

---

# CGImageSource Reference

[Graphics & Imaging](#) > Quartz



2007-12-04



Apple Inc.  
© 2004, 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, Carbon, Cocoa, Mac, Mac OS, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

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

---

## **CGImageSource Reference 5**

---

Overview	5
Functions by Task	5
Creating an Image Source	5
Creating Images From an Image Source	5
Updating an Image Source	6
Getting Information From an Image Source	6
Functions	6
CGImageSourceCopyProperties	6
CGImageSourceCopyPropertiesAtIndex	7
CGImageSourceCopyTypeIdentifiers	7
CGImageSourceCreateImageAtIndex	8
CGImageSourceCreateIncremental	8
CGImageSourceCreateThumbnailAtIndex	9
CGImageSourceCreateWithData	10
CGImageSourceCreateWithDataProvider	10
CGImageSourceCreateWithURL	11
CGImageSourceGetCount	11
CGImageSourceGetStatus	12
CGImageSourceGetStatusAtIndex	12
CGImageSourceGetType	13
CGImageSourceGetTypeID	13
CGImageSourceUpdateData	14
CGImageSourceUpdateDataProvider	14
Data Types	15
CGImageSourceRef	15
Constants	15
Image Source Status	15
Image Source Option Dictionary Keys	16

---

## **Document Revision History 19**

---

## **Index 21**

---



# CGImageSource Reference

---

<b>Derived From:</b>	CType
<b>Framework:</b>	ApplicationServices/ImageIO
<b>Declared in</b>	CGImageSource.h
<b>Companion guides</b>	Quartz 2D Programming Guide CGImage Reference

## Overview

CGImageSource objects, available in Mac OS X v10.4 or later, abstract the data-reading task. An image source can read image data from a URL, a CFData object, or a data consumer.

After creating a CGImageSource object for the appropriate source, you can obtain images, thumbnails, image properties, and other image information using CGImageSource functions.

## Functions by Task

### Creating an Image Source

[CGImageSourceCreateWithDataProvider](#) (page 10)

Creates an image source that reads data from the specified data provider.

[CGImageSourceCreateWithData](#) (page 10)

Creates an image source that reads from a Core Foundation data object.

[CGImageSourceCreateWithURL](#) (page 11)

Creates an image source that reads from a location specified by a URL.

### Creating Images From an Image Source

[CGImageSourceCreateImageAtIndex](#) (page 8)

Creates a CGImage object for the image data associated with the specified index in an image source.

[CGImageSourceCreateThumbnailAtIndex](#) (page 9)

Creates a thumbnail image of the image located at a specified location in an image source.

[CGImageSourceCreateIncremental](#) (page 8)

Create an incremental image source.

## Updating an Image Source

[CGImageSourceUpdateData](#) (page 14)

Updates an incremental image source with new data.

[CGImageSourceUpdateDataProvider](#) (page 14)

Updates an incremental image source with a new data provider.

## Getting Information From an Image Source

[CGImageSourceGetTypeID](#) (page 13)

Returns the unique type identifier of an image source opaque type.

[CGImageSourceGetType](#) (page 13)

Returns the uniform type identifier of the source container.

[CGImageSourceCopyTypeIdentifiers](#) (page 7)

Returns an array of uniform type identifiers (UTIs) that are supported for image sources.

[CGImageSourceGetCount](#) (page 11)

Returns the number of images (not including thumbnails) in the image source.

[CGImageSourceCopyProperties](#) (page 6)

Returns the properties of the image source.

[CGImageSourceCopyPropertiesAtIndex](#) (page 7)

Returns the properties of the image at a specified location in an image source.

[CGImageSourceGetStatus](#) (page 12)

Return the status of an image source.

[CGImageSourceGetStatusAtIndex](#) (page 12)

Returns the current status of an image that is at a specified location in an image source.

## Functions

### CGImageSourceCopyProperties

Returns the properties of the image source.

```
CFDictionaryRef CGImageSourceCopyProperties (
    CGImageSourceRef isrc,
    CFDictionaryRef options
);
```

#### Parameters

*isrc*

An image source.

*options*

A dictionary you can use to request additional options. See “[Image Source Option Dictionary Keys](#)” (page 16) for the keys you can supply.

**Return Value**

A dictionary that contains the properties associated with the image source container. See *CGImageProperties Reference* for a list of properties that can be in the dictionary.

**Discussion**

These properties apply to the container in general but not necessarily to any individual image contained in the image source.

**Availability**

Available in Mac OS X version 10.4 and later.

**Declared In**

CGImageSource.h

**CGImageSourceCopyPropertiesAtIndex**

Returns the properties of the image at a specified location in an image source.

```
CFDictionaryRef CGImageSourceCopyPropertiesAtIndex (
    CGImageSourceRef isrc,
    size_t index,
    CFDictionaryRef options
);
```

**Parameters**

*isrc*

An image source.

*index*

The index of the image whose properties you want to obtain. The index is zero-based.

*options*

A dictionary you can use to request additional options. See “[Image Source Option Dictionary Keys](#)” (page 16) for the keys you can supply.

**Return Value**

A dictionary that contains the properties associated with the image. See *CGImageProperties Reference* for a list of properties that can be in the dictionary.

**Availability**

Available in Mac OS X version 10.4 and later.

**Declared In**

CGImageSource.h

**CGImageSourceCopyTypeIdentifiers**

Returns an array of uniform type identifiers (UTIs) that are supported for image sources.

```
CFArrayRef CGImageSourceCopyTypeIdentifiers (
    void
);
```

**Return Value**

Returns an array of the UTIs that are supported for image sources.

**Discussion**

See Uniform Type Identifiers Overview for a list of system-declared and third-party UTIs.

**Availability**

Available in Mac OS X version 10.4 and later.

**Related Sample Code**

CarbonCocoa\_PictureCursor

**Declared In**

CGImageSource.h

**CGImageSourceCreateImageAtIndex**

Creates a CGImage object for the image data associated with the specified index in an image source.

```
CGImageRef CGImageSourceCreateImageAtIndex (
    CGImageSourceRef isrc,
    size_t index,
    CFDictionaryRef options
);
```

**Parameters**

*isrc*

An image source.

*index*

The index that specifies the location of the image. The index is zero-based.

*options*

A dictionary that specifies additional creation options. See [“Image Source Option Dictionary Keys”](#) (page 16) for the keys you can supply.

**Return Value**

Returns a CGImage object. You are responsible for releasing this object using `CGImageRelease`.

**Availability**

Available in Mac OS X version 10.4 and later.

**Related Sample Code**

CarbonCocoa\_PictureCursor

**Declared In**

CGImageSource.h

**CGImageSourceCreateIncremental**

Create an incremental image source.



```
CGImageSourceRef CGImageSourceCreateIncremental (
    CFDictionaryRef options
);
```

**Parameters***options*

A dictionary that specifies additional creation options. See [“Image Source Option Dictionary Keys”](#) (page 16) for the keys you can supply.

**Return Value**

Returns an image source object. You are responsible for releasing this object using `CFRelease`.

**Discussion**

The function `CGImageSourceCreateIncremental` creates an empty image source container to which you can add data later by calling the functions `CGImageSourceUpdateDataProvider` or `CGImageSourceUpdateData`. You don't provide data when you call this function.

An incremental image is an image that is created in chunks, similar to the way large images viewed over the web are loaded piece by piece.

**Availability**

Available in Mac OS X version 10.4 and later.

**Declared In**

`CGImageSource.h`

**CGImageSourceCreateThumbnailAtIndex**

Creates a thumbnail image of the image located at a specified location in an image source.

```
CGImageRef CGImageSourceCreateThumbnailAtIndex (
    CGImageSourceRef isrc,
    size_t index,
    CFDictionaryRef options
);
```

**Parameters***isrc*

An image source.

*index*

The index that specifies the location of the image. The index is zero-based.

*options*

A dictionary that specifies additional creation options. See [“Image Source Option Dictionary Keys”](#) (page 16) for the keys you can supply.

**Return Value**

A `CGImage` object. You are responsible for releasing this object using `CGImageRelease`.

**Discussion**

If the image source is a PDF, this function creates a 72 dpi image of the PDF page specified by the index that you pass. You must, however, pass an options dictionary that contains either the `kCGImageSourceCreateThumbnailFromImageIfAbsent` or `kCGImageSourceCreateThumbnailFromImageAlways` keys, with the value of the key set to `TRUE`.

**Availability**

Available in Mac OS X version 10.4 and later.

**Declared In**

CGImageSource.h

**CGImageSourceCreateWithData**

Creates an image source that reads from a Core Foundation data object.

```
CGImageSourceRef CGImageSourceCreateWithData (
    CFDataRef data,
    CFDictionaryRef options
);
```

**Parameters**

*data*

The data object to read from. For more information on data objects, see *CFData Reference* and *Data Objects*.

*options*

A dictionary that specifies additional creation options. See [“Image Source Option Dictionary Keys”](#) (page 16) for the keys you can supply.

**Return Value**

An image source. You are responsible for releasing this object using `CFRelease`.

**Availability**

Available in Mac OS X version 10.4 and later.

**Declared In**

CGImageSource.h

**CGImageSourceCreateWithDataProvider**

Creates an image source that reads data from the specified data provider.

```
CGImageSourceRef CGImageSourceCreateWithDataProvider (
    CGDataProviderRef provider,
    CFDictionaryRef options
);
```

**Parameters**

*provider*

The data provider to read from. For more information on data providers, see *CGDataProvider Reference* and *Quartz 2D Programming Guide*.

*options*

A dictionary that specifies additional creation options. See [“Image Source Option Dictionary Keys”](#) (page 16) for the keys you can supply.

**Return Value**

An image source. You are responsible for releasing this object using `CFRelease`.

**Availability**

Available in Mac OS X version 10.4 and later.

**Declared In**

CGImageSource.h

**CGImageSourceCreateWithURL**

Creates an image source that reads from a location specified by a URL.

```
CGImageSourceRef CGImageSourceCreateWithURL (
    CFURLRef url,
    CFDictionaryRef options
);
```

**Parameters**

*url*

The URL to read from.

*options*

A dictionary that specifies additional creation options. See [“Image Source Option Dictionary Keys”](#) (page 16) for the keys you can supply.

**Return Value**

An image source. You are responsible for releasing this object using `CFRelease`.

**Availability**

Available in Mac OS X version 10.4 and later.

**Related Sample Code**

CarbonCocoa\_PictureCursor

**Declared In**

CGImageSource.h

**CGImageSourceGetCount**

Returns the number of images (not including thumbnails) in the image source.

```
size_t CGImageSourceGetCount (
    CGImageSourceRef isrc
);
```

**Parameters**

*isrc*

An image source.

**Return Value**

The number of images. If the image source is a multilayered PSD file, the function returns 1.

**Discussion**

This function does not extract the layers of a PSD file.

**Availability**

Available in Mac OS X version 10.4 and later.

**Declared In**

CGImageSource.h

**CGImageSourceGetStatus**

Return the status of an image source.

```
CGImageSourceStatus CGImageSourceGetStatus (
    CGImageSourceRef isrc
);
```

**Parameters***isrc*

An image source.

**Return Value**Returns the current status of the image source. See “[Image Source Status](#)” (page 15) for a list of possible values.**Discussion**

The status is particularly informative for incremental image sources, but may also be used by clients that provide non-incremental data.

**Availability**

Available in Mac OS X version 10.4 and later.

**Declared In**

CGImageSource.h

**CGImageSourceGetStatusAtIndex**

Returns the current status of an image that is at a specified location in an image source.

```
CGImageSourceStatus CGImageSourceGetStatusAtIndex (
    CGImageSourceRef isrc,
    size_t index
);
```

**Parameters***isrc*

An image source.

*index*

The index of the image whose status you want to obtain. The index is zero-based.

**Return Value**Returns the current status of the image. See “[Image Source Status](#)” (page 15) for a list of possible values.**Discussion**

The status is particularly informative for incremental image sources, but may also be used by clients that provide non-incremental data.

**Availability**

Available in Mac OS X version 10.4 and later.

**Declared In**

CGImageSource.h

**CGImageSourceType**

Returns the uniform type identifier of the source container.

```
CFStringRef CGImageSourceType (
    CGImageSourceRef isrc
);
```

**Parameters***isrc*

An image source.

**Return Value**

The uniform type identifier of the image.

**Discussion**

The uniform type identifier (UTI) of the source container can be different from the type of the images in the container. For example, the `.icns` format supports embedded JPEG2000. The type of the source container is `"com.apple.icns"` but type of the images is `JPEG2000`.

See Uniform Type Identifier Concepts for a list of system-declared and third-party UTIs.

**Availability**

Available in Mac OS X version 10.4 and later.

**Declared In**

CGImageSource.h

**CGImageSourceTypeID**

Returns the unique type identifier of an image source opaque type.

```
CTypeID CGImageSourceTypeID (
    void
);
```

**Return Value**

Returns the Core Foundation type ID for an image source.

**Discussion**

A type identifier is an integer that identifies the opaque type to which a Core Foundation object belongs. You use type IDs in various contexts, such as when you are operating on heterogeneous collections. Note that a CTypeID is different from a uniform type identifier (UTI).

**Availability**

Available in Mac OS X version 10.4 and later.

**Declared In**

CGImageSource.h

## CGImageSourceUpdateData

Updates an incremental image source with new data.

```
void CGImageSourceUpdateData (
    CGImageSourceRef isrc,
    CFDataRef data,
    bool final
);
```

### Parameters

*isrc*

An image source.

*data*

The data to add to the image source. Each time you call the function `CGImageSourceUpdateData`, the `data` parameter must contain all of the image file data accumulated so far.

*final*

A value that specifies whether the data is the final set. Pass `true` if it is, `false` otherwise.

### Availability

Available in Mac OS X version 10.4 and later.

### Declared In

`CGImageSource.h`

## CGImageSourceUpdateDataProvider

Updates an incremental image source with a new data provider.

```
void CGImageSourceUpdateDataProvider (
    CGImageSourceRef isrc,
    CGDataProviderRef provider,
    bool final
);
```

### Parameters

*isrc*

An image source.

*provider*

The new data provider. The new data provider must provide all the previous data supplied to the image source plus any additional new data.

*final*

A value that specifies whether the data is the final set. Pass `true` if it is, `false` otherwise.

### Availability

Available in Mac OS X version 10.4 and later.

### Declared In

`CGImageSource.h`

## Data Types

### CGImageSourceRef

An opaque type that represents an image source.

```
typedef struct CGImageSource *CGImageSourceRef;
```

#### Availability

Available in Mac OS X v10.4 and later.

#### Declared In

CGImageSource.h

## Constants

### Image Source Status

Status states for images and image sources.

```
enum CGImageSourceStatus {
    kCGImageStatusUnexpectedEOF = -5,
    kCGImageStatusInvalidData = -4,
    kCGImageStatusUnknownType = -3,
    kCGImageStatusReadingHeader = -2,
    kCGImageStatusIncomplete = -1,
    kCGImageStatusComplete = 0
};
typedef enum CGImageSourceStatus CGImageSourceStatus;
```

#### Constants

kCGImageStatusUnexpectedEOF

The end of the file was encountered unexpectedly.

Available in Mac OS X v10.4 and later.

Declared in CGImageSource.h.

kCGImageStatusInvalidData

The data is not valid.

Available in Mac OS X v10.4 and later.

Declared in CGImageSource.h.

kCGImageStatusUnknownType

The image is an unknown type.

Available in Mac OS X v10.4 and later.

Declared in CGImageSource.h.

`kCGImageStatusReadingHeader`  
**In the process of reading the header.**  
**Available in Mac OS X v10.4 and later.**  
**Declared in `CGImageSource.h`.**

`kCGImageStatusIncomplete`  
**The operation is not complete**  
**Available in Mac OS X v10.4 and later.**  
**Declared in `CGImageSource.h`.**

`kCGImageStatusComplete`  
**The operation is complete.**  
**Available in Mac OS X v10.4 and later.**  
**Declared in `CGImageSource.h`.**

### Discussion

These status values are returned by the functions [CGImageSourceGetStatus](#) (page 12) and [CGImageSourceGetStatusAtIndex](#) (page 12).

### Declared In

`CGImageSource.h`

## Image Source Option Dictionary Keys

Keys that you can include in the options dictionary to create an image source.

```
CFStringRef kCGImageSourceTypeIdentifierHint;
CFStringRef kCGImageSourceShouldAllowFloat;
CFStringRef kCGImageSourceShouldCache;
CFStringRef kCGImageSourceCreateThumbnailFromImageIfAbsent;
CFStringRef kCGImageSourceCreateThumbnailFromImageAlways;
CFStringRef kCGImageSourceThumbnailMaxPixelSize;
CFStringRef kCGImageSourceCreateThumbnailWithTransform
```

### Constants

`kCGImageSourceTypeIdentifierHint`  
**The best guess of the uniform type identifier (UTI) for the format of the image source file. If specified, the value of this key must be a CFString object. This key can be provided in the options dictionary when you create a CGImageSource object.**  
**Available in Mac OS X v10.4 and later.**  
**Declared in `CGImageSource.h`.**

`kCGImageSourceShouldAllowFloat`  
**Whether the image should be returned as a CGImage object that uses floating-point values, if supported by the file format. CGImage objects that use extended-range floating-point values may require additional processing to render in a pleasing manner. The value of this key must be a CFBoolean value. The default value is `kCFBooleanFalse`.**  
**Available in Mac OS X v10.4 and later.**  
**Declared in `CGImageSource.h`.**



`kCGImageSourceShouldCache`

Whether the image should be cached in a decoded form. The value of this key must be a `CFBoolean` value. The default value is `kCFBooleanTrue`. This key can be provided in the options dictionary that you can pass to the functions [CGImageSourceCopyPropertiesAtIndex](#) (page 7) and [CGImageSourceCreateImageAtIndex](#) (page 8).

Available in Mac OS X v10.4 and later.

Declared in `CGImageSource.h`.

`kCGImageSourceCreateThumbnailFromImageIfAbsent`

Whether a thumbnail should be automatically created for an image if a thumbnail isn't present in the image source file. The thumbnail is created from the full image, subject to the limit specified by `kCGImageSourceThumbnailMaxPixelSize`. If a maximum pixel size isn't specified, then the thumbnail is the size of the full image, which in most cases is not desirable. This key must be a `CFBoolean` value. The default value is `kCFBooleanFalse`. This key can be provided in the options dictionary that you pass to the function [CGImageSourceCreateThumbnailAtIndex](#) (page 9).

Available in Mac OS X v10.4 and later.

Declared in `CGImageSource.h`.

`kCGImageSourceCreateThumbnailFromImageAlways`

Whether a thumbnail should be created from the full image even if a thumbnail is present in the image source file. The thumbnail is created from the full image, subject to the limit specified by `kCGImageSourceThumbnailMaxPixelSize`. If a maximum pixel size isn't specified, then the thumbnail is the size of the full image, which probably isn't what you want. This key must be a `CFBoolean` value. The default value is `kCFBooleanFalse`. This key can be provided in the options dictionary that you can pass to the function [CGImageSourceCreateThumbnailAtIndex](#) (page 9).

Available in Mac OS X v10.4 and later.

Declared in `CGImageSource.h`.

`kCGImageSourceThumbnailMaxPixelSize`

The maximum width and height in pixels of a thumbnail. If this key is not specified, the width and height of a thumbnail is not limited and thumbnails may be as big as the image itself. If present, this key must be a `CFNumber` value. This key can be provided in the options dictionary that you pass to the function [CGImageSourceCreateThumbnailAtIndex](#) (page 9).

Available in Mac OS X v10.4 and later.

Declared in `CGImageSource.h`.

`kCGImageSourceCreateThumbnailWithTransform`

Whether the thumbnail should be rotated and scaled according to the orientation and pixel aspect ratio of the full image. The value of this key must be a `CFBoolean` value. The default value is `kCFBooleanFalse`.

Available in Mac OS X v10.4 and later.

Declared in `CGImageSource.h`.

**Discussion**

Except for `kCGImageSourceTypeIdentifierHint`, which you use when creating an image source, these constants specify options that you can set when creating an image from image source. Each constant is a key; you must supply the appropriate value when you add this option to the options dictionary.

**Declared In**

`CGImageSource.h`



# Document Revision History

This table describes the changes to *CGImageSource Reference*.

Date	Notes
2007-12-04	Added release information to several functions.
	Added a discussion to <a href="#">"Image Source Option Dictionary Keys"</a> (page 16).
2007-10-31	Made several technical corrections and removed image property constants.
	Removed the constants <code>kCGImageSourceShouldToneCompressFloat</code> and <code>kCGImagePropertyCompressionQuality</code> , added header file information, and moved <code>kCGImageDestinationLossyCompressionQuality</code> to <i>CGImageDestination Reference</i> .
	Moved all image property constants to <i>CGImageProperties Reference</i> .
	Adding information about PSD files to <a href="#">CGImageSourceGetCount</a> (page 11).
	Added information about creating thumbnail images from a PDF image source to <a href="#">CGImageSourceCreateThumbnailAtIndex</a> (page 9).
2006-03-08	Clarified description of the function <code>CGImageSourceUpdateData</code> .
2006-01-10	Added documentation for the constant <code>kCGImageSourceCreateThumbnailWithTransform</code> .
2005-04-29	First version.

## REVISION HISTORY

### Document Revision History

# Index

---

## C

---

- CGImageSourceCopyProperties **function** [6](#)
- CGImageSourceCopyPropertiesAtIndex **function** [7](#)
- CGImageSourceCopyTypeIdentifiers **function** [7](#)
- CGImageSourceCreateImageAtIndex **function** [8](#)
- CGImageSourceCreateIncremental **function** [8](#)
- CGImageSourceCreateThumbnailAtIndex **function** [9](#)
- CGImageSourceCreateWithData **function** [10](#)
- CGImageSourceCreateWithDataProvider **function** [10](#)
- CGImageSourceCreateWithURL **function** [11](#)
- CGImageSourceGetCount **function** [11](#)
- CGImageSourceGetStatus **function** [12](#)
- CGImageSourceGetStatusAtIndex **function** [12](#)
- CGImageSourceGetType **function** [13](#)
- CGImageSourceGetTypeID **function** [13](#)
- CGImageSourceRef **data type** [15](#)
- CGImageSourceUpdateData **function** [14](#)
- CGImageSourceUpdateDataProvider **function** [14](#)
- kCGImageSourceTypeIdentifierHint **constant** [16](#)
- kCGImageStatusComplete **constant** [16](#)
- kCGImageStatusIncomplete **constant** [16](#)
- kCGImageStatusInvalidData **constant** [15](#)
- kCGImageStatusReadingHeader **constant** [16](#)
- kCGImageStatusUnexpectedEOF **constant** [15](#)
- kCGImageStatusUnknownType **constant** [15](#)

## I

---

- Image Source Option Dictionary Keys [16](#)
- Image Source Status [15](#)

## K

---

- kCGImageSourceCreateThumbnailFromImageAlways **constant** [17](#)
- kCGImageSourceCreateThumbnailFromImageIfAbsent **constant** [17](#)
- kCGImageSourceCreateThumbnailWithTransform **constant** [17](#)
- kCGImageSourceShouldAllowFloat **constant** [16](#)
- kCGImageSourceShouldCache **constant** [17](#)
- kCGImageSourceThumbnailMaxPixelSize **constant** [17](#)