setting. Some image file formats support more than one pixel depth.

Version Notes
Introduced in QuickTime 4.

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

Related Sample Code
Graphic Import-Export

ImproveYourImage

TextNameTool

Declared In
`ImageCompression.h`

GraphicsExportSetDontRecompress

Requests that the original compressed data for a graphics export operation not be decompressed and recompressed, but be copied through to the output file.

```
ComponentResult GraphicsExportSetDontRecompress (
   GraphicsExportComponent ci,
   Boolean dontRecompress
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

dontRecompress

If TRUE, requests not to recompress the image data.

Return Value

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

Discussion

Even though it is not decompressed and recompressed, graphics data may be modified when it is copied through.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportSetExifEnabled

Determines whether or not the graphics exporter component should create Exif files.

```
ComponentResult GraphicsExportSetExifEnabled (
   GraphicsExportComponent ci,
   Boolean enableExif
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component. This function is supported only by the TIFF and JPEG graphics exporters.

enableExif

Pass TRUE to enable `Exif` file creation.

Return Value

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

Discussion

Turning on `Exif` export disables incompatible settings, such as grayscale JPEG and compressed TIFF, and enables export of `Exif` metadata.

Version Notes

Introduced in QuickTime 6.

Availability

Available in Mac OS X v10.1 and later.

Declared In
`ImageCompression.h`

GraphicsExportSetInputCGBitmapContext

Sets the CGBitmapContext that the graphics exporter will use as its input image.

```
ComponentResult GraphicsExportSetInputCGBitmapContext (
   GraphicsExportComponent ci,
   CGContextRef bitmapContextRef
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics exporter component.

`bitmapContextRef`

A reference to the Core Graphics context.

Return Value
See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 6.4.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ImageCompression.h`

GraphicsExportSetInputCGImage

Specifies a Core Graphics CGImage as the source for a graphics export operation.

```
ComponentResult GraphicsExportSetInputCGImage (
   GraphicsExportComponent ci,
   CGImageRef imageRef
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics exporter component.

`imageRef`

A reference to a `CG` image.

Return Value
See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 6.4.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ImageCompression.h`

GraphicsExportSetInputDataReference

Specifies that the source image for a graphics export operation is a compressed image stored in a data reference.

```
ComponentResult GraphicsExportSetInputDataReference (
   GraphicsExportComponent ci,
   Handle dataRef,
   OSType dataRefType,
   ImageDescriptionHandle desc
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics exporter component.

`dataRef`

A QuickTime data reference. See `Data References`.

`dataRefType`

The type of the data reference; see `Data References`.

`desc`

A handle to an `ImageDescription` structure, describing the compressed data.

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

Discussion
You can use this function to specify a source before you call `GraphicsExportDoExport` (page 19). The source can be a QuickTime graphics importer component instance, a QuickDraw `Picture`, a graphics world, a `PixMap` structure, or a piece of compressed data described by an `ImageDescription` structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportSetInputFile

Specifies that the source image for a graphics export operation is a compressed image stored in a file.

```
ComponentResult GraphicsExportSetInputFile (
   GraphicsExportComponent ci,
   const FSSpec *theFile,
   ImageDescriptionHandle desc
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

theFile

A pointer to the `FSSpec` structure for the file containing the graphics data.

desc

A handle to an `ImageDescription` structure that describes the compressed data.

Return Value

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

Discussion

You can use this function to specify a source before you call `GraphicsExportDoExport` (page 19). The source can be a QuickTime graphics importer component instance, a QuickDraw `Picture`, a graphics world, a `PixMap` structure, or a piece of compressed data described by an `ImageDescription` structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportSetInputGraphicsImporter

Specifies that the source image for a graphics export operation is to be drawn by a graphics importer instance.

```
ComponentResult GraphicsExportSetInputGraphicsImporter (
   GraphicsExportComponent ci,
   GraphicsImportComponent grip
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

grip

The source graphics importer component instance.

Return Value

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

Discussion

You can use this function to specify a source before you call GraphicsExportDoExport (page 19). The source can be a QuickTime graphics importer component instance, a QuickDraw Picture, a graphics world, a PixMap structure, or a piece of compressed data described by an ImageDescription structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Special Considerations

It is the caller's responsibility to dispose of the graphics importer.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

qtgraphics

qtgraphics.win

ThreadsExporter

Declared In

`ImageCompression.h`

GraphicsExportSetInputGWorld

Specifies that the source image for a graphics export operation is a graphics world.

```
ComponentResult GraphicsExportSetInputGWorld (
   GraphicsExportComponent ci,
   GWorldPtr gworld
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

gworld

The source graphics world. It must be a real graphics world; you may not pass an ordinary color GrafPort.

Return Value

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

Discussion

You can use this function to specify a source before you call GraphicsExportDoExport (page 19). The source can be a QuickTime graphics importer component instance, a QuickDraw Picture, a graphics world, a PixMap structure, or a piece of compressed data described by an ImageDescription structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Special Considerations

The graphics exporter will never dispose the graphics world.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Carbon GLSnapshot

Graphic Import-Export

ImproveYourImage

TextNameTool

Declared In

`ImageCompression.h`

GraphicsExportSetInputHandle

Specifies that the source image for a graphics export operation is a compressed image referenced by a handle.

```
ComponentResult GraphicsExportSetInputHandle (
   GraphicsExportComponent ci,
   Handle h,
   ImageDescriptionHandle desc
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

h

A handle to graphics data.

desc

A handle to an `ImageDescription` structure that describes the compressed data.

Return Value

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

Discussion

You can use this function to specify a source before you call `GraphicsExportDoExport` (page 19). The source can be a QuickTime graphics importer component instance, a QuickDraw `Picture`, a graphics world, a `PixMap` structure, or a piece of compressed data described by an `ImageDescription` structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In
`ImageCompression.h`

GraphicsExportSetInputOffsetAndLimit

Specifies the portion of an input data reference, file, handle or pointer that a graphics exporter is permitted to read.

```
ComponentResult GraphicsExportSetInputOffsetAndLimit (
   GraphicsExportComponent ci,
   unsigned long offset,
   unsigned long limit
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics exporter component.

`offset`

The byte offset of the input image data from the beginning of the data reference.

`limit`

The offset of the byte following the last byte of the input image data. (If you don't need to apply any limit, pass `(unsigned long)-1`.) Both the `offset` parameter and the `limit` parameter values are relative to the start of the compressed data. `GraphicsExportGetInputDataSize` (page 30) and `GraphicsExportReadInputData` (page 47) take the offset and limit values into account automatically.

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

Discussion
This routine would be useful if, for example, the source was a JPEG image embedded within a larger file.

Special Considerations

This function is only applicable when the input is a data reference, file, handle, or pointer.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportSetInputPicture

Specifies that the source image for a graphics export operation is a picture.

```
ComponentResult GraphicsExportSetInputPicture (
   GraphicsExportComponent ci,
   PicHandle picture
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

picture

A handle to the source picture.

Return Value

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

Discussion

You can use this function to specify a source before you call `GraphicsExportDoExport` (page 19). The source can be a QuickTime graphics importer component instance, a QuickDraw `Picture`, a graphics world, a `PixMap` structure, or a piece of compressed data described by an `ImageDescription` structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportSetInputPixmap

Specifies that the source image for a graphics export operation is a pixmap.

```
ComponentResult GraphicsExportSetInputPixmap (
   GraphicsExportComponent ci,
   PixMapHandle pixmap
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

pixmap

The source `PixMap` structure.

Return Value

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

Discussion

You can use this function to specify a source before you call `GraphicsExportDoExport` (page 19). The source can be a QuickTime graphics importer component instance, a QuickDraw `Picture`, a graphics world, a `PixMap` structure, or a piece of compressed data described by an `ImageDescription` structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Special Considerations

It is the caller's responsibility to dispose of the `pixmap`.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportSetInputPtr

Specifies that the source image for a graphics export operation is a compressed image stored at a fixed address in memory.

```
ComponentResult GraphicsExportSetInputPtr (
   GraphicsExportComponent ci,
   Ptr p,
   unsigned long size,
   ImageDescriptionHandle desc
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

p

A pointer to a value the image.

size

A value describing the size of the image data in bytes.

desc

A handle to an `ImageDescription` structure that describes the compressed data.

Return Value

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

Discussion

You can use this function to specify a source before you call `GraphicsExportDoExport` (page 19). The source can be a QuickTime graphics importer component instance, a QuickDraw `Picture`, a graphics world, a `PixMap` structure, or a piece of compressed data described by an `ImageDescription` structure. Compressed data can be in a file, handle, pointer, or other data reference. The application must make sure that the source is not disposed of before the graphics exporter instance is closed or given a new source. All of the get and set functions for these sources are implemented by the base graphics exporter; format-specific importers should delegate all of them.

Version Notes

Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportSetInterlaceStyle

Defines the interlace style for a graphics export operation.

```
ComponentResult GraphicsExportSetInterlaceStyle (
   GraphicsExportComponent ci,
   unsigned long interlaceStyle
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics exporter component.

`interlaceStyle`

The new interlace `style` to use. Valid values and interpretations are defined by individual exporters. In QuickTime 4, the PNG graphics exporter supports the `interlaceStyle` settings shown below.
See these constants:

`kQTPNGInterlaceNone`

`kQTPNGInterlaceAdam7`

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

Discussion
A common use for this function is in the PNG and GIF formats, which rearrange data so that low-resolution images can be displayed from incomplete data streams.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportSetMetaData

Defines supplemental data for a graphics export operation, such as copyright text.

```
ComponentResult GraphicsExportSetMetaData (
   GraphicsExportComponent ci,
   void *userData
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

userData

A pointer to user data. The value you pass should have the type userData, which is a pointer to a `UserDataRecord`.

Return Value

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

Special Considerations

In QuickTime 4, none of the supplied graphics exporters support setting user data.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportSetOutputDataReference

Returns the current output data reference for a graphics export operation.

```
ComponentResult GraphicsExportSetOutputDataReference (
   GraphicsExportComponent ci,
   Handle dataRef,
   OSType dataRefType
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

dataRef

A QuickTime data reference.

dataRefType

The type of the data reference; see `Data References`.

Return Value

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

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In
`ImageCompression.h`

GraphicsExportSetOutputFile

Defines the output file for a graphics export operation.

```
ComponentResult GraphicsExportSetOutputFile (
   GraphicsExportComponent ci,
   const FSSpec *theFile
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

theFile

an `FSSpec` structure that identifies the file.

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

Version Notes
Introduced in QuickTime 4.

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

Related Sample Code
Graphic Import-Export

ImproveYourImage

qtgraphics

qtgraphics.win

ThreadsExporter

Declared In
`ImageCompression.h`

GraphicsExportSetOutputFileTypeAndCreator

Sets the file type and creator codes for the output file of a graphics export operation.

```
ComponentResult GraphicsExportSetOutputFileTypeAndCreator (
   GraphicsExportComponent ci,
   OSType fileType,
   OSType fileCreator
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

fileType

The file type for the new image file, such as `'JPEG'`. See `File Types and Creators`.

fileCreator

The file creator for the new image file. This parameter may be 0, in which case a default file creator for this file type is used. See `File Types and Creators`.

Return Value

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

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportSetOutputHandle

Sets a handle to the output of a graphics export operation.

```
ComponentResult GraphicsExportSetOutputHandle (
   GraphicsExportComponent ci,
   Handle h
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

h

The output handle.

Return Value

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

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

TextNameTool

Declared In

`ImageCompression.h`

GraphicsExportSetOutputMark

Seeks to the specified file position in a graphics export operation.

```
ComponentResult GraphicsExportSetOutputMark (
   GraphicsExportComponent ci,
   unsigned long mark
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

mark

The new file position, specified as a byte offset from the beginning of the output data reference.

Return Value

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

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportSetOutputOffsetAndMaxSize

Specifies the output starting offset and maximum size limit for a graphics export operation.

```
ComponentResult GraphicsExportSetOutputOffsetAndMaxSize (
   GraphicsExportComponent ci,
   unsigned long offset,
   unsigned long maxSize,
   Boolean truncateFile
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

offset

The byte offset of the image data from the beginning of the data reference.

maxSize

A value describing the maximum size limit.

truncateFile

A Boolean value; TRUE means to truncate the file.

Return Value

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

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In
`ImageCompression.h`

GraphicsExportSetProgressProc

Installs a progress function in a graphics export operation.

```
ComponentResult GraphicsExportSetProgressProc (
   GraphicsExportComponent ci,
   ICMProgressProcRecordPtr progressProc
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

progressProc

Points to an `ICMProgressProc` callback. If you pass a value of -1, QuickTime provides a standard progress function. If you want to remove the existing progress function, pass `NIL`.

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

Discussion
This function is always implemented by the base graphics exporter.

Special Considerations

If your progress function does any drawing, you should take care to set a safe graphics state before doing so, and to restore the graphics state afterwards. In particular, the current graphics device may be an offscreen device.

Version Notes
Introduced in QuickTime 4.

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

Related Sample Code
ThreadsExporter

Declared In
`ImageCompression.h`

GraphicsExportSetResolution

Defines the resolution to store in the image file for a graphics export operation.

```
ComponentResult GraphicsExportSetResolution (
   GraphicsExportComponent ci,
   Fixed horizontalResolution,
   Fixed verticalResolution
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

horizontalResolution

A value describing the horizontal resolution of the image, where the upper byte is dots per inch. The value 0x00480000 represents 72.0 dpi.

verticalResolution

A value describing the vertical resolution of the image, where the upper byte is dots per inch. The value 0x00480000 represents 72.0 dpi.

Return Value

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

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsExportSetSettingsFromAtomContainer

Sets the graphics exporter component's current configuration to match the settings in a passed atom container.

```
ComponentResult GraphicsExportSetSettingsFromAtomContainer (
   GraphicsExportComponent ci,
   void *qtAtomContainer
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

qtAtomContainer

A pointer to a QuickTime atom container that contains settings.

Return Value

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

Discussion

The settings atom container may contain atoms other than those expected by the graphics exporter component or may be missing certain atoms. This function will use only the settings it understands.

Version Notes

Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportSetTargetDataSize

Defines a desired maximum data size for a graphics export operation and asks for a quality that does not exceed that size.

```
ComponentResult GraphicsExportSetTargetDataSize (
   GraphicsExportComponent ci,
   unsigned long targetDataSize
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

targetDataSize

A value that describes the maximum size of the image data in bytes.

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

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsExportSetThumbnailEnabled

Determines whether or not the graphics exporter component should create an embedded thumbnail inside an exported Exif file.

```
ComponentResult GraphicsExportSetThumbnailEnabled (
   GraphicsExportComponent ci,
   Boolean enableThumbnail,
   long maxThumbnailWidth,
   long maxThumbnailHeight
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component. This function is currently supported only by the TIFF and JPEG graphics exporters.

enableThumbnail

Pass TRUE to turn thumbnail creation on; otherwise pass FALSE.

maxThumbnailWidth

The maximum width for created thumbnails.

maxThumbnailHeight

The maximum height for created thumbnails. If one maximum dimension is 0, only the other will be used. If both maximum dimensions are 0, the graphics exporter will decide for itself. The graphics exporter will not change the aspect ratio of the `Exif` image when creating the thumbnail, nor will it create a thumbnail larger than the image.

Return Value

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

Discussion

The JPEG graphics exporter can create thumbnails only when writing `Exif` files.

Version Notes

Introduced in QuickTime 6.

Availability

Available in Mac OS X v10.1 and later.

Declared In

`ImageCompression.h`

GraphicsExportWriteOutputData

Writes output image data in a graphics export operation.

```
ComponentResult GraphicsExportWriteOutputData (
   GraphicsExportComponent ci,
   const void *dataPtr,
   unsigned long dataSize
);
```

Parameters

ci

The component instance that identifies your connection to the graphics exporter component.

dataPtr

A pointer to a memory block containing the data.

dataSize

The number of bytes of image data to write.

Return Value

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

Discussion

This function is used by format-specific graphics exporters to write output data.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

ElectricImageComponent

ElectricImageComponent.win

Declared In

`ImageCompression.h`

GraphicsImportCreateCGImage

Imports an image as a Core Graphics CGImage.

```
ComponentResult GraphicsImportCreateCGImage (
   GraphicsImportComponent ci,
   CGImageRef *imageRefOut,
   UInt32 flags
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

imageRefOut

A reference to the `CG` image to be created.

flags

A flag (see below) that determines the settings to use. `kGraphicsImportCreateCGImageUsingCurrentSettings` Use the current settings. See these constants:

`kGraphicsImportCreateCGImageUsingCurrentSettings`

Return Value

See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 6.4.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ImageCompression.h`

GraphicsImportDoesDrawAllPixels

Asks whether the graphics importer expects to draw every pixel.

```
ComponentResult GraphicsImportDoesDrawAllPixels (
   GraphicsImportComponent ci,
   short *drawsAllPixels
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

drawsAllPixels

A pointer to a value (see below) that describes the predicted drawing behavior. See these constants:

`graphicsImporterDrawsAllPixels`

`graphicsImporterDoesntDrawAllPixels`

`graphicsImporterDontKnowIfDrawAllPixels`

Return Value

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

Discussion

Some image file formats permit non-rectangular images or images with transparent regions. When such an image is drawn, not every pixel in the boundary rectangle will be changed. `GraphicsImportDoesDrawAllPixels` lets you try to find out whether this will be the case. For instance, you might choose to erase the area behind the image before drawing. If the graphics import component supports this function, `drawsAllPixels` will contain one of the constants shown above on return.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Graphic Import-Export

ImproveYourImage

Declared In

`ImageCompression.h`

GraphicsImportDoExportImageFileDialog

Presents a dialog box letting the user save an imported image in a foreign file format.

```
ComponentResult GraphicsImportDoExportImageFileDialog (
   GraphicsImportComponent ci,
   const FSSpec *inDefaultSpec,
   StringPtr prompt,
   ModalFilterYDUPP filterProc,
   OSType *outExportedType,
   FSSpec *outExportedSpec,
   ScriptCode *outScriptTag
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

inDefaultSpec

A pointer to an `FSSpec` that suggests a default name for the file. If you don't want to suggest a default name, pass `NIL`.

prompt

A pointer to a prompt string that appears in the standard put dialog box; it may be `NIL`, in which case a default string is used.

filterProc

A modal filter function to be passed to the Mac OS function `CustomPutFile`; see *Inside Macintosh: Files* for more information. If you don't need to filter events, pass `NIL`.

outExportedType

A pointer to a variable that will receive the type of the export file that was chosen by the user. If you don't want this information, pass `NIL`. See `File Types and Creators`.

outExportedSpec

A pointer to a variable that will receive the `FSSpec` of the file that was written. If you don't want this information, pass `NIL`.

outScriptTag

A pointer to a variable that will receive the script system in which the exported file name is to be displayed. See `Localization Codes`. If you don't want this information, pass `NIL`.

Return Value

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

Discussion

This function presents the user with an extended Standard File dialog box that allows the image currently in use by the graphics import component to be exported to a file, in a format of the user's choice.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

qtgraphics

qtgraphics.win

QTGraphicsImport

qtgraphimp

qtgraphimp.win

Declared In

`ImageCompression.h`

GraphicsImportDoExportImageFileToDataRefDialog

Presents a dialog box that lets the user save an imported image in a foreign file format.

```
ComponentResult GraphicsImportDoExportImageFileToDataRefDialog (
   GraphicsImportComponent ci,
   Handle inDataRef,
   OSType inDataRefType,
   CFStringRef prompt,
   ModalFilterYDUPP filterProc,
   OSType *outExportedType,
   Handle *outDataRef,
   OSType *outDataRefType
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

inDefaultDataRef

A data reference that specifies the default export location.

inDefaultDataRefType

The type of the data reference that specifies the default export location.

prompt

A reference to a `CFString` that contains the prompt text string for the dialog.

filterProc

A modal filter function; see `ModalFilterYDProc` in the QuickTime API Reference.

outExportedType

A pointer to an `OSType` entity where the type of the exported file will be returned.

outExportedDataRef

A pointer to an handle where the data reference to the exported file will be returned.

outExportedDataRefType

A pointer to an `OSType` entity where the type of the data reference that points to the exported file will be returned.

Return Value

See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Discussion

This function presents a file dialog that lets the user to specify a file to which the exported data goes and a format into which image data is exported. By using data references, a long file name or Unicode file name can be used as a default file name as well as the name of the file into which the export data goes. This function is equivalent to `GraphicsImportDoExportImageFileDialog`.

Version Notes

Introduced in QuickTime 6.4.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ImageCompression.h`

GraphicsImportDraw

Draws an imported image.

```
ComponentResult GraphicsImportDraw (
   GraphicsImportComponent ci
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

Return Value

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

Discussion

This function draws the image currently in use by the graphics import component to the graphics port and device specified by `GraphicsImportSetGWorld` (page 116). `GraphicsImportDraw` takes into account all settings previously specified for the image, such as the source rectangle, transformation matrix, clipping region, graphics mode, and image quality.

Special Considerations

The base graphics importer's drawing function uses the results of `GraphicsImportGetImageDescription` (page 96) and `GraphicsImportGetDataOffsetAndSize` (page 83) to create a decompression sequence, which it uses to draw the image. Subsequent draw operations with the same connection may reuse the decompression sequence. Other graphics importers may override this behavior.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Graphic Import-Export

ImproveYourImage

qtgraphics

qtgraphics.win

vrmakepano

Declared In

`ImageCompression.h`

GraphicsImportExportImageFile

Saves an imported image in a foreign file format.

```
ComponentResult GraphicsImportExportImageFile (
   GraphicsImportComponent ci,
   OSType fileType,
   OSType fileCreator,
   const FSSpec *fss,
   ScriptCode scriptTag
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

fileType

The file type for the new image file, such as 'JPEG'. See File Types and Creators.

fileCreator

The file creator for the new image file. See File Types and Creators. You may pass 0, in which case a default file creator for this file type is used.

fss

A pointer to the FSSpec structure that identifies the file that is to receive the exported image.

scriptTag

The script system in which the file name is to be displayed; see Localization Codes. If you have established the name and location of the file using one of the Standard File Package functions, use the script code returned in the reply record (reply.sfScript). Otherwise, specify the system script by setting the scriptTag parameter to the value smSystemScript. See *Inside Macintosh: Files* for more information about script specifications.

Return Value

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

Discussion

This function creates a new file containing the image currently in use by the graphics import component. The new file is compressed in a format corresponding to the provided file type. If a non-identity matrix has been applied to the graphics import component, this matrix is applied to the image before export. Since most image formats don't support nonzero top-left coordinates, the matrix is temporarily adjusted to ensure that the exported image's bounds have top-left coordinates at (0,0). If the matrix does not map the graphics import component's source rectangle to a rectangle, there will be extra white space left around the image.

Special Considerations

Graphics import components can save data in several formats, including QuickDraw pictures and QuickTime Image files. This capability is only needed by applications that perform file format translation. Applications that only wish to draw the image can use GraphicsImportDraw (page 74).

Version Notes

In QuickTime 3, the supported export file types are kQTFileTypePicture, kQTFileTypeQuickTimeImage, kQTFileTypeBMP, kQTFileTypeJPEG, and kQTFileTypePhotoShop. QuickTime 4 uses graphics exporter components to implement image export.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

qtgraphics

qtgraphics.win

qtgraphimp

qtgraphimp.win

Declared In
`ImageCompression.h`

GraphicsImportExportImageFileToDataRef

Saves an imported image in a foreign file format.

```
ComponentResult GraphicsImportExportImageFileToDataRef (
   GraphicsImportComponent ci,
   OSType fileType,
   OSType fileCreator,
   Handle dataRef,
   OSType dataRefType
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

fileType

The Mac OS file type for the new file, which determines the file format.

fileCreator

The creator type of the new file.

dataRef

A data reference that specifies a storage location to which the image is to be exported.

dataRefType

The type of the data reference.

Return Value
See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Discussion
This function exports the imported image as a foreign file format specified by `fileType`. The exported data will be saved into a storage location specified by a data reference. You can use data reference functions to create a data reference for a file that has long or Unicode file name. This function is equivalent to `GraphicsImportExportImageFile`.

Version Notes
Introduced in QuickTime 6.4.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ImageCompression.h`

GraphicsImportGetAliasedDataReference

Deprecated.

```
ComponentResult GraphicsImportGetAliasedDataReference (
   GraphicsImportComponent ci,
   Handle *dataRef,
   OSType *dataRefType
);
```

Version Notes

This function is listed for historical purposes only. It may be unsupported or removed in future versions of QuickTime.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportGetAsPicture

Creates a QuickDraw picture handle to an imported image.

```
ComponentResult GraphicsImportGetAsPicture (
   GraphicsImportComponent ci,
   PicHandle *picture
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

picture

Points to a handle to a `Picture` structure that is to receive the image.

Return Value

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

Discussion

This function creates a new QuickDraw picture handle containing the image currently in use by the graphics import component. If possible, the image will remain in the compressed format. For example, if the image is from a JFIF file, the picture will contain compressed JPEG data. It is the responsibility of the caller to dispose of the picture handle.

Special Considerations

Graphics import components can save data in several formats, including QuickDraw pictures and QuickTime Image files. This capability is only needed by applications that perform file format translation. Applications that only wish to draw the image can use `GraphicsImportDraw` (page 74).

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Graphic Import-Export

ImproveYourImage

qtskins

qtskins.win

RollerCoaster.win

Declared In

`ImageCompression.h`

GraphicsImportGetBaseDataOffsetAndSize64

Undocumented

```
ComponentResult GraphicsImportGetBaseDataOffsetAndSize64 (
   GraphicsImportComponent ci,
   wide *offset,
   wide *size
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics importer component.

`offset`

Undocumented

`size`

Undocumented

Return Value

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

Version Notes

Introduced in QuickTime 6.

Availability

Available in Mac OS X v10.1 and later.

Declared In

`ImageCompression.h`

GraphicsImportGetBoundsRect

Returns the bounding rectangle for drawing an imported image.

```
ComponentResult GraphicsImportGetBoundsRect (
   GraphicsImportComponent ci,
   Rect *bounds
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics importer component.

`bounds`

A pointer to a `Rect` structure describing the bounding rectangle that has been defined for the image.

Return Value

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

Discussion

This is a convenience function. It is implemented by calling `GraphicsImportGetMatrix` (page 97) and `GraphicsImportGetNaturalBounds` (page 99) and using the results to calculate the drawing rectangle.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

qteffects.win

qtgraphics

qtgraphics.win

qtstreamsplicer.win

vrmakepano

Declared In

`ImageCompression.h`

GraphicsImportGetClip

Returns the current clipping region for an imported image.

```
ComponentResult GraphicsImportGetClip (
   GraphicsImportComponent ci,
   RgnHandle *clipRgn
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

clipRgn

A handle to the `MacRegion` structure that has been defined as the clipping region for the image. Returns `NIL` if there is no clipping region.

Return Value

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

Discussion

The caller must dispose of the returned region handle.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportGetColorSyncProfile

Returns a ColorSync profile for an imported image, if one is embedded in the image file.

```
ComponentResult GraphicsImportGetColorSyncProfile (
   GraphicsImportComponent ci,
   Handle *profile
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

profile

A pointer to receive a handle containing a ColorSync profile, or `NIL` if the image file does not contain one.

Return Value

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

Discussion

Some graphics importers don't implement this function. The caller is responsible for disposing of the returned handle.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Graphic Import-Export

ImproveYourImage

QTtoCG

Declared In

`ImageCompression.h`

GraphicsImportGetDataFile

Returns the file containing the graphics data for an imported image.

```
ComponentResult GraphicsImportGetDataFile (
   GraphicsImportComponent ci,
   FSSpec *theFile
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

theFile

A pointer in which to return the `FSSpec` structure of the file containing the graphics data.

Return Value

See `Error Codes`. Returns `noErr` if there is no error. If the data source is not a file, the function returns `paramErr`.

Discussion

Use this function to get the file system specification record for the file where the imported graphics data resides.

Special Considerations

Graphics importer components use QuickTime data handler components to obtain their data. Applications, however, will use graphics importer functions rather than directly calling a data handler. Besides `GraphicsImportGetDataFile`, these functions include `GraphicsImportSetDataFile` (page 109), `GraphicsImportSetDataHandle` (page 110), `GraphicsImportGetDataHandle` (page 82), `GraphicsImportSetDataReference` (page 111), `GraphicsImportSetDataReferenceOffsetAndLimit` (page 112), and `GraphicsImportGetDataReferenceOffsetAndLimit` (page 85). These functions allow the data source to be a file, a handle, or a QuickTime data reference. You only need to use these functions if you open the graphics importer component directly. You don't need to call them if you use one of the `GetGraphicsImporter`... functions such as `GetGraphicsImporterForDataRef`. The `GetGraphicsImporter`... functions automatically open the graphics importer component and set its data source.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportGetDataHandle

Returns a handle to imported graphics data.

```
ComponentResult GraphicsImportGetDataHandle (
   GraphicsImportComponent ci,
   Handle *h
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics importer component.

`h`

A pointer in which to return a handle to the graphics data.

Return Value

See `Error Codes`. Returns `noErr` if there is no error. If the data source is not a handle, the function returns `paramErr`.

Discussion

You use this function to get the handle that the graphics data resides in. The handle belongs to the component instance. You shouldn't dispose of it.

Special Considerations

Graphics importer components use QuickTime data handler components to obtain their data. Applications, however, will use graphics importer functions rather than directly calling a data handler. Besides `GraphicsImportGetDataHandle`, these functions include `GraphicsImportSetDataFile` (page 109), `GraphicsImportSetDataHandle` (page 110), `GraphicsImportGetDataFile` (page 81), `GraphicsImportSetDataReference` (page 111), `GraphicsImportSetDataReferenceOffsetAndLimit` (page 112), and `GraphicsImportGetDataReferenceOffsetAndLimit` (page 85). These functions allow the data source to be a file, a handle, or a QuickTime data reference. You only need to use these functions if you open the graphics importer component directly. You don't need to call them if you use one of the `GetGraphicsImporter`... functions such as `GetGraphicsImporterForDataRef`. The `GetGraphicsImporter`... functions automatically open the graphics importer component and set its data source.

Version Notes
Introduced in QuickTime 3 or earlier.

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

Declared In
`ImageCompression.h`

GraphicsImportGetDataOffsetAndSize

Returns the offset and size of the compressed image data within an imported image file.

```
ComponentResult GraphicsImportGetDataOffsetAndSize (
   GraphicsImportComponent ci,
   unsigned long *offset,
   unsigned long *size
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

offset

A pointer to a value describing the byte offset of the image data from the beginning of the data source.

size

A pointer to a value describing the size of the image data in bytes.

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

Discussion
This function returns the offset and size of the actual image data within the data source. By default, the offset returned is 0 and the size returned is the size of the file. However, some graphics import components will override this function to skip over unneeded information at the beginning or end of the file.

Version Notes
Introduced in QuickTime 3 or earlier.

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

Related Sample Code
ElectricImageComponent

ElectricImageComponent.win

Declared In
`ImageCompression.h`

GraphicsImportGetDataOffsetAndSize64

Provides a 64-bit version of GraphicsImportGetDataOffsetAndSize.

```
ComponentResult GraphicsImportGetDataOffsetAndSize64 (
   GraphicsImportComponent ci,
   wide *offset,
   wide *size
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

offset

A pointer to a value describing the byte offset of the image data.

size

A pointer to the size of the data, in bytes.

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

Discussion
Format-specific importers may delegate this function, in which case the base importer's implementation will call the 32-bit equivalent, `GraphicsImportGetDataOffsetAndSize` (page 83). If neither function is implemented by the format-specific importer, then both functions will return an offset of 0 and the full size of the data reference, taking into account any data reference offset and limit.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsImportGetDataReference

Returns a data reference to imported graphics data.

```
ComponentResult GraphicsImportGetDataReference (
   GraphicsImportComponent ci,
   Handle *dataRef,
   OSType *dataReType
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

dataRef

A pointer in which to return a QuickTime data reference. If you don't want this information, pass `NIL`.

dataReType

A pointer to receive the type of the data reference; see `Data References`. If you don't want this information, pass `NIL`.

Return Value

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

Discussion

You use this function to get the data reference that the graphics data resides in. The `GraphicsImportGetDataHandle` (page 82) and `GraphicsImportGetDataFile` (page 81) functions call `GraphicsImportGetDataReference` and then manipulate the result accordingly. The caller should dispose of the returned `dataRef`.

Special Considerations

Graphics importer components use QuickTime data handler components to obtain their data. Applications, however, will use graphics importer functions rather than directly calling a data handler. Besides `GraphicsImportGetDataReference`, these functions include `GraphicsImportSetDataFile` (page 109), `GraphicsImportSetDataHandle` (page 110), `GraphicsImportGetDataFile` (page 81), `GraphicsImportSetDataReference` (page 111), `GraphicsImportSetDataReferenceOffsetAndLimit` (page 112), and `GraphicsImportGetDataReferenceOffsetAndLimit` (page 85). These functions allow the data source to be a file, a handle, or a QuickTime data reference. You only need to use these functions if you open the graphics importer component directly. You don't need to call them if you use one of the `GetGraphicsImporter`... functions such as `GetGraphicsImporterForDataRef`. The `GetGraphicsImporter`... functions automatically open the graphics importer component and set its data source.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportGetDataReferenceOffsetAndLimit

Returns the data reference starting offset and data size limit for an imported image.

```
ComponentResult GraphicsImportGetDataReferenceOffsetAndLimit (
   GraphicsImportComponent ci,
   unsigned long *offset,
   unsigned long *limit
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

offset

A pointer to a value specifying the byte offset of the image data from the beginning of the data reference.

limit

The offset of the byte following the last byte of the image data.

Return Value

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

Discussion

This function returns the values set by the GraphicsImportSetDataReferenceOffsetAndLimit (page 112) function. By default, the offset is 0 and the limit is MaxInt (2^32 - 1).

Special Considerations

Graphics importer components use QuickTime data handler components to obtain their data. Applications, however, will use graphics importer functions rather than directly calling a data handler. Besides GraphicsImportGetDataReferenceOffsetAndLimit, these functions include GraphicsImportSetDataFile (page 109), GraphicsImportSetDataHandle (page 110), GraphicsImportGetDataFile (page 81), GraphicsImportSetDataReference (page 111), GraphicsImportSetDataReferenceOffsetAndLimit (page 112), and GraphicsImportGetDataReference (page 84). These functions allow the data source to be a file, a handle, or a QuickTime data reference. You only need to use these functions if you open the graphics importer component directly. You don't need to call them if you use one of the GetGraphicsImporter... functions such as GetGraphicsImporterForDataRef. The GetGraphicsImporter... functions automatically open the graphics importer component and set its data source.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

ImageCompression.h

GraphicsImportGetDataReferenceOffsetAndLimit64

Provides a 64-bit version of GraphicsImportGetDataReferenceOffsetAndLimit.

```
ComponentResult GraphicsImportGetDataReferenceOffsetAndLimit64 (
   GraphicsImportComponent ci,
   wide *offset,
   wide *limit
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

offset

A pointer to receive a value specifying the offset of the byte data following the last byte of the image data.

limit

A pointer to the data limit.

Return Value

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

Discussion

The only difference between this function and `GraphicsImportGetDataReferenceOffsetAndLimit` (page 85) is that the `offset` parameter and the `limit` parameter are 64-bit integers instead of 32-bit integers.

Special Considerations

New applications should use this function instead of the 32-bit version.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportGetDefaultClip

Returns the default clipping region for an imported image, if one is stored there.

```
ComponentResult GraphicsImportGetDefaultClip (
   GraphicsImportComponent ci,
   RgnHandle *defaultRgn
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

defaultRgn

A pointer to a handle to a `MacRegion` structure to receive the default clipping region.

Return Value

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

Special Considerations

Most graphics importers don't implement this function. The caller is responsible for disposing of the returned region.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Graphic Import-Export

ImproveYourImage

Declared In

`ImageCompression.h`

GraphicsImportGetDefaultGraphicsMode

Returns the default graphics mode for an imported image, if one is stored there.

```
ComponentResult GraphicsImportGetDefaultGraphicsMode (
   GraphicsImportComponent ci,
   long *defaultGraphicsMode,
   RGBColor *defaultOpColor
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

defaultGraphicsMode

A pointer to receive the graphics mode; see `Graphics Transfer Modes`.

defaultOpColor

A pointer to receive a color; see `Color Constants`.

Return Value

See `Error Codes`. Returns `noErr` if there is no error. If this function returns `badComponentSelector`, you should assume a mode of `ditherCopy`.

Special Considerations

Most graphics importers don't implement this function.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Graphic Import-Export

ImproveYourImage

Declared In
`ImageCompression.h`

GraphicsImportGetDefaultMatrix

Returns the default matrix for an imported image, if one is stored there.

```
ComponentResult GraphicsImportGetDefaultMatrix (
   GraphicsImportComponent ci,
   MatrixRecord *defaultMatrix
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics importer component.

`defaultMatrix`

Receives a matrix record.

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

Discussion
If this function returns `badComponentSelector`, you should assume an identity matrix.

Special Considerations

Most graphics importers don't implement this function.

Version Notes
Introduced in QuickTime 4.

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

Related Sample Code
Graphic Import-Export

ImproveYourImage

Declared In
`ImageCompression.h`

GraphicsImportGetDefaultSourceRect

Returns the default source rectangle for an imported image, if one is stored there.

```
ComponentResult GraphicsImportGetDefaultSourceRect (
   GraphicsImportComponent ci,
   Rect *defaultSourceRect
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics importer component.

defaultSourceRect

Pointer to receive a `Rect` structure that describes the default source rectangle.

Return Value

See `Error Codes`. Returns `noErr` if there is no error. If this function returns `badComponentSelector`, the source rectangle is equal to the image's natural bounds.

Special Considerations

Most graphics importers don't implement this function.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Graphic Import-Export

ImproveYourImage

Declared In

`ImageCompression.h`

GraphicsImportGetDestinationColorSyncProfileRef

Retrieves a ColorSync profile from a graphics importer component.

```
ComponentResult GraphicsImportGetDestinationColorSyncProfileRef (
   GraphicsImportComponent ci,
   CMProfileRef *destinationProfileRef
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

destinationProfileRef

On return, a pointer to an opaque struct containing a ColorSync profile.

Return Value

See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 6.4.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ImageCompression.h`

GraphicsImportGetDestRect

Returns the destination rectangle for an imported image.

```
ComponentResult GraphicsImportGetDestRect (
   GraphicsImportComponent ci,
   Rect *destRect
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

destRect

A pointer to receive a Rect structure that describes the destination rectangle.

Return Value

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

Discussion

If the source rectangle is equal to the natural bounds, this function is equivalent to GraphicsImportGetBoundsRect (page 79).

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

ImageCompression.h

GraphicsImportGetExportImageTypeList

Returns information about available export formats for a graphics importer.

```
ComponentResult GraphicsImportGetExportImageTypeList (
   GraphicsImportComponent ci,
   void *qtAtomContainerPtr
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

qtAtomContainerPtr

A pointer to a QuickTime atom container that is to receive the type list.

Return Value

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

Discussion

This function creates and returns a QuickTime atom container of type 'expo' containing information about the file types that can be exported by the graphics import component. For each file type, the atom container contains the following child atoms: 'ftyp', the exported file type; 'mime'[atom], the MIME type for this format (optional); 'ext ', the suggested file extension for this format; and 'desc', a human-readable name for this format. The 'ftyp' atom contains an OSType; the other atoms contain nonterminated strings.

Special Considerations

It is the responsibility of the caller to dispose of the 'expo' atom container.

Version Notes

In QuickTime 3, the supported export file types are kQTFileTypePicture, kQTFileTypeQuickTimeImage, kQTFileTypeBMP, kQTFileTypeJPEG, and kQTFileTypePhotoShop. In QuickTime 4, the generic graphics importer builds this atom container from the values returned by the installed graphics exporter components.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

qtgraphimp

qtgraphimp.win

Declared In

ImageCompression.h

GraphicsImportGetExportSettingsAsAtomContainer

Retrieves settings for image files exported by the graphics importer.

```
ComponentResult GraphicsImportGetExportSettingsAsAtomContainer (
   GraphicsImportComponent ci,
   void *qtAtomContainerPtr
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

qtAtomContainerPtr

A pointer to a QuickTime atom container that is to receive the settings information.

Return Value

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

Discussion

This function creates and returns a new QuickTime atom container which holds information about how images will be saved by GraphicsImportExportImageFile (page 75).

Special Considerations

It is the responsibility of the caller to dispose of this atom container.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Fiendishthngs

Declared In

ImageCompression.h

GraphicsImportGetFlags

Returns the current flags of a graphics importer component.

```
ComponentResult GraphicsImportGetFlags (
   GraphicsImportComponent ci,
   long *flags
);
```

Parameters

ci

The component instance that identifies your connection to a graphics importer component.

flags

Pointer to a long integer to receive the current flags (see below). See these constants:

`kGraphicsImporterDontDoGammaCorrection`

Return Value

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

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

ColorMatching

Declared In

`ImageCompression.h`

GraphicsImportGetGenericColorSyncProfile

Retrieves the generic colorsync profile for a graphics importer component.

```
ComponentResult GraphicsImportGetGenericColorSyncProfile (
   GraphicsImportComponent ci,
   OSType pixelFormat,
   void *reservedSetToNULL,
   UInt32 flags,
   Handle *genericColorSyncProfileOut
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

pixelFormat

See `Pixel Formats` in the QuickTime API Reference.

reservedSetToNULL

Pass `NIL`.

flags

Currently not used.

genericColorSyncProfileOut

A handle to the the generic `colorsync` profile for the graphics importer.

Return Value

See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 6.4.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ImageCompression.h`

GraphicsImportGetGraphicsMode

Returns the graphics transfer mode for an imported image.

```
ComponentResult GraphicsImportGetGraphicsMode (
   GraphicsImportComponent ci,
   long *graphicsMode,
   RGBColor *opColor
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

graphicsMode

A pointer to a long integer; see `Graphics Transfer Modes`. The function returns the QuickDraw graphics transfer mode setting for the image. Set to `NIL` if you are not interested in this information.

opColor

A pointer to an `RGBColor` structure. The function returns the color currently specified for blend and transparent operations. Set to `NIL` if you are not interested in this information.

Return Value

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

Discussion

Use this function to find out the current graphics transfer mode and color to use for blending and transparent operations. The default graphics mode is `ditherCopy` and the default opColor is 50% gray.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportGetGWorld

Returns the current graphics port and device for drawing an imported image.

```
ComponentResult GraphicsImportGetGWorld (
   GraphicsImportComponent ci,
   CGrafPtr *port,
   GDHandle *gd
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

port

Returns a pointer to the `CGrafPort` structure for the current destination graphics port. Set to `NIL` if you are not interested in this information.

gd

Returns a pointer to the `GDevice` structure for the destination graphics device. Set to `NIL` if you are not interested in this information.

Return Value

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

Discussion

This function returns the graphics port and device that will be used to draw the image. The graphics world is initialized to the current port and device when the graphics importer component is opened.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportGetImageCount

Returns the number of images in an imported image file.

```
ComponentResult GraphicsImportGetImageCount (
   GraphicsImportComponent ci,
   unsigned long *imageCount
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

imageCount

Points to a variable to receive the number of images.

Return Value

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

Discussion

Most image file formats don't support multiple images. Of the image formats supported by QuickTime 4, however, TIFF files can support multiple images, Photoshop files can contain multiple layers and FlashPix files can contain multiple resolutions. The base graphics importer returns a count of 1.

Special Considerations

Format-specific importers for multiple-image formats should override this function; other importers should delegate it.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Graphic Import-Export

ImproveYourImage

qtgraphics

qtgraphics.win

qtmultiimage

Declared In

`ImageCompression.h`

GraphicsImportGetImageDescription

Returns image description information for an imported image.

```
ComponentResult GraphicsImportGetImageDescription (
   GraphicsImportComponent ci,
   ImageDescriptionHandle *desc
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

desc

Points to a handle to an `ImageDescription` structure.

Return Value

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

Discussion

This function returns an `ImageDescription` structure containing information such as the format of the compressed data, its bit depth, natural bounds, and resolution.

Special Considerations

The caller is responsible for disposing of the returned image description handle.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

CTMClip

Graphic Import-Export
ImproveYourImage
TexturePerformanceDemo
TextureRange

Declared In

`ImageCompression.h`

GraphicsImportGetImageIndex

Returns the current image index for an imported image.

```
ComponentResult GraphicsImportGetImageIndex (
   GraphicsImportComponent ci,
   unsigned long *imageIndex
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

imageIndex

Points to a variable to receive the image index. Image indexes are one-based; 0 is considered a special index by some importers, and treated the same as 1 by others. The default image index is 1.

Return Value

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

Discussion

The base graphics importer implements this function. Format-specific importers should delegate it.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportGetMatrix

Returns the transformation matrix to be used for drawing an imported image.

```
ComponentResult GraphicsImportGetMatrix (
   GraphicsImportComponent ci,
   MatrixRecord *matrix
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

matrix

A pointer to a `MatrixRecord` structure that defines the transformation matrix that applies to the image.

Return Value

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

Discussion

The transformation matrix is initialized to the identity matrix when the graphics import component is instantiated.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Graphic Import-Export

ImproveYourImage

qtgraphics

qtgraphics.win

Declared In

`ImageCompression.h`

GraphicsImportGetMetaData

Extracts user data from an imported image file.

```
ComponentResult GraphicsImportGetMetaData (
   GraphicsImportComponent ci,
   void *userData
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

userData

A pointer to a `UserDataRecord` structure.

Return Value

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

Discussion

You may create a new user data structure by calling `NewUserData`. Alternatively, you can obtain a pointer to an existing one by calling `GetMovieUserData`, `GetTrackUserData` or `GetMediaUserData`. If the user data passed to `GraphicsImportGetMetaData` belongs to a movie, track or media, then whatever user data is extracted will be added to that movie, track or media.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Graphic Import-Export

ImproveYourImage

Declared In

`ImageCompression.h`

GraphicsImportGetMIMETypeList

Returns a list of MIME types supported by the graphics importer component.

```
ComponentResult GraphicsImportGetMIMETypeList (
   GraphicsImportComponent ci,
   void *qtAtomContainerPtr
);
```

Parameters

`ci`

Specifies an instance of a graphics importer component.

`qtAtomContainerPtr`

A pointer to an atom container that holds a series of atom triplets for each MIME type, including an atom of type `'mime'[atom]` that contains a list of MIME types supported by the graphics import component.

Return Value

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

Discussion

Your graphics import component can support MIME types that correspond to graphics formats it supports. To make a list of these MIME types available to applications or other software, it must implement `GraphicsImportGetMIMETypeList`. To indicate that your graphics import component supports this function, set the `hasMovieImportMIMEList` flag in the `componentFlags` field of your component's `ComponentDescription` structure.

Special Considerations

This function does not access any file-specific information.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportGetNaturalBounds

Returns the bounding rectangle of an imported image.

```
ComponentResult GraphicsImportGetNaturalBounds (
   GraphicsImportComponent ci,
   Rect *naturalBounds
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

naturalBounds

A pointer to a `Rect` structure that describes the size of the bounding rectangle for the image.

Return Value

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

Discussion

Use this function to determine the native size of the image associated with a graphics importer component. The natural bounds are always zero-based. This is a convenience function that simply calls `GraphicsImportGetImageDescription` (page 96) and extracts the `width` and height fields.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

DelegateOnlyComponent

Graphic Import-Export

ImproveYourImage

qtgraphics

qtgraphics.win

Declared In

`ImageCompression.h`

GraphicsImportGetOverrideSourceColorSyncProfileRef

Retrieves the override ColorSync profile for a graphics importer component.

```
ComponentResult GraphicsImportGetOverrideSourceColorSyncProfileRef (
   GraphicsImportComponent ci,
   CMProfileRef *outOverrideSourceProfileRef
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

outOverrideSourceProfileRef

A pointer to an opaque struct containing a ColorSync profile.

Return Value

See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Version Notes
Introduced in QuickTime 6.4.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ImageCompression.h`

GraphicsImportGetProgressProc

Returns the current progress function for a graphics import operation.

```
ComponentResult GraphicsImportGetProgressProc (
   GraphicsImportComponent ci,
   ICMProgressProcRecordPtr progressProc
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

progressProc

A pointer to an `ICMProgressProc` callback.

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

Discussion
By default, graphics import components have no progress functions.

Version Notes
Introduced in QuickTime 3 or earlier.

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

Declared In
`ImageCompression.h`

GraphicsImportGetQuality

Returns the image quality value for an imported image.

```
ComponentResult GraphicsImportGetQuality (
   GraphicsImportComponent ci,
   CodecQ *quality
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

quality

A pointer to a constant (see below) that defines the currently specified quality value. See these constants:

`codecMinQuality`

`codecLowQuality`

`codecNormalQuality`

`codecHighQuality`

`codecMaxQuality`

`codecLosslessQuality`

Return Value

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

Discussion

The quality value indicates how precisely the decompressor will decompress the image data. Some decompressors may choose to ignore some image data to improve decompression speed.

Version Notes

With QuickTime 3 the default quality is `codecHighQuality`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportGetSourceRect

Returns the current source rectangle for an imported image.

```
ComponentResult GraphicsImportGetSourceRect (
   GraphicsImportComponent ci,
   Rect *sourceRect
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

sourceRect

A pointer to a `Rect` structure that defines the source rectangle currently specified for the image.

Return Value

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

Discussion

This function returns the current source rectangle, as specified by `GraphicsImportSetSourceRect` (page 122). The default source rectangle is the image's natural bounds.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code
CarbonQTGraphicImport

Declared In
`ImageCompression.h`

GraphicsImportReadData

Reads imported image data.

```
ComponentResult GraphicsImportReadData (
   GraphicsImportComponent ci,
   void *dataPtr,
   unsigned long dataOffset,
   unsigned long dataSize
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

dataPtr

A pointer to a memory block to receive the data.

dataOffset

The offset of the image data within the data reference. The function begins reading image data from this offset.

dataSize

The number of bytes of image data to read.

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

Discussion
This function communicates with the appropriate data handler to retrieve image data. Typically, only developers of graphics importer components will need to use this function. This function should always be used to retrieve data from the data source, rather than reading the data directly. This function automatically honors any offset and limit values set with `GraphicsImportSetDataReferenceOffsetAndLimit` (page 112). For instance, if the offset is set to 100 and `GraphicsImportReadData` is called to read bytes from `dataOffset` 5, it will return bytes starting at actual offset 105.

Special Considerations

This function is used by format-specific graphics import components to read data from the data source. It is implemented by the base graphics importer.

Version Notes
Introduced in QuickTime 3 or earlier.

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

Related Sample Code
ElectricImageComponent

ElectricImageComponent.win

Declared In

`ImageCompression.h`

GraphicsImportReadData64

Provides a 64-bit version of GraphicsImportReadData.

```
ComponentResult GraphicsImportReadData64 (
   GraphicsImportComponent ci,
   void *dataPtr,
   const wide *dataOffset,
   unsigned long dataSize
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

dataPtr

A pointer to a memory block to receive the data.

dataOffset

A pointer to the offset of the image data within the data reference.

dataSize

The number of bytes of image data to read.

Return Value

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

Discussion

This function is a 64-bit analog of `GraphicsImportReadData` (page 103). Format-specific importers may call either version.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportSaveAsPicture

Creates a QuickDraw picture file for an imported image.

```
ComponentResult GraphicsImportSaveAsPicture (
   GraphicsImportComponent ci,
   const FSSpec *fss,
   ScriptCode scriptTag
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

fss

A pointer to an FSSpec structure that defines the file to receive the image.

scriptTag

The script system in which the file name is to be displayed; see Localization Codes. If you have established the name and location of the file using one of the Standard File Package functions, use the script code returned in the reply record (reply.sfScript). Otherwise, specify the system script by setting the scriptTag parameter to the value smSystemScript. See *Inside Macintosh: Files* for more information about script specifications.

Return Value

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

Discussion

This function creates a new QuickDraw picture file containing the image currently in use by the graphics import component. If possible, the image will remain in the compressed format. For example, if the image is from a JFIF file, the picture will contain compressed JPEG data. Applications that only wish to draw the image can use GraphicsImportDraw (page 74).

Special Considerations

Graphics import components can save data in several formats, including QuickDraw pictures and QuickTime Image files. This capability is only needed by applications that perform file format translation.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

ImageCompression.h

GraphicsImportSaveAsPictureToDataRef

Creates a storage location that contains a QuickDraw picture for an imported image.

```
ComponentResult GraphicsImportSaveAsPictureToDataRef (
   GraphicsImportComponent ci,
   Handle dataRef,
   OSType dataRefType
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

dataRef

A data reference that specifies a storage location to which the picture is to be saved.

dataRefType

The type of the data reference.

Return Value

See Error Codes in the QuickTime API Reference. Returns noErr if there is no error.

Discussion

This function saves the imported image as a QuickDraw picture into a storage location specified through a data reference. You can use Data Reference Utilities to create a data reference for a file that has long or Unicode file name. This function is equivalent to `GraphicsImporterSaveAsPictureFile`.

Version Notes

Introduced in QuickTime 6.4.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ImageCompression.h`

GraphicsImportSaveAsQuickTimeImageFile

Creates a QuickTime Image file of an imported image.

```
ComponentResult GraphicsImportSaveAsQuickTimeImageFile (
   GraphicsImportComponent ci,
   const FSSpec *fss,
   ScriptCode scriptTag
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

fss

A pointer to the `FSSpec` that defines the file to receive the image.

scriptTag

The script system in which the file name is to be displayed; see `Localization Codes`. If you have established the name and location of the file using one of the Standard File Package functions, use the script code returned in the reply record (`reply.sfScript`). Otherwise, specify the system script by setting the `scriptTag` parameter to the value `smSystemScript`. See *Inside Macintosh: Files* for more information about script specifications.

Return Value

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

Discussion

This function creates a new QuickTime Image file containing the image currently in use by the graphics import component. If possible, the image remains in the compressed format. For example, if the image is from a JFIF file, the QuickTime Image file will contain compressed JPEG data.

Special Considerations

Graphics import components can save data in several formats, including QuickDraw pictures and QuickTime Image files. This capability is only needed by applications that perform file format translation. Applications that only wish to draw the image can use the `GraphicsImportDraw` (page 74) function.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportSaveAsQuickTimeImageFileToDataRef

Creates a storage location that contains a QuickTime image of an imported image.

```
ComponentResult GraphicsImportSaveAsQuickTimeImageFileToDataRef (
   GraphicsImportComponent ci,
   Handle dataRef,
   OSType dataRefType
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

dataRef

A data reference that specifies a storage location to which the picture is to be saved.

dataRefType

The type of the data reference.

Return Value

See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Discussion

This function saves the imported image as a QuickTime image into a storage location specified through a data reference. You can use data reference functions to create a data reference for a file that has long or Unicode file name. This function is equivalent to `GraphicsImportSaveAsQuickTimeImageFile`.

Version Notes

Introduced in QuickTime 6.4.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ImageCompression.h`

GraphicsImportSetBoundsRect

Defines the rectangle in which to draw an imported image.

```
ComponentResult GraphicsImportSetBoundsRect (
   GraphicsImportComponent ci,
   const Rect *bounds
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

bounds

A pointer to a `Rect` structure that describes the bounding rectangle into which the image will be drawn.

Return Value

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

Discussion

You use this function to define the rectangle into which the graphics image should be drawn. The function creates a transformation matrix to map the image's natural bounds to the specified bounds and then calls `GraphicsImportSetMatrix` (page 119).

Special Considerations

Because this function affects the transformation matrix, you should use `GraphicsImportSetMatrix` (page 119) instead of this function if you also need to specify more complex transformations of the matrix.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

CarbonQTGraphicImport

qtstreamsplicer.win

ThreadsImporter

vrmakepano

vrmakepano.win

Declared In

`ImageCompression.h`

GraphicsImportSetClip

Defines the clipping region for drawing an imported image.

```
ComponentResult GraphicsImportSetClip (
   GraphicsImportComponent ci,
   RgnHandle clipRgn
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

clipRgn

A handle to a `MacRegion` structure that defines the clipping region in the destination coordinate system. Set to `NIL` to disable clipping. The graphics import component makes a copy of this region.

Return Value

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

Discussion

Because all drawing operations ignore the port clipping region, you must use this function to clip an image. The graphics importer component draws only that portion of the image that lies within the specified clipping region.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

DropDraw

QTGraphicsImport

Declared In

`ImageCompression.h`

GraphicsImportSetDataFile

Specifies the file that contains imported graphics data.

```
ComponentResult GraphicsImportSetDataFile (
   GraphicsImportComponent ci,
   const FSSpec *theFile
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

theFile

A pointer to an `FSSpec` structure that defines the file containing the graphics data. The returned file will be opened for read access.

Return Value

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

Special Considerations

Graphics importer components use QuickTime data handler components to obtain their data. Applications, however, will use graphics importer functions rather than directly calling a data handler. Besides `GraphicsImportSetDataFile`, these functions include `GraphicsImportGetDataFile` (page 81), `GraphicsImportSetDataHandle` (page 110), `GraphicsImportGetDataHandle` (page 82), `GraphicsImportSetDataReference` (page 111), `GraphicsImportSetDataReferenceOffsetAndLimit` (page 112), and `GraphicsImportGetDataReferenceOffsetAndLimit` (page 85). These functions allow the data source to be a file, a handle, or a QuickTime data reference. You only need to use these functions if you open the graphics importer component directly. You don't need to call them if you use one of the `GetGraphicsImporter`... functions such as `GetGraphicsImporterForDataRef`. The `GetGraphicsImporter`... functions automatically open the graphics importer component and set its data source.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportSetDataHandle

Specifies the handle that references imported graphics data.

```
ComponentResult GraphicsImportSetDataHandle (
   GraphicsImportComponent ci,
   Handle h
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

h

Specifies a handle containing graphics data. The format of the data in the handle is the same as that found in a file.

Return Value

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

Discussion

The graphics importer component doesn't make a copy of this data. Therefore, you must not dispose this handle until the graphics importer has been closed.

Special Considerations

Graphics importer components use QuickTime data handler components to obtain their data. Applications, however, will use graphics importer functions rather than directly calling a data handler. Besides `GraphicsImportSetDataHandle`, these functions include `GraphicsImportGetDataFile` (page 81), `GraphicsImportSetDataFile` (page 109), `GraphicsImportGetDataHandle` (page 82), `GraphicsImportSetDataReference` (page 111), `GraphicsImportSetDataReferenceOffsetAndLimit` (page 112), and `GraphicsImportGetDataReferenceOffsetAndLimit` (page 85). These functions allow the data source to be a file, a handle, or a QuickTime data reference. You only need to use these functions if you open the graphics importer component directly. You don't need to call them if you use one of the `GetGraphicsImporter`... functions such as `GetGraphicsImporterForDataRef`. The `GetGraphicsImporter`... functions automatically open the graphics importer component and set its data source.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

CarbonQTGraphicImport

qtdataref

qtmakemovie

ThreadsImporter
ThreadsImportMovie

Declared In

`ImageCompression.h`

GraphicsImportSetDataReference

Specifies the data reference for imported graphics data.

```
ComponentResult GraphicsImportSetDataReference (
   GraphicsImportComponent ci,
   Handle dataRef,
   OSType dataReType
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics importer component.

`dataRef`

A handle to a QuickTime data reference.

`dataReType`

The data reference type. See `Data References`.

Return Value

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

Discussion

Applications typically don't use this function. The `GraphicsImportSetDataFile` (page 109) and `GraphicsImportSetDataHandle` (page 110) functions both call this function, with the appropriate data reference and data reference type. This function makes a copy of the passed data reference, so it is safe to dispose of the handle immediately after the call.

Special Considerations

Graphics importer components use QuickTime data handler components to obtain their data. Applications, however, will use graphics importer functions rather than directly calling a data handler. Besides `GraphicsImportSetDataReference`, these functions include `GraphicsImportGetDataFile` (page 81), `GraphicsImportSetDataHandle` (page 110), `GraphicsImportGetDataHandle` (page 82), `GraphicsImportSetDataFile` (page 109), `GraphicsImportSetDataReferenceOffsetAndLimit` (page 112), and `GraphicsImportGetDataReferenceOffsetAndLimit` (page 85). These functions allow the data source to be a file, a handle, or a QuickTime data reference. You only need to use these functions if you open the graphics importer component directly. You don't need to call them if you use one of the `GetGraphicsImporter`... functions such as `GetGraphicsImporterForDataRef`. The `GetGraphicsImporter`... functions automatically open the graphics importer component and set its data source.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportSetDataReferenceOffsetAndLimit

Specifies the data reference starting offset and data size limit for an imported image.

```
ComponentResult GraphicsImportSetDataReferenceOffsetAndLimit (
   GraphicsImportComponent ci,
   unsigned long offset,
   unsigned long limit
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics importer component.

`offset`

The byte offset of the image data from the beginning of the data reference.

`limit`

A pointer to a value specifying the offset of the byte following the last byte of the image data.

Return Value

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

Discussion

A data reference typically refers to an entire file. However, there are times when the data being referenced is embedded in a larger file. In these cases, it is necessary to indicate where the data begins in the data reference and where it ends. This function lets you specify the starting offset and ending offset. All requests to read data are then relative to the specified offset, and are pinned to the data size, so the graphics import component cannot accidentally read outside the end (or beginning) of the segment.

Special Considerations

Graphics importer components use QuickTime data handler components to obtain their data. Applications, however, will use graphics importer functions rather than directly calling a data handler. Besides `GraphicsImportSetDataReferenceOffsetAndLimit`, these functions include `GraphicsImportGetDataFile` (page 81), `GraphicsImportSetDataHandle` (page 110), `GraphicsImportGetDataHandle` (page 82), `GraphicsImportSetDataReference` (page 111), `GraphicsImportSetDataFile` (page 109), and `GraphicsImportGetDataReferenceOffsetAndLimit` (page 85). These functions allow the data source to be a file, a handle, or a QuickTime data reference. You only need to use these functions if you open the graphics importer component directly. You don't need to call them if you use one of the `GetGraphicsImporter`... functions such as `GetGraphicsImporterForDataRef`. The `GetGraphicsImporter`... functions automatically open the graphics importer component and set its data source.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportSetDataReferenceOffsetAndLimit64

Provides a 64-bit version of GraphicsImportSetDataReferenceOffsetAndLimit.

```
ComponentResult GraphicsImportSetDataReferenceOffsetAndLimit64 (
   GraphicsImportComponent ci,
   const wide *offset,
   const wide *limit
);
```

Parameters

ci

A component instance. Your software obtains this reference from OpenComponent or OpenDefaultComponent.

offset

A pointer to a value specifying the byte offset of the image data from the beginning of the data source.

limit

A pointer to a value specifying the offset of the byte following the last byte of the image data.

Return Value

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

Discussion

This function is a 64-bit analog of GraphicsImportSetDataReferenceOffsetAndLimit (page 112).

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Declared In

ImageCompression.h

GraphicsImportSetDestinationColorSyncProfileRef

Sets the ColorSync profile for a graphics importer component.

```
ComponentResult GraphicsImportSetDestinationColorSyncProfileRef (
   GraphicsImportComponent ci,
   CMProfileRef newDestinationProfileRef
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

newDestinationProfileRef

A pointer to an opaque struct containing a ColorSync profile.

Return Value

See Error Codes in the QuickTime API Reference. Returns noErr if there is no error.

Version Notes

Introduced in QuickTime 6.4.

Availability
Available in Mac OS X v10.3 and later.

Declared In
`ImageCompression.h`

GraphicsImportSetDestRect

Sets the destination rectangle for a graphics import operation.

```
ComponentResult GraphicsImportSetDestRect (
   GraphicsImportComponent ci,
   const Rect *destRect
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

destRect

Points to a `Rect` structure.

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

Discussion
Use this function to define the rectangle into which the extracted source rectangle should be drawn. This function creates a transformation matrix to map the source rectangle to the specified destination rectangle and then calls the `GraphicsImportSetMatrix` (page 119) function.

Version Notes
Introduced in QuickTime 4.

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

Declared In
`ImageCompression.h`

GraphicsImportSetExportSettingsFromAtomContainer

Determines settings for the export of imported image files.

```
ComponentResult GraphicsImportSetExportSettingsFromAtomContainer (
   GraphicsImportComponent ci,
   void *qtAtomContainer
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

qtAtomContainer

A pointer to a QuickTime atom container that holds new settings information.

Return Value

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

Discussion

This function extracts export settings from a QuickTime atom container. These settings configure how images will be saved by GraphicsImportExportImageFile (page 75).

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

ImageCompression.h

GraphicsImportSetFlags

Sets the flags for a graphics importer component.

```
ComponentResult GraphicsImportSetFlags (
   GraphicsImportComponent ci,
   long flags
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

flags

The new flags (see below) to use. See these constants:

kGraphicsImporterDontDoGammaCorrection

Return Value

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

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

ColorMatching

DropDraw

ImproveYourImage

SampleDS

Declared In

ImageCompression.h

GraphicsImportSetGraphicsMode

Sets the graphics transfer mode for an imported image.

```
ComponentResult GraphicsImportSetGraphicsMode (
   GraphicsImportComponent ci,
   long graphicsMode,
   const RGBColor *opColor
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

graphicsMode

The graphics transfer mode to use for drawing the image; see `Graphics Transfer Modes`.

opColor

A pointer to an `RGBColor` structure that describes the color to use for blending and transparent operations.

Return Value

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

Discussion

Use this function to specify the graphics transfer mode and color to use for blending and transparent operations.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Graphic Import-Export

ImproveYourImage

Declared In

`ImageCompression.h`

GraphicsImportSetGWorld

Sets the graphics port and device for drawing an imported image.

```
ComponentResult GraphicsImportSetGWorld (
   GraphicsImportComponent ci,
   CGrafPtr port,
   GDHandle gd
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

port

A pointer to the `CGrafPort` structure that defines the destination graphics port or graphics world. Set to `NIL` to use the current port.

gd

A handled to the `GDevice` structure that defines the destination graphics device. Set to `NIL` to use the current device. If the `port` parameter specifies a graphics world, set this parameter to `NIL` to use that graphics world's device.

Return Value

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

Discussion

The graphics world is initialized to the current port and device when the graphics importer component is opened. Use this function to select another port or device.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Graphic Import-Export

ImproveYourImage

qtgraphics.win

qtstreamsplicer.win

vrmakepano

Declared In

`ImageCompression.h`

GraphicsImportSetImageIndex

Specifies the image index for an imported image.

```
ComponentResult GraphicsImportSetImageIndex (
   GraphicsImportComponent ci,
   unsigned long imageIndex
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

imageIndex

The image index. Image indexes are one-based; 0 is considered a special index by some importers, and treated the same as 1 by others. The default image index is 1.

Return Value

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

Discussion

The base graphics importer ensures that the image index is no greater than the image count returned by `GraphicsImportGetImageCount` (page 95).

Special Considerations

The base graphics importer implements this function. Format-specific importers should delegate it.

Version Notes

Introduced in QuickTime 4.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Graphic Import-Export

ImproveYourImage

qtgraphics

qtgraphics.win

qtmultiimage

Declared In

`ImageCompression.h`

GraphicsImportSetImageIndexToThumbnail

Looks for a graphics subimage that contains a thumbnail.

```
ComponentResult GraphicsImportSetImageIndexToThumbnail (
   GraphicsImportComponent ci
);
```

Parameters

`ci`

The component instance that identifies your connection to the graphics importer component.

Return Value

See `Error Codes`. If the function does not find a thumbnail, it returns `noThumbnailFoundErr`. It returns `noErr` if there is no error.

Discussion

This function looks for a subimage that contains a thumbnail. If the function finds one, it sets the image index to that subimage. The base graphics importer's implementation of `SetImageIndexToThumbnail` works by looking for the first image index that returns a `kQTIndexedImageType` metadata item containing the `kQTIndexedImageIsThumbnail` tag. Format-specific graphics importers may override this process with more efficient algorithms.

Version Notes

Introduced in QuickTime 6.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ImageCompression.h`

GraphicsImportSetMatrix

Defines the transformation matrix to use for drawing an imported image.

```
ComponentResult GraphicsImportSetMatrix (
   GraphicsImportComponent ci,
   const MatrixRecord *matrix
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

matrix

A pointer to a matrix structure that specifies how to transform the image during decompression. For example, you can use a transformation matrix to scale or rotate the image. To set the matrix to identity, pass `NIL` in this parameter.

Return Value

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

Discussion

This function establishes the transformation matrix to be applied to an image, which determines where and how it will be drawn.

Special Considerations

This function affects the bounding rectangle defined for the image. You can specify where an image will be drawn by setting either a transformation matrix or a bounding rectangle, but it is usually more convenient for applications to set a bounding rectangle using the `GraphicsImportSetBoundsRect` (page 107) function.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

DropDraw

Graphic Import-Export

ImproveYourImage

qtgraphics

qtgraphics.win

Declared In

`ImageCompression.h`

GraphicsImportSetOverrideSourceColorSyncProfileRef

Sets the override ColorSync profile for a graphics importer component.

```
ComponentResult GraphicsImportSetOverrideSourceColorSyncProfileRef (
   GraphicsImportComponent ci,
   CMProfileRef newOverrideSourceProfileRef
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

newOverrideSourceProfileRef

A pointer to an opaque struct containing a ColorSync profile.

Return Value

See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 6.4.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ImageCompression.h`

GraphicsImportSetProgressProc

Installs a progress procedure to call while drawing an imported image.

```
ComponentResult GraphicsImportSetProgressProc (
   GraphicsImportComponent ci,
   ICMProgressProcRecordPtr progressProc
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

progressProc

Points to an `ICMProgressProc` callback. If you pass a value of -1, QuickTime provides a standard progress function. If you want to remove the existing progress function, pass `NIL`.

Return Value

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

Discussion

This function sets a progress function that will be installed in the image decompression sequence used to draw the image.

Special Considerations

If your progress function does any drawing, you should take care to set a safe graphics state before doing so, and to restore the graphics state afterwards. In particular, the current graphics device may be an offscreen device.

Version Notes

Introduced in QuickTime 3 or earlier.

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

Related Sample Code
qtdataexchange

qtdataexchange.win

ThreadsExporter

ThreadsImporter

Declared In
`ImageCompression.h`

GraphicsImportSetQuality

Sets the image quality value for an imported image.

```
ComponentResult GraphicsImportSetQuality (
   GraphicsImportComponent ci,
   CodecQ quality
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

quality

Contains a constant (see below) that defines the desired image quality for decompression. Values for this parameter are on the same scale as compression quality. See these constants:

`codecMinQuality`

`codecLowQuality`

`codecNormalQuality`

`codecHighQuality`

`codecMaxQuality`

`codecLosslessQuality`

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

Discussion
The quality parameter controls how precisely the decompressor decompresses the image data. Some decompressors may choose to ignore some image data to improve decompression speed.

Version Notes
Introduced in QuickTime 3 or earlier.

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

Related Sample Code
CTMClip

CTMDemo

TexturePerformanceDemo

TextureRange

ThreadsImporter

Declared In
`ImageCompression.h`

GraphicsImportSetSourceRect

Sets the source rectangle to use for an imported image.

```
ComponentResult GraphicsImportSetSourceRect (
   GraphicsImportComponent ci,
   const Rect *sourceRect
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

sourceRect

A pointer to a `Rect` structure defining the portion of the image to decompress. This rectangle must lie within the boundary rectangle of the source image. Set to `NIL` to use the entire image.

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

Discussion
This function provides a way to use only a portion of the source image.

Version Notes
Introduced in QuickTime 3 or earlier.

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

Related Sample Code
CarbonQTGraphicImport

DropDraw

Declared In
`ImageCompression.h`

GraphicsImportValidate

Validates image data for a data reference to an imported image.

```
ComponentResult GraphicsImportValidate (
   GraphicsImportComponent ci,
   Boolean *valid
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

valid

Pointer to a Boolean value. On return, this parameter is set to TRUE if the the graphics importer component can draw the data reference. If the graphics importer component cannot draw the data reference, this parameter is set to FALSE.

Return Value

See `Error Codes`. Returns `noErr` if there is no error. Not all graphics importer components implement this function. A component that does not implement the function will return the `badComponentSelector` result code. This does not indicate that the file is valid or invalid.

Discussion

This function allows a graphics importer component to determine if its current data reference contains valid image data. For example, a JFIF graphics importer component might check for the presence of a JFIF marker at the start of the data stream. This function is provided for applications to use to determine what type of image data a particular file may contain. Sometimes a file may not have the correct file type or file extension. In this case, the application will not know which graphics importer component to use. By iterating through all graphics importer components and calling `GraphicsImportValidate` for each one, it may be possible to locate a graphics importer component that can draw the specified file.

Special Considerations

`GraphicsImportValidate` does not perform an exhaustive check on the file. It is possible for `GraphicsImportValidate` to claim a data reference is valid but for `GraphicsImportDraw` (page 74) to return an error due to bad data. Format-specific importers that implement the `GraphicsImportValidate` call should have the `canMovieImportValidateFile` bit set in the `flags` field of their `ComponentDescription` structures.

Version Notes

Introduced in QuickTime 3 or earlier.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`ImageCompression.h`

GraphicsImportWillUseColorMatching

Asks whether GraphicsImportDraw will use color matching if called with the current importer settings.

```
ComponentResult GraphicsImportWillUseColorMatching (
   GraphicsImportComponent ci,
   Boolean *outWillMatch
);
```

Parameters

ci

The component instance that identifies your connection to the graphics importer component.

outWillMatch

On return, a pointer to a Boolean set to TRUE if the graphics importer will use color matching, FALSE otherwise.

Return Value

See `Error Codes` in the QuickTime API Reference. Returns `noErr` if there is no error.

Version Notes

Introduced in QuickTime 6.4.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`ImageCompression.h`

Callbacks

ModalFilterYDProc

Determines how the Dialog Manager filters events.

```
typedef Boolean (*ModalFilterYDProcPtr) (DialogPtr theDialog, EventRecord *theEvent,
 short *itemHit, void *yourDataPtr);
```

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

```
Boolean MyModalFilterYDProc (
    DialogPtr      theDialog,
    EventRecord    *theEvent,
    short          *itemHit,
    void           *yourDataPtr );
```

Parameters

theDialog

A pointer to the dialog record.

theEvent

A pointer to the event record.

itemHit

The item number.

yourDataPtr

A pointer to the data received from your application, if any.

Return Value

Your `ModalFilterProc` callback returns a Boolean value that reports whether it handled the event. If your function returns a value of FALSE, QuickTime processes the event through its own filters. If your function returns a value of TRUE, QuickTime returns with no further action.

Discussion

The `ModalFilterProc` callback used with custom file dialogs requires the additional `yourDataPtr` parameter.

Declared In

`ImageCompression.h`

Data Types

GraphicsExportComponent

Represents a type used by the Graphics Import and Export API.

```
typedef ComponentInstance GraphicsExportComponent;
```

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

Declared In
`ImageCompression.h`

GraphicsImportComponent

Represents a type used by the Graphics Import and Export API.

```
typedef ComponentInstance GraphicsImportComponent;
```

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

Declared In
`ImageCompression.h`

ModalFilterYDUPP

Represents a type used by the Graphics Import and Export API.

```
typedef STACK_UPP_TYPE(ModalFilterYDProcPtr) ModalFilterYDUPP;
```

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

Declared In
`Dialogs.h`

Constants

GraphicsImportDoesDrawAllPixels Values

Constants passed to GraphicsImportDoesDrawAllPixels.

```
enum {
  graphicsImporterDrawsAllPixels = 0,
  graphicsImporterDoesntDrawAllPixels = 1,
  graphicsImporterDontKnowIfDrawAllPixels = 2
};
```

Declared In
`ImageCompression.h`

Graphics Importer Flags

Constants that represent the flags of graphics importers.

```
enum {
  graphicsImporterIsBaseImporter = 1L << 0,
  graphicsImporterCanValidateFile = 1L << 9,
  graphicsImporterSubTypeIsFileExtension = 1L << 12,
  graphicsImporterHasMIMEList   = 1L << 14,
  graphicsImporterUsesImageDecompressor = 1L << 23
};
enum {
  kGraphicsImporterDontDoGammaCorrection = 1L << 0,
  kGraphicsImporterTrustResolutionFromFile = 1L << 1,
  kGraphicsImporterEnableSubPixelPositioning = 1L << 2,
 kGraphicsImporterDontUseColorMatching = 1L << 3 /* set this flag (*before* calling
 GraphicsImportGetColorSyncProfile) if you do matching yourself */
};
```

Declared In
`ImageCompression.h`

GraphicsImportCreateCGImage Values

Constants passed to GraphicsImportCreateCGImage.

```
enum {
  kGraphicsImportCreateCGImageUsingCurrentSettings = 1L << 0
};
```

Declared In
`ImageCompression.h`

PNG Properties

Constants that represent the properties of PNGs.

```
enum {
  kQTPNGFilterPreference        = 'pngf', /* UInt32*/
  kQTPNGFilterBestForColorType  = 'bflt',
  kQTPNGFilterNone              = 0,
  kQTPNGFilterSub               = 1,
  kQTPNGFilterUp                = 2,
  kQTPNGFilterAverage           = 3,
  kQTPNGFilterPaeth             = 4,
  kQTPNGFilterAdaptivePerRow    = 'aflt',
  kQTPNGInterlaceStyle          = 'ilac', /* UInt32*/
  kQTPNGInterlaceNone           = 0,
  kQTPNGInterlaceAdam7          = 1
};
```

Constants

`kQTPNGFilterPreference`

UInt32.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`kQTPNGInterlaceStyle`

UInt32.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

Declared In

`ImageCompression.h`

TIFF Properties

Constants that represent the properties of TIFFs.

```
enum {
  kQTTIFFCompressionMethod      = 'tifc', /* UInt32*/
  kQTTIFFCompression_None       = 1,
  kQTTIFFCompression_PackBits   = 32773L,
  kQTTIFFLittleEndian           = 'tife' /* UInt8 (boolean)*/
};
enum {
  kQTTIFFUserDataModelPixelScale = 0x7469830E, /* 3 DOUBLEs */
  kQTTIFFUserDataModelTransformation = 0x746985D8, /* 16 DOUBLEs */
  kQTTIFFUserDataModelTiepoint  = 0x74698482, /* n DOUBLEs */
  kQTTIFFUserDataGeoKeyDirectory = 0x746987AF, /* n SHORTs */
  kQTTIFFUserDataGeoDoubleParams = 0x746987B0, /* n DOUBLEs */
  kQTTIFFUserDataGeoAsciiParams = 0x746987B1, /* n ASCIIs */
  kQTTIFFUserDataIntergraphMatrix = 0x74698480 /* 16 or 17 DOUBLEs */
};
enum {
  kQTTIFFUserDataOrientation    = 0x74690112, /* 1 SHORT */
  kQTTIFFUserDataTransferFunction = 0x7469012D, /* n SHORTs */
  kQTTIFFUserDataWhitePoint     = 0x7469013E, /* 2 RATIONALs */
  kQTTIFFUserDataPrimaryChromaticities = 0x7469013F, /* 6 RATIONALs */
  kQTTIFFUserDataTransferRange  = 0x74690156, /* 6 SHORTs */
  kQTTIFFUserDataYCbCrPositioning = 0x74690213, /* 1 SHORT */
  kQTTIFFUserDataReferenceBlackWhite = 0x74690214 /* n LONGs */
};
```

Constants

`kQTTIFFCompressionMethod`

UInt32.

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`kQTTIFFCompression_PackBits`

PackBits compression. This value is 32773L

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`kQTTIFFLittleEndian`

UInt8 (Boolean).

Available in Mac OS X v10.0 and later.

Declared in `ImageCompression.h`.

`kQTTIFFUserDataModelPixelScale`

3 DOUBLEs.

Available in Mac OS X v10.2 and later.

Declared in `ImageCompression.h`.

`kQTTIFFUserDataModelTransformation`

16 DOUBLEs.

Available in Mac OS X v10.2 and later.

Declared in `ImageCompression.h`.

`kQTTIFFUserDataModelTiepoint`

N DOUBLEs.

Available in Mac OS X v10.2 and later.

Declared in `ImageCompression.h`.

`kQTTIFFUserDataGeoKeyDirectory`

N SHORTs.

Available in Mac OS X v10.2 and later.

Declared in `ImageCompression.h`.

`kQTTIFFUserDataGeoDoubleParams`

N DOUBLEs.

Available in Mac OS X v10.2 and later.

Declared in `ImageCompression.h`.

`kQTTIFFUserDataGeoAsciiParams`

N ASCIIs.

Available in Mac OS X v10.2 and later.

Declared in `ImageCompression.h`.

`kQTTIFFUserDataIntergraphMatrix`

16 or 17 DOUBLEs.

Available in Mac OS X v10.2 and later.

Declared in `ImageCompression.h`.

`kQTTIFFUserDataOrientation`

1 SHORT.

Available in Mac OS X v10.2 and later.

Declared in `ImageCompression.h`.

`kQTTIFFUserDataTransferFunction`

N SHORTs.

Available in Mac OS X v10.2 and later.

Declared in `ImageCompression.h`.

`kQTTIFFUserDataWhitePoint`

2 RATIONALs.

Available in Mac OS X v10.2 and later.

Declared in `ImageCompression.h`.

`kQTTIFFUserDataPrimaryChromaticities`

6 RATIONALs.

Available in Mac OS X v10.2 and later.

Declared in `ImageCompression.h`.

`kQTTIFFUserDataTransferRange`

6 SHORTs.

Available in Mac OS X v10.2 and later.

Declared in `ImageCompression.h`.

`kQTTIFFUserDataYCbCrPositioning`

1 SHORT.

Available in Mac OS X v10.2 and later.

Declared in `ImageCompression.h`.

Declared In

`ImageCompression.h`

Document Revision History

This table describes the changes to *Import and Export Reference for QuickTime*.

Date	Notes
2006-05-23	New document, based on previously published material, that describes the API for QuickTime graphics importing and exporting.

Index