

---

# QLThumbnailRequest Reference

Data Management: File Management



2007-10-31



Apple Inc.  
© 2007 Apple Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

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

Apple, the Apple logo, Mac, Mac OS, Objective-C, Quartz, and Xcode are trademarks of Apple Inc., registered in the United States and other countries.

Finder and Spotlight are trademarks of Apple Inc.

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

---

## **QLThumbnailRequest Reference 5**

---

Overview	5
Functions by Task	6
Assigning content to the thumbnail request	6
Getting attributes of the thumbnail request	6
Getting the QLThumbnailRequest Type ID	6
Client Functions	6
Functions	7
QLThumbnailImageCreate	7
QLThumbnailRequestCopyContentUTI	7
QLThumbnailRequestCopyOptions	8
QLThumbnailRequestCopyURL	9
QLThumbnailRequestCreateContext	9
QLThumbnailRequestFlushContext	10
QLThumbnailRequestGetGeneratorBundle	11
QLThumbnailRequestGetMaximumSize	11
QLThumbnailRequestGetTypeID	12
QLThumbnailRequestIsCancelled	12
QLThumbnailRequestSetImage	12
QLThumbnailRequestSetImageWithData	13
Callbacks	14
CancelThumbnailGeneration	14
GenerateThumbnailForURL	15
Data Types	16
QLThumbnailRequestRef	16
Constants	16
General Thumbnail Options	16
Client-Specific Thumbnail Options	16

---

## **Document Revision History 19**

---

## **Index 21**

---



# QLThumbnailRequest Reference

---

<b>Derived From:</b>	CType
<b>Framework:</b>	QuickLook/QuickLook.h
<b>Companion guide</b>	Quick Look Programming Guide
<b>Declared in</b>	QLGenerator.h QLThumbnailImage.h

## Overview

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

A generator must implement the [GenerateThumbnailForURL](#) (page 15) callback function to create and return a thumbnail image requested for a given document. `QLThumbnailRequest` gives you four ways to designate a thumbnail for a document:

- You can create a graphics context to draw the thumbnail in. Graphics contexts for thumbnails are of two kinds: bitmap and single-page vector. You use the [QLThumbnailRequestCreateContext](#) (page 9) for this purpose.
- You can extract the thumbnail from the document and return it using the [QLThumbnailRequestSetImageWithData](#) (page 13) function. For this approach, the document's application must save the thumbnail as part of the document data.
- You can create the thumbnail image and assign it to the thumbnail request using the [QLThumbnailRequestSetImage](#) (page 12) function.

Most of the other functions of `QLThumbnailRequest` let you get the data associated with the thumbnail request, such as the URL locating the document, the UTI identifying the content type of the document, and the maximum size allowed for the thumbnail image.

**Note:** This reference documentation also describes the use of the [QLThumbnailImageCreate](#) (page 7) function, which is called by *clients* of Quick Look, not by generator plug-ins. For example, an application could call this function to obtain a thumbnail image of a document that it is listing.

## Functions by Task

### Assigning content to the thumbnail request

- [QLThumbnailRequestCreateContext](#) (page 9)  
Creates a graphics context to draw the thumbnail in.
- [QLThumbnailRequestSetImage](#) (page 12)  
Sets the thumbnail request to a specified image.
- [QLThumbnailRequestSetImageWithData](#) (page 13)  
Sets the response to the thumbnail request to image data saved within the document.
- [QLThumbnailRequestFlushContext](#) (page 10)  
Flush the graphics context and sets the thumbnail response.

### Getting attributes of the thumbnail request

- [QLThumbnailRequestCopyContentUTI](#) (page 7)  
Returns the UTI for the thumbnail request.
- [QLThumbnailRequestCopyOptions](#) (page 8)  
Returns the options specified for the thumbnail request.
- [QLThumbnailRequestCopyURL](#) (page 9)  
Returns the URL of the document for which the thumbnail request is requested.
- [QLThumbnailRequestGetGeneratorBundle](#) (page 11)  
Get the bundle of the generator receiving the thumbnail request.
- [QLThumbnailRequestGetMaximumSize](#) (page 11)  
Returns the maximum size (in points) specified for the thumbnail image.
- [QLThumbnailRequestIsCancelled](#) (page 12)  
Returns whether the thumbnail request has been cancelled by the client.

### Getting the QLThumbnailRequest Type ID

- [QLThumbnailRequestGetTypeID](#) (page 12)  
Gets the type identifier for the `QLThumbnailRequest` opaque type.

### Client Functions

The following functions are for use by clients of Quick Look (generally applications), not by generator plug-ins.

[QLThumbnailImageCreate](#) (page 7)

Creates a thumbnail image for the specified file.

## Functions

### QLThumbnailImageCreate

Creates a thumbnail image for the specified file.

```
QL_EXPORT CGImageRef QLThumbnailImageCreate(CFAllocatorRef allocator, CFURLRef url,
      CGSize maxThumbnailSize, CFDictionaryRef options);
```

#### Parameters

*allocator*

The allocator to use to create the thumbnail image.

*url*

The URL locating the file to create a thumbnail image for.

*maxThumbnailSize*

The maximum desired size of the thumbnail image.

*options*

A dictionary of options that affect the creation of the thumbnail image. Two options are supported: [kQLThumbnailOptionIconModeKey](#) (page 17) and [kQLThumbnailOptionScaleFactorKey](#) (page 16).

#### Return Value

Returns the thumbnail image or NULL if Quick Look does not support this file type

#### Discussion

This function does not supplant the use of Icon Services by applications as a way to get generic file icons and custom icons stored in the metadata fork of files.

#### Special Considerations

**Thread-safety:** This function is thread-safe so you can call it from any thread. However, because it is synchronous, you should generally call it in a background thread.

#### Availability

Available in Mac OS X v10.5 and later.

#### Declared In

QLThumbnailImage.h

### QLThumbnailRequestCopyContentUTI

Returns the UTI for the thumbnail request.

```
QL_EXPORT CFStringRef QLThumbnailRequestCopyContentUTI(
    QLThumbnailRequestRef thumbnail
);
```

**Parameters***thumbnail*

The thumbnail request object.

**Return Value**

The UTI identifying the content type of the document for which the thumbnail is requested; returns `NULL` if the UTI cannot be located. You should explicitly release this string object when it is no longer needed.

**Special Considerations**

**Thread-safety:** This function should be called in the same thread as the thumbnail request is made in; generally, this is the same thread in which the [GenerateThumbnailForURL](#) (page 15) callback was invoked.

**Availability**

Available in Mac OS X version 10.5 and later.

**Declared In**

QLGenerator.h

**QLThumbnailRequestCopyOptions**

Returns the options specified for the thumbnail request.

```
QL_EXPORT CFDictionaryRef QLThumbnailRequestCopyOptions(
    QLThumbnailRequestRef thumbnail
);
```

**Parameters***thumbnail*

A thumbnail request object.

**Return Value**

A dictionary containing the options that the client specified for the thumbnail request. See [“General Thumbnail Options”](#) (page 16) for supported options. You should explicitly release the dictionary when it is no longer needed.

**Discussion**

The client sets options in the thumbnail request to give hints to the generator. (For Mac OS X v10.5 no options are supported.) You should explicitly release the dictionary when it is no longer needed.

**Special Considerations**

**Thread-safety:** This function should be called in the same thread as the thumbnail request is made in; generally, this is the same thread in which the [GenerateThumbnailForURL](#) (page 15) callback was invoked.

**Availability**

Available in Mac OS X version 10.5 and later.

**Declared In**

QLGenerator.h

## QLThumbnailRequestCopyURL

Returns the URL of the document for which the thumbnail request is requested.

```

QL_EXPORT CFURLRef QLThumbnailRequestCopyURL(
    QLThumbnailRequestRef thumbnail
);

```

### Parameters

*thumbnail*

The thumbnail request object.

### Return Value

The URL identifying the document for which the thumbnail is requested. You should explicitly release the CFURL object when it is no longer needed.

### Special Considerations

**Thread-safety:** This function should be called in the same thread as the thumbnail request is made in; generally, this is the same thread in which the [GenerateThumbnailForURL](#) (page 15) callback was invoked.

### Availability

Available in Mac OS X version 10.5 and later.

### Declared In

QLGenerator.h

## QLThumbnailRequestCreateContext

Creates a graphics context to draw the thumbnail in.

```

QL_EXPORT CGContextRef QLThumbnailRequestCreateContext(
    QLThumbnailRequestRef preview,
    CGSize size,
    Boolean isBitmap,
    CFDictionaryRef properties
);

```

### Parameters

*preview*

The thumbnail request object.

*size*

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

*isBitmap*

*true* if the thumbnail data is bitmap-based, *false* if vector-based. This value of this parameter affects the interpretation of the *size* parameter.

*properties*

A dictionary containing properties for the thumbnail response. (For Mac OS X v10.5 no properties have been defined.).

### Return Value

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

**Discussion**

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

With this function you can create two types of graphics contexts for drawing thumbnails: bitmap and single-page vector-based; you use the `isBitmap` flag to distinguish between the two. Quick Look handles bitmap thumbnail context differently than non-bitmap contexts; in the latter case, Quick Look might scale the drawing up or down if necessary, and it respects the scale factor (for HiDPI support).

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 thumbnail request is made in; generally, this is the same thread in which the [GenerateThumbnailForURL](#) (page 15) callback was invoked.

**Availability**

Available in Mac OS X version 10.5 and later.

**Related Sample Code**

QuickLookSketch

**Declared In**

QLGenerator.h

**QLThumbnailRequestFlushContext**

Flush the graphics context and sets the thumbnail response.

```
QL_EXPORT void QLThumbnailRequestFlushContext(
    QLThumbnailRequestRef preview,
    CGContextRef context
);
```

**Parameters**

*preview*

The preview request object.

*context*

The graphics context to flush.

**Discussion**

You should call this method after drawing in the graphics context created by [QLThumbnailRequestCreateContext](#) (page 9).

**Special Considerations**

**Thread-safety:** This function should be called in the same thread as the thumbnail request is made in; generally, this is the same thread in which the [GenerateThumbnailForURL](#) (page 15) callback was invoked.

**Availability**

Available in Mac OS X version 10.5 and later.

**Related Sample Code**

QuickLookSketch

**Declared In**

QLGenerator.h

**QLThumbnailRequestGetGeneratorBundle**

Get the bundle of the generator receiving the thumbnail request.

```
QL_EXPORT CFBundleRef QLThumbnailRequestGetGeneratorBundle(
    QLThumbnailRequestRef thumbnail
);
```

**Parameters***thumbnail*

The thumbnail request object.

**Return Value**

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

**Special Considerations**

**Thread-safety:** This function should be called in the same thread as the thumbnail request is made in; generally, this is the same thread in which the [GenerateThumbnailForURL](#) (page 15) callback was invoked.

**Availability**

Available in Mac OS X version 10.5 and later.

**Declared In**

QLGenerator.h

**QLThumbnailRequestGetMaximumSize**

Returns the maximum size (in points) specified for the thumbnail image.

```
QL_EXPORT CGSize QLThumbnailRequestGetMaximumSize(
    QLThumbnailRequestRef thumbnail
);
```

**Parameters***thumbnail*

The thumbnail request object.

**Return Value**

The maximum size (in points) specified for the thumbnail image.

**Special Considerations**

**Thread-safety:** This function should be called in the same thread as the thumbnail request is made in; generally, this is the same thread in which the [GenerateThumbnailForURL](#) (page 15) callback was invoked.

**Availability**

Available in Mac OS X version 10.5 and later.

**Declared In**

QLGenerator.h

## QLThumbnailRequestGetTypeID

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

```
QL_EXPORT CTypeID QLThumbnailRequestGetTypeID();
```

### Return Value

The type identifier for the `QLThumbnailRequest` opaque type.

### Special Considerations

**Thread-safety:** This function should be called in the same thread as the thumbnail request is made in; generally, this is the same thread in which the [GenerateThumbnailForURL](#) (page 15) callback was invoked.

### Availability

Available in Mac OS X version 10.5 and later.

### Declared In

`QLGenerator.h`

## QLThumbnailRequestIsCancelled

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

```
QL_EXPORT Boolean QLThumbnailRequestIsCancelled(
    QLThumbnailRequestRef thumbnail
);
```

### Parameters

*thumbnail*

The thumbnail request object.

### Return Value

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

### Special Considerations

**Thread-safety:** This function should be called in the same thread as the thumbnail request is made in; generally, this is the same thread in which the [GenerateThumbnailForURL](#) (page 15) callback was invoked.

### Availability

Available in Mac OS X v10.5 and later.

### Declared In

`QLGenerator.h`

## QLThumbnailRequestSetImage

Sets the thumbnail request to a specified image.

```

QL_EXPORT void QLThumbnailRequestSetImage(
    QLThumbnailRequestRef thumbnail,
    CGImageRef image,
    CFDictionaryRef properties);
);

```

**Parameters***thumbnail*

The thumbnail request object.

*image*

The image object to be used as the thumbnail of the document.

*properties*

A dictionary of properties for the thumbnail. (For Mac OS X v10.5 no properties have been defined.)

**Discussion**

You call this function to have Quick Look use the `CGImage` object supplied in *image* as the thumbnail. Call [QLThumbnailRequestGetMaximumSize](#) (page 11) to get the maximum allowed size for thumbnail and resize it if necessary before calling `QLThumbnailSetImage`.

**Special Considerations**

**Thread-safety:** This function should be called in the same thread as the thumbnail request is made in; generally, this is the same thread in which the [GenerateThumbnailForURL](#) (page 15) callback was invoked.

**Availability**

Available in Mac OS X version 10.5 and later.

**Declared In**

QLGenerator.h

**QLThumbnailRequestSetImageWithData**

Sets the response to the thumbnail request to image data saved within the document.

```

QL_EXPORT void QLThumbnailRequestSetImageWithData(
    QLThumbnailRequestRef thumbnail,
    CFDataRef data,
    CFDictionaryRef properties);
);

```

**Parameters***thumbnail*

The thumbnail request object.

*data*

The image data, which must be in a format supported by the Image I/O framework (JPG, PNG, and so on). In other words, a content type of `kUTTypeImage` is assumed. (`ImageIO.framework` is a subframework of the umbrella Application Services framework.)

*properties*

A dictionary of properties. The only property that you can currently specify is `kCGImageSourceTypeIdentifierHint`; see *Quartz 2D Programming Guide* for information about this property.

**Discussion**

This function returns the thumbnail as a `CFData` object containing image data. The document's application must save this data as part of the document's data; the generator retrieves it and uses this function to return it to the client. Before you call this function, call [QLThumbnailRequestGetMaximumSize](#) (page 11) to obtain the maximum allowed size for the thumbnail and resize the image if necessary.

**Special Considerations**

**Thread-safety:** This function should be called in the same thread as the thumbnail request is made in; generally, this is the same thread in which the [GenerateThumbnailForURL](#) (page 15) callback was invoked.

**Availability**

Available in Mac OS X version 10.5 and later.

**Declared In**

`QLGenerator.h`

## Callbacks

**CancelThumbnailGeneration**

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

```
void (*CancelThumbnailGeneration)(
    void *thisInterface,
    QLThumbnailThumbnailRef thumbnail
);
```

**Parameters**

*thisInterface*

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

*thumbnail*

The object representing the thumbnail request.

**Discussion**

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

An alternative to implementing this callback is to periodically poll the [QLThumbnailRequestRef](#) (page 16) object with [QLThumbnailRequestIsCancelled](#) (page 12) 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 thumbnail be created, you should be extra careful about thread safety. If you have any doubts about thread safety, do not implement this callback.

**Availability**

Available in Mac OS X version 10.5 and later.

**Declared In**

`QuickLook/QLGenerator.h`

## GenerateThumbnailForURL

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

```
OSStatus (*GenerateThumbnailForURL)(
    void *thisInterface,
    QLThumbnailRequestRef thumbnail,
    CFURLRef url,
    CFStringRef contentTypeUTI,
    CFDictionaryRef options,
    CGSize maxSize
);
```

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

### Parameters

*thisInterface*

The CFPlugIn COM-style interface for the generator.

*thumbnail*

The object containing all information relevant to the thumbnail request. The generator’s role is to assign a document thumbnail (in one of the native Quick Look types) to this object before it returns.

*url*

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

*contentTypeUTI*

The UTI specifying the content type of the document for which the thumbnail is requested.

*options*

A dictionary of options passed from the client (for example, Finder or Spotlight) that are hints for processing the thumbnail.

*maxSize*

The maximum desired size of the thumbnail.

### 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 `GenerateThumbnailForURL` callback function implemented by a Quick Look generator may be called one or more times concurrently if the `QLSupportsConcurrentRequests` option in the generator bundle’s `Info.plist` file is set to `true`. If this is the case, use the *thumbnail* 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

### QLThumbnailRequestRef

An opaque reference representing an `QLThumbnailRequest` object.

```
typedef struct __QLThumbnailRequest *QLThumbnailRequestRef;
```

#### Availability

Available in Mac OS X version 10.5 and later.

#### Declared In

`QLGenerator.h`

## Constants

### General Thumbnail Options

Keys used for the options passed by the client to the generator and by the client when requesting a thumbnail image.

```
QL_EXPORT const CFStringRef kQLThumbnailOptionScaleFactorKey;
```

#### Constants

`kQLThumbnailOptionScaleFactorKey`

A key used to specify the user scale factor for the thumbnail. The `float` value is encapsulated in a `CFNumber` object. If this option is absent, the default value is 1.0

Available in Mac OS X v10.5 and later.

Declared in `QLThumbnailImage.h`.

#### Discussion

This constant is a key for accessing the scale-factor value in the dictionary passed into [GenerateThumbnailForURL](#) (page 15) in the `options` parameter. It can also be used for the [QLThumbnailImageCreate](#) (page 7) function, which is called by clients (generally applications) of Quick Look.

#### Availability

Available in Mac OS X version 10.5 and later.

#### Declared In

`QuickLook/QLThumbnailImage.h`

### Client-Specific Thumbnail Options

Clients of Quick Look can specify the following option-dictionary properties for special handling of thumbnail images.

```
QL_EXPORT const CFStringRef kQLThumbnailOptionIconModeKey;
```

### Constants

`kQLThumbnailOptionIconModeKey`

If the value is `kCFBooleanTrue`, Quick Look produces the thumbnail as an icon. The image includes all the typical icon decor, such as shadows and a curled corner. If `kCFBooleanFalse` (the default) is specified, no icon decor is included with the thumbnail image.

Available in Mac OS X v10.5 and later.

Declared in `QLThumbnailImage.h`.

### Declared In

`QuickLook/QLThumbnailImage.h`



# Document Revision History

---

This table describes the changes to *QLThumbnailRequest Reference*.

Date	Notes
2007-10-31	New document that describes the opaque type used by Quick Look generators to create thumbnail images for documents.

**REVISION HISTORY**

Document Revision History

# Index

---

## C

---

`CancelThumbnailGeneration` **callback** [14](#)  
`Client-Specific Thumbnail Options` [16](#)

## G

---

`General Thumbnail Options` [16](#)  
`GenerateThumbnailForURL` **callback** [15](#)

## K

---

`kQLThumbnailOptionIconModeKey` **constant** [17](#)  
`kQLThumbnailOptionScaleFactorKey` **constant** [16](#)

## Q

---

`QLThumbnailImageCreate` **function** [7](#)  
`QLThumbnailRequestCopyContentUTI` **function** [7](#)  
`QLThumbnailRequestCopyOptions` **function** [8](#)  
`QLThumbnailRequestCopyURL` **function** [9](#)  
`QLThumbnailRequestCreateContext` **function** [9](#)  
`QLThumbnailRequestFlushContext` **function** [10](#)  
`QLThumbnailRequestGetGeneratorBundle` **function** [11](#)  
`QLThumbnailRequestGetMaximumSize` **function** [11](#)  
`QLThumbnailRequestGetTypeID` **function** [12](#)  
`QLThumbnailRequestIsCancelled` **function** [12](#)  
`QLThumbnailRequestRef` **structure** [16](#)  
`QLThumbnailRequestSetImage` **function** [12](#)  
`QLThumbnailRequestSetImageWithData` **function** [13](#)