QLPreviewRequest Reference

User Experience > Files & Software Installation

2007-04-20

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

Simultaneously published in the United States and Canada.

Contents

QLPreviewRequest Reference 5

Overview 5
Functions by Task 6
- Assigning content to the preview request 6
- Getting attributes of the preview request 6
- Getting the QLPreviewRequest Type ID 6

Functions 6
- QLPreviewRequestCopyContentUTI 6
- QLPreviewRequestCopyOptions 7
- QLPreviewRequestCopyURL 8
- QLPreviewRequestCreateContext 8
- QLPreviewRequestCreatePDFContext 9
- QLPreviewRequestFlushContext 10
- QLPreviewRequestGetGeneratorBundle 10
- QLPreviewRequestGetTypeID 11
- QLPreviewRequestIsCancelled 11
- QLPreviewRequestSetDataRepresentation 12

Callbacks 13
- CancelPreviewGeneration 13
- GeneratePreviewForURL 14

Data Types 15
- QLPreviewRequestRef 15

Constants 15
- Preview Properties 15

Document Revision History 19

Index 21

QLPreviewRequest Reference

Derived From:	CFType
Framework:	QuickLook/QuickLook.h
Companion guide	Quick Look Programming Guide
Declared in	QLGenerator.h

Overview

Quick Look generators use the `QLPreviewRequest` API to help create a document preview that is returned to Quick Look clients (such as Finder and Spotlight) for display. An object of the `QLPreviewRequest` opaque type represents a request from a client for a preview of a document and is used to return the preview image the generator creates or locates

A generator must implement the `GeneratePreviewForURL` (page 14) callback function to create and return a preview image requested for a given document. `QLPreviewRequest` gives you four ways to create or locate a preview for a document:

- You can create a graphics context to draw the preview in. Graphics contexts for previews are of three kinds: bitmap, single-page vector, and multi-page vector.
  - For bitmap and single-page vector contexts, use the `QLPreviewRequestCreateContext` (page 8) function.
  - For multi-page (PDF) contexts, use the `QLPreviewRequestCreatePDFContext` (page 9) function.
- The generator can extract the preview data from the document and return it to the client using the `QLPreviewRequestSetDataRepresentation` (page 12) function. (For this approach to work the application's document must save the preview as part of the document.) The generator can also dynamically generate the preview and return it with this function.
- As a refinement of this same approach, the generator can dynamically generate the preview as HTML data that references attachments. It can then call `QLPreviewRequestSetDataRepresentation`, passing in a content type of HTML and dictionaries specifying the attachments.

Most of the other functions of `QLPreviewRequest` let you get the data associated with the preview request, such as the URL locating the document and the UTI identifying the content type of the document.

Functions by Task

Assigning content to the preview request

`QLPreviewRequestCreateContext` (page 8)
Creates a graphics context to draw the preview in.

`QLPreviewRequestCreatePDFContext` (page 9)
Creates a PDF context suitable to draw a multi-page preview.

`QLPreviewRequestSetDataRepresentation` (page 12)
Sets the preview request to data saved within the document or to dynamically generated data.

`QLPreviewRequestFlushContext` (page 10)
Flush the context and sets the preview response.

Getting attributes of the preview request

`QLPreviewRequestCopyContentUTI` (page 6)
Returns the UTI for the preview request.

`QLPreviewRequestCopyOptions` (page 7)
Returns the options specified for the preview request.

`QLPreviewRequestCopyURL` (page 8)
Returns the URL of the document for which a preview is requested.

`QLPreviewRequestGetGeneratorBundle` (page 10)
Get the bundle of the generator receiving the preview request.

`QLPreviewRequestIsCancelled` (page 11)
Returns whether the preview request has been cancelled by the client.

Getting the QLPreviewRequest Type ID

`QLPreviewRequestGetTypeID` (page 11)
Gets the type identifier for the `QLPreviewReqest` opaque type.

Functions

QLPreviewRequestCopyContentUTI

Returns the UTI for the preview request.

```
QL_EXPORT CFStringRef QLPreviewRequestCopyContentUTI(
    QLPreviewRequestRef preview
);
```

Parameters

preview

The preview request object.

Return Value

The UTI of the associated preview request; returns `NULL` if the UTI is not available. You should explicitly release this object when it is no longer needed.

Special Considerations

Thread-safety: This function should be called in the same thread as the preview request is made in; generally, this is the same thread in which the `GeneratePreviewForURL` (page 14) callback was invoked.

Availability

Available in Mac OS X version 10.5 and later.

Declared In

`QLGenerator.h`

QLPreviewRequestCopyOptions

Returns the options specified for the preview request.

```
QL_EXPORT CFDictionaryRef QLPreviewRequestCopyOptions(
    QLPreviewRequestRef preview
);
```

Parameters

preview

A preview request object.

Return Value

A dictionary containing the properties specified for the preview request. See "Preview Properties" (page 15) for supported options. You should explicitly release this object when it is no longer needed.

Discussion

The client sets options in the preview request to give hints to the generator. (For Mac OS X v10.5 no options are supported.)

Special Considerations

Thread-safety: This function should be called in the same thread as the preview request is made in; generally, this is the same thread in which the `GeneratePreviewForURL` (page 14) callback was invoked.

Availability

Available in Mac OS X version 10.5 and later.

Declared In

`QLGenerator.h`

QLPreviewRequestCopyURL

Returns the URL of the document for which a preview is requested.

```
QL_EXPORT CFURLRef QLPreviewRequestCopyURL(
    QLPreviewRequestRef preview
);
```

Parameters

preview

The preview request object.

Return Value

The URL identifying the file for which the preview is requested.

Special Considerations

This function is thread safe.

Availability

Available in Mac OS X version 10.5 and later.

Declared In

`QLGenerator.h`

QLPreviewRequestCreateContext

Creates a graphics context to draw the preview in.

```
QL_EXPORT CGContextRef QLPreviewRequestCreateContext(
    QLPreviewRequestRef preview,
    CGSize size,
    Boolean isBitmap,
    CFDictionaryRef properties
);
```

Parameters

preview

The preview request object.

size

The size of the preview; if *isBitmap* is `true` the size is in pixels, otherwise it is in points.

isBitmap

`true` if the preview uses a bitmap-based graphics context, `false` otherwise. This value of this parameter affects the interpretation of the *size* parameter.

properties

A dictionary containing properties for the preview response. "Preview Properties" (page 15) lists the current property keys and describes their values.

Return Value

A Core Graphics graphics-context object that you can draw your preview image in. You should explicitly release this object when it is no longer needed.

Discussion
You can directly draw your preview data in the graphics-context object created by this function. After calling this function, you should flush the context with `QLPreviewRequestFlushContext` (page 10). Also be sure to release the `CGContext` object.

Quick Look provides three types of graphics contexts for drawing previews: bitmap, single-page vector-based, and multi-page vector-based (for PDF previews). You use this function to acquire a context for bitmap and single-page vector drawing; the *isBitmap* parameter is used to distinguish between them. For multi-page contexts, use the `QLPreviewRequestCreatePDFContext` (page 9) function.

If you prefer to work in Objective-C code, you can convert the created `CGContextRef` to a `NSGraphicsContext` object using `graphicsContextWithGraphicsPort:flipped:`.

Special Considerations

Thread-safety: This function should be called in the same thread as the preview request is made in; generally, this is the same thread in which the `GeneratePreviewForURL` (page 14) callback was invoked.

Availability
Available in Mac OS X version 10.5 and later.

Declared In
`QLGenerator.h`

QLPreviewRequestCreatePDFContext

Creates a PDF context suitable to draw a multi-page preview.

```
QL_EXPORT CGContextRef QLPreviewRequestCreatePDFContext(
    QLPreviewRequestRef preview,
    const CGRect * mediaBox,
    CFDictionaryRef auxiliaryInfo,
    CFDictionaryRef properties);
);
```

Parameters

preview
The preview request object.

mediaBox
A pointer to the media box of the context.

A media box is a rectangle that defines the size and location of the PDF page. The origin of the rectangle should typically be (0,0). If you pass NULL, Quartz uses a default page size of 8.5 by 11 inches (612 by 792 points). For information see the description for `CGPDFContextCreate`.

auxiliaryInfo
A dictionary containing PDF auxiliary information. See the description of the auxiliary dictionary keys in *CGPDFContext Reference* for more information about the keys and values of this dictionary.

properties
A dictionary containing additional properties for the preview response. For information on acceptable keys and values, see "Preview Properties" (page 15).

Return Value
A reference to a Core Graphics context object that is used to display a PDF version of the preview. You should explicitly release this object when it is no longer needed.

Discussion
Be sure to bracket each PDF page written to the context with `CGPDFContextBeginPage` and `CGPDFContextEndPage` calls. After calling this function, you should flush the context with `QLPreviewRequestFlushContext` (page 10).

Special Considerations

Thread-safety: This function should be called in the same thread as the preview request is made in; generally, this is the same thread in which the `GeneratePreviewForURL` (page 14) callback was invoked.

Availability
Available in Mac OS X version 10.5 and later.

Declared In
`QLGenerator.h`

QLPreviewRequestFlushContext

Flush the context and sets the preview response.

```
QL_EXPORT void QLPreviewRequestFlushContext(
    QLPreviewRequestRef preview,
    CGContextRef context
);
```

Parameters

preview
The preview request object.

context
The graphics context to flush.

Discussion
You should call this method immediately after drawing in the graphics contexts created by `QLPreviewRequestCreateContext` (page 8) and `QLPreviewRequestCreatePDFContext` (page 9).

Special Considerations

Thread-safety: This function should be called in the same thread as the preview request is made in; generally, this is the same thread in which the `GeneratePreviewForURL` (page 14) callback was invoked.

Availability
Available in Mac OS X version 10.5 and later.

Declared In
`QLGenerator.h`

QLPreviewRequestGetGeneratorBundle

Get the bundle of the generator receiving the preview request.

```
QL_EXPORT CFBundleRef QLPreviewRequestGetGeneratorBundle(
    QLPreviewRequestRef preview
);
```

Parameters

preview

The preview request object.

Return Value

A reference to a bundle object representing the generator's bundle.

Special Considerations

Thread-safety: This function should be called in the same thread as the preview request is made in; generally, this is the same thread in which the `GeneratePreviewForURL` (page 14) callback was invoked.

Availability

Available in Mac OS X version 10.5 and later.

Declared In

`QLGenerator.h`

QLPreviewRequestGetTypeID

Gets the type identifier for the `QLPreviewReqest` opaque type.

```
CFTypeID QLPreviewRequestGetTypeID();
```

Return Value

The type identifier for the `QLPreviewReqest` opaque type.

Special Considerations

Thread-safety: This function should be called in the same thread as the preview request is made in; generally, this is the same thread in which the `GeneratePreviewForURL` (page 14) callback was invoked.

Availability

Available in Mac OS X version 10.5 and later.

Declared In

`QLGenerator.h`

QLPreviewRequestIsCancelled

Returns whether the preview request has been cancelled by the client.

```
QL_EXPORT Boolean QLPreviewRequestIsCancelled(
    QLPreviewRequestRef preview
);
```

Parameters

preview

The object representing the preview request.

Return Value

`true` if the request is being canceled, `false` otherwise.

Discussion
While computing the response, the generator can poll the preview request object with this function to determine if the client has cancelled the request. Alternatively, the generator can implement the `CancelPreviewGeneration` (page 13) callback. but since that function is called in a secondary thread, it is generally safer to take the polling approach.

Special Considerations

Thread-safety: This function should be called in the same thread as the preview request is made in; generally, this is the same thread in which the `GeneratePreviewForURL` (page 14) callback was invoked.

Availability
Available in Mac OS X version 10.5 and later.

Declared In
`QLGenerator.h`

QLPreviewRequestSetDataRepresentation

Sets the preview request to data saved within the document or to dynamically generated data.

```
QL_EXPORT void QLPreviewRequestSetDataRepresentation(
    QLPreviewRequest preview,
    CFDataRef data,
    CFStringRef contentTypeUTI,
    CFDictionary properties
);
```

Parameters

preview
The preview request object.

data
The data of the preview returned to the client. .

contentTypeUTI
The UTI specifying the content type of the preview.

properties
Additional properties for the preview response. For more on supported keys and values for this dictionary, see "Preview Properties" (page 15). If the saved data is HTML, you may specify a special set of properties; see the discussion below for more information.

Discussion
This function returns preview data to the client. The data is either extracted from a document (where the document's application has saved it,) or it is dynamically generated. How Quick Look handles the data depends upon the value of *contentTypeUTI*. The content data of the preview must be of a native Quick Look type. Currently supported UTIs for these types are: `kUTTypeImage`, `kUTTypePDF`, `kUTTypeHTML`, `kUTTypeXML`, `kUTTypePlainText`, `kUTTypeRTF`, `kUTTypeMovie`, and `kUTTypeAudio`.

If the UTI type is `kUTTypeHTML`, you can have the WebKit handle the layout and display of your preview. You must provide the HTML in *data* plus any attachments (for example, Address Book cards, Mail messages, or Omni Outliner documents) in the *properties* dictionary. This dictionary takes `kQLPreviewPropertyAttachmentsKey` (page 16) as its key and consists of one ore more subdictionaries (one per attachment). Each subdictionary uses an arbitrary string identifier as a key; the attachment should be referenced within the HTML data using the `kQLPreviewContentIDScheme` URL scheme ("cid") and the identifier as the URL resource specifier—for example, "cid:the_identifier". The keys of the subdictionary

properties are `kQLPreviewPropertyMIMETypeKey` (page 16), `kQLPreviewPropertyTextEncodingNameKey` (page 16), and `kQLPreviewPropertyAttachmentDataKey` (page 16).

Special Considerations

Thread-safety: This function should be called in the same thread as the preview request is made in; generally, this is the same thread in which the `GeneratePreviewForURL` (page 14) callback was invoked.

Availability
Available in Mac OS X version 10.5 and later.

Declared In
`QLGenerator.h`

Callbacks

CancelPreviewGeneration

Defines a pointer to a function that Quick Look calls to cancel a preview request.

```
void (*CancelPreviewGeneration)(
    void *thisInterface,
    QLPreviewRequestRef preview
);
```

Parameters

thisInterface
The `CFPlugIn` COM-style interface for the generator.

preview
The object representing the preview request.

Discussion
If the client application indicates (usually because of user choice) that it is no longer interested in the preview currently being created by a Quick Look generator, this callback function is invoked. The generator can implement this function to stop creating the preview and clean up the resources used for creating it.

An alternative to implementing this callback is to periodically poll the `QLPreviewRequestRef` (page 15) object with `QLPreviewRequestIsCancelled` (page 11) to see if the request has been canceled.

Important: Because this function is called on a thread different from the one used to request that the preview be created, you should be extra careful about thread safety. If you have any doubts about thread safety, do not implement this callback.

Special Considerations

Thread-safety: For a discussions of issues and possible approaches, see Overview of Generator Implementation in *Quick Look Programming Guide*.

Availability
Available in Mac OS X version 10.5 and later.

Declared In
`QuickLook/QLGenerator.h`

GeneratePreviewForURL

Defines a pointer to the callback function that Quick Look calls to request a preview from a generator.

```
OSStatus (*GeneratePreviewForURL)(
    void *thisInterface,
    QLPreviewRequestRef preview,
    CFURLRef url,
    CFStringRef contentTypeUTI,
    CFDictionaryRef options
);
```

The Xcode template for Quick Look generators automatically creates a "skeletal" function with the same name as the callback symbol name: `GeneratePreviewForURL`.

Parameters

`thisInterface`
The `CFPlugIn` COM-style interface for the generator.

`preview`
The object containing all information relevant to the preview request. The generator's role is to assign a document preview (in the requested native type) to this object before it returns.

`url`
A URL (represented by a `CFURLRef` object) that locates the document for which a preview is requested.

`contentTypeUTI`
The UTI specifying the content type of the document for which the preview is requested.

`options`
A dictionary of options for processing the preview. Options may be passed from the client (for example, Finder or Spotlight).

Return Value
A status code representing the result of the request. For the current version of this callback, you should always return `noErr`.

Discussion
The `GeneratePreviewForURL` callback function implemented by a Quick Look generator may be called one or more times concurrenly if the `QLSupportsConcurrentRequests` property in the generator bundle's `Info.plist` file is set to `true`. If this is the case, use the *`preview`* parameter to distinguish among current requests. The generator might also have the `QLNeedsToBeRunInMainThread` property set to `true`, in which case the callback is always invoked in the main thread.

Special Considerations

Thread-safety: For a discussions of issues and possible approaches, see "Overview of Generator Implementation" in *Quick Look Programming Guide*.

Availability
Available in Mac OS X version 10.5 and later.

Declared In
`QuickLook/QLGenerator.h`

Data Types

QLPreviewRequestRef

An opaque reference representing an `QLPreviewRequest` object.

```
typedef struct __QLPreviewRequest *QLPreviewRequestRef;
```

Availability
Available in Mac OS X version 10.5 and later.

Declared In
`QLGenerator.h`

Constants

Preview Properties

Keys of the properties dictionary in functions used to create a preview.

```
QL_EXPORT const CFStringRef kQLPreviewPropertyDisplayNameKey;
QL_EXPORT const CFStringRef kQLPreviewPropertyWidthKey;
QL_EXPORT const CFStringRef kQLPreviewPropertyHeightKey;
QL_EXPORT const CFStringRef kQLPreviewPropertyStringEncodingKey;
QL_EXPORT const CFStringRef kQLPreviewPropertyMIMETypeKey;
QL_EXPORT const CFStringRef kQLPreviewPropertyTextEncodingNameKey;
QL_EXPORT const CFStringRef kQLPreviewPropertyAttachmentDataKey;
QL_EXPORT const CFStringRef kQLPreviewPropertyAttachmentsKey;
QL_EXPORT const CFStringRef kQLPreviewContentIDScheme;
```

Constants

`kQLPreviewPropertyDisplayNameKey`

Specifies a custom display name in the preview panel. The default display name is the document title. The value must be encapsulated in a `CFString` object.

Available in Mac OS X v10.5 and later.

Declared in `QLGenerator.h`.

`kQLPreviewPropertyWidthKey`

Specifies the width (in points) of the preview. Note that this property is a hint; Quick Look might set the width automatically for some types of previews. The value must be encapsulated in a `CFNumber` object.

Available in Mac OS X v10.5 and later.

Declared in `QLGenerator.h`.

`kQLPreviewPropertyHeightKey`
Specifies the height (in points) of the preview. Note that this property is a hint; Quick Look might set the height automatically for some types of previews. The value must be encapsulated in a `CFNumber` object.

Available in Mac OS X v10.5 and later.

Declared in `QLGenerator.h`.

`kQLPreviewPropertyStringEncodingKey`
Specifies the string encoding (as an `CFStringEncoding` value) of the preview data if the native type is plain text. The value must be encapsulated in a `CFNumber` object.

Available in Mac OS X v10.5 and later.

Declared in `QLGenerator.h`.

`kQLPreviewPropertyMIMETypeKey`
Gives the web content or attachment mime type. For the main data, the default type is "text/html". The value is a `CFString` object.

Available in Mac OS X v10.5 and later.

Declared in `QLGenerator.h`.

`kQLPreviewPropertyTextEncodingNameKey`
Specifies the encoding of the web content or attachment text. For the value, use IANA encodings like "UTF-8". The value value is a `CFString` object.

Available in Mac OS X v10.5 and later.

Declared in `QLGenerator.h`.

`kQLPreviewPropertyAttachmentDataKey`
Gives the attachment data. The value is a `CFData` object.

Available in Mac OS X v10.5 and later.

Declared in `QLGenerator.h`.

`kQLPreviewPropertyAttachmentsKey`
Gives the list of attachments (or sub-resources). Value is a `CFDictionary` object.

The keys of the dictionary are the attachment identifiers (`CFString` objects) that can be referenced with the `cid:id` URL; the values are dictionaries using the `kQLPreviewPropertyAttachmentDataKey`, `kQLPreviewPropertyMIMETypeKey` and `kQLPreviewPropertyTextEncodingNameKey` properties.

Available in Mac OS X v10.5 and later.

Declared in `QLGenerator.h`.

`kQLPreviewContentIDScheme`
The `cid` URL scheme.

The `cid` URL scheme permits the HTML of a mail message to refer to the images or other data included in the message. For more information go to http://www.rfc-archive.org/getrfc.php?rfc=2111.

Available in Mac OS X v10.5 and later.

Declared in `QLGenerator.h`.

Discussion
You use these keys to specify properties of the generated preview in the dictionary passed into the functions `QLPreviewRequestCreateContext` (page 8), `QLPreviewRequestCreatePDFContext` (page 9), and `QLPreviewRequestSetDataRepresentation` (page 12) as the final parameter.

Availability
Available in Mac OS X version 10.5 and later.

Declared In
`QuickLook/QLGenerator.h`

Document Revision History

This table describes the changes to *QLPreviewRequest Reference*.

Date	Notes
2007-04-20	New document that describes the API related to the QLPreviewRequest opaque type of the Quick Look framework.

Document Revision History

Index

C

CancelPreviewGeneration callback 13

G

GeneratePreviewForURL callback 14

K

kQLPreviewContentIDScheme constant 16
kQLPreviewPropertyAttachmentDataKey constant 16
kQLPreviewPropertyAttachmentsKey constant 16
kQLPreviewPropertyDisplayNameKey constant 15
kQLPreviewPropertyHeightKey constant 16
kQLPreviewPropertyMIMETypeKey constant 16
kQLPreviewPropertyStringEncodingKey constant 16
kQLPreviewPropertyTextEncodingNameKey constant 16
kQLPreviewPropertyWidthKey constant 15

P

Preview Properties 15

Q

QLPreviewRequestCopyContentUTI function 6
QLPreviewRequestCopyOptions function 7
QLPreviewRequestCopyURL function 8
QLPreviewRequestCreateContext function 8
QLPreviewRequestCreatePDFContext function 9
QLPreviewRequestFlushContext function 10
QLPreviewRequestGetGeneratorBundle function 10
QLPreviewRequestGetTypeID function 11
QLPreviewRequestIsCancelled function 11
QLPreviewRequestRef structure 15
QLPreviewRequestSetDataRepresentation function 12