
NSImageRep Class Reference

[Cocoa](#) > [Graphics & Imaging](#)



2009-01-06



Apple Inc.
© 2009 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, Cocoa, Mac, Mac OS, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

OpenGL is a registered trademark of Silicon Graphics, 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

NSImageRep Class Reference 5

Overview	5
Adopted Protocols	5
Tasks	6
Creating an NSImageRep	6
Determining the Supported Image Types	6
Setting the Size of the Image	7
Specifying Information About the Representation	7
Drawing the Image	7
Managing NSImageRep Subclasses	8
Class Methods	8
canInitWithData:	8
canInitWithPasteboard:	9
imageFileTypes	9
imagePasteboardTypes	10
imageRepClassForData:	10
imageRepClassForFileType:	10
imageRepClassForPasteboardType:	11
imageRepClassForType:	11
imageRepsWithContentsOfFile:	11
imageRepsWithContentsOfURL:	12
imageRepsWithPasteboard:	13
imageRepWithContentsOfFile:	14
imageRepWithContentsOfURL:	15
imageRepWithPasteboard:	15
imageTypes	16
imageUnfilteredFileTypes	16
imageUnfilteredPasteboardTypes	17
imageUnfilteredTypes	18
registeredImageRepClasses	18
registerImageRepClass:	19
unregisterImageRepClass:	19
Instance Methods	20
bitsPerSample	20
colorSpaceName	20
draw	21
drawAtPoint:	21
drawInRect:	22
hasAlpha	22
isOpaque	23
pixelsHigh	23

- pixelsWide 24
- setAlpha: 24
- setBitsPerSample: 25
- setColorSpaceName: 25
- setOpaque: 26
- setPixelsHigh: 26
- setPixelsWide: 27
- setSize: 27
- size 28
- Constants 29
 - Display Device Matching 29
 - Obsolete Change Notification Name 29
- Notifications 29
 - NSImageRepRegistryDidChangeNotification 29

Document Revision History 31

Index 33

NSImageRep Class Reference

Inherits from	NSObject
Conforms to	NSCoding NSCopying NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Cocoa Drawing Guide
Declared in	NSImageRep.h
Related sample code	LayerBackedOpenGLView NSOpenGL Fullscreen Reducer

Overview

The `NSImageRep` class is a semiabstract superclass (“semi” because it has some instance variables and implementation of its own). Each of its subclasses knows how to draw an image from a particular kind of source data. While an `NSImageRep` subclass can be used directly, it is typically through an `NSImage` object. An `NSImage` object manages a group of image representations, choosing the best one for the current output device.

Adopted Protocols

NSCoding
 `encodeWithCoder:`
 `initWithCoder:`

NSCopying
 `copyWithZone:`

Tasks

Creating an NSImageRep

- + [imageRepsWithContentsOfFile:](#) (page 11)
Creates and returns an array of image representation objects initialized using the contents of the specified file.
- + [imageRepsWithPasteboard:](#) (page 13)
Creates and returns an array of image representation objects initialized using the contents of the pasteboard.
- + [imageRepsWithContentsOfURL:](#) (page 12)
Creates and returns an array of image representation objects initialized using the contents of the specified URL.
- + [imageRepWithContentsOfFile:](#) (page 14)
Creates and returns an image representation object using the contents of the specified file.
- + [imageRepWithPasteboard:](#) (page 15)
Creates and returns an image representation object using the contents of the specified pasteboard.
- + [imageRepWithContentsOfURL:](#) (page 15)
Creates and returns an image representation object using the data at the specified URL.

Determining the Supported Image Types

- + [canInitWithData:](#) (page 8)
Returns a Boolean value indicating whether the receiver can initialize itself from the specified data.
- + [canInitWithPasteboard:](#) (page 9)
Returns a Boolean value indicating whether the receiver can initialize itself from the data on the specified pasteboard.
- + [imageTypes](#) (page 16)
Returns an array of UTI strings identifying the image types supported by the receiver, either directly or through a user-installed filter service.
- + [imageUnfilteredTypes](#) (page 18)
Returns an array of UTI strings identifying the image types supported directly by the receiver.
- + [imageFileTypes](#) (page 9)
Returns the file types supported by NSImageRep or one of its subclasses.
- + [imagePasteboardTypes](#) (page 10)
Returns the pasteboard types supported by NSImageRep or one of its subclasses.
- + [imageUnfilteredFileTypes](#) (page 16)
Returns the list of file types supported directly by the receiver.
- + [imageUnfilteredPasteboardTypes](#) (page 17)
Returns the list of pasteboard types supported directly by the receiver.

Setting the Size of the Image

- `setSize:` (page 27)
Sets the size of the image representation to the specified value.
- `size` (page 28)
Returns the size of the image representation.

Specifying Information About the Representation

- `bitsPerSample` (page 20)
Returns the number of bits per sample in the receiver.
- `colorSpaceName` (page 20)
Returns the name of the receiver's color space.
- `hasAlpha` (page 22)
Returns a Boolean value indicating whether the receiver has an alpha channel.
- `isOpaque` (page 23)
Returns a Boolean value indicating whether the receiver is opaque.
- `pixelsHigh` (page 23)
Returns the height of the image, measured in pixels.
- `pixelsWide` (page 24)
Returns the width of the image, measured in pixels.
- `setAlpha:` (page 24)
Informs the receiver that its image data has an alpha component.
- `setBitsPerSample:` (page 25)
Informs the receiver that its image data has the specified number of bits for each component of a pixel.
- `setColorSpaceName:` (page 25)
Informs the receiver of the color space used by the image data.
- `setOpaque:` (page 26)
Sets whether the receiver's image is opaque.
- `setPixelsHigh:` (page 26)
Informs the receiver of the image data height.
- `setPixelsWide:` (page 27)
Informs the receiver of the image data width.

Drawing the Image

- `draw` (page 21)
Implemented by subclasses to draw the image in the current coordinate system.
- `drawAtPoint:` (page 21)
Draws the receiver's image data at the specified point in the current coordinate system.
- `drawInRect:` (page 22)
Draws the image, scaling it (as needed) to fit the specified rectangle.

Managing NSImageRep Subclasses

- + [imageRepClassForType:](#) (page 11)
Returns the NSImageRep subclass that handles image data for the specified UTI.
- + [imageRepClassForData:](#) (page 10)
Returns the NSImageRep subclass that handles the specified type of data.
- + [imageRepClassForFileType:](#) (page 10)
Returns the NSImageRep subclass that handles data with the specified type.
- + [imageRepClassForPasteboardType:](#) (page 11)
Returns the NSImageRep subclass that handles data with the specified pasteboard type.
- + [registeredImageRepClasses](#) (page 18)
Returns an array containing the registered NSImageRep classes.
- + [registerImageRepClass:](#) (page 19)
Adds the specified class to the registry of available NSImageRep subclasses.
- + [unregisterImageRepClass:](#) (page 19)
Removes the specified NSImageRep subclass from the registry of available image representations.

Class Methods

canInitWithData:

Returns a Boolean value indicating whether the receiver can initialize itself from the specified data.

```
+ (BOOL)canInitWithData:(NSData *)data
```

Parameters

data

The image data.

Return Value

YES if the receiver understands the format of the specified data and can use it to initialize itself; otherwise, NO.

Discussion

This method should be overridden by subclasses. Note that this method does not need to do a comprehensive check of the image data; it should return NO only if it knows it cannot initialize itself from the data.

Availability

Available in Mac OS X v10.0 and later.

See Also

+ [canInitWithPasteboard:](#) (page 9)

Declared In

NSImageRep.h

canInitWithPasteboard:

Returns a Boolean value indicating whether the receiver can initialize itself from the data on the specified pasteboard.

```
+ (BOOL)canInitWithPasteboard:(NSPasteboard *)pasteboard
```

Parameters

pasteboard

The pasteboard containing the image data.

Return Value

YES if the receiver understands the format of the specified data and can use it to initialize itself; otherwise, NO.

Discussion

This method invokes the [imageUnfilteredPasteboardTypes](#) (page 17) class method and checks the list of types returned by that method against the data types in *pasteboard*. If it finds a match, it returns YES. When creating a subclass of NSImageRep that accepts image data from a non-default pasteboard type, override the [imageUnfilteredPasteboardTypes](#) (page 17) method to assure this method returns the correct response.

Availability

Available in Mac OS X v10.0 and later.

See Also

+ [canInitWithData:](#) (page 8)

Declared In

NSImageRep.h

imageFileTypes

Returns the file types supported by NSImageRep or one of its subclasses.

```
+ (NSArray *)imageFileTypes
```

Return Value

An array of NSString objects, each of which contains a filename extension or HFS file type of a supported format.

Discussion

The list includes both those types returned by the [imageUnfilteredFileTypes](#) (page 16) class method plus those that can be converted to a supported type by a user-installed filter service. The returned file types can include encoded HFS file types as well as filename extensions.

Don't override this method when subclassing NSImageRep—it always returns a valid list for any subclass of NSImageRep that correctly overrides the [imageUnfilteredFileTypes](#) (page 16) method.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSImageRep.h

imagePasteboardTypes

Returns the pasteboard types supported by `NSImageRep` or one of its subclasses.

```
+ (NSArray *)imagePasteboardTypes
```

Return Value

An array of `NSString` objects, each of which contains a supported pasteboard format.

Discussion

The list includes both those types returned by the `imageUnfilteredPasteboardTypes` (page 17) class method plus those that can be converted to a supported type by a user-installed filter service. Don't override this method when subclassing `NSImageRep`—it always returns a valid list for any subclass of `NSImageRep` that correctly overrides the `imageUnfilteredPasteboardTypes` (page 17) method.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSImageRep.h`

imageRepClassForData:

Returns the `NSImageRep` subclass that handles the specified type of data.

```
+ (Class)imageRepClassForData:(NSData *)data
```

Parameters

data

The image data.

Return Value

A `Class` object for the image representation that can handle the data, or `nil` if no image representation could handle the data.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSImageRep.h`

imageRepClassForFileType:

Returns the `NSImageRep` subclass that handles data with the specified type.

```
+ (Class)imageRepClassForFileType:(NSString *)type
```

Parameters

type

A string containing the filename extension or an encoded HFS type.

Return Value

A `Class` object for the image representation that can handle the type of data, or `nil` if no image representation could handle the type.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSImageRep.h

imageRepClassForPasteboardType:

Returns the `NSImageRep` subclass that handles data with the specified pasteboard type.

```
+ (Class)imageRepClassForPasteboardType:(NSString *)type
```

Parameters

type

The pasteboard type.

Return Value

A `Class` object for the image representation that can handle the specified pasteboard type, or `nil` if no image representation could handle the type.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSImageRep.h

imageRepClassForType:

Returns the `NSImageRep` subclass that handles image data for the specified UTI.

```
+ (Class)imageRepClassForType:(NSString *)type
```

Parameters

type

The UTI string identifying the desired image type. Some sample image-related UTI strings include "public.image", "public.jpeg", and "public.tiff". For a list of supported types, see `UTCoreTypes.h`.

Return Value

A `Class` object for the image representation that can handle the UTI, or `nil` if no image representation could handle the data.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSImageRep.h

imageRepsWithContentsOfFile:

Creates and returns an array of image representation objects initialized using the contents of the specified file.

```
+ (NSArray *)imageRepsWithContentsOfFile:(NSString *)filename
```

Parameters

filename

A full or relative pathname specifying the file to open. This string should include the filename extension.

Return Value

An array of image representation objects. The array contains one object for each image in the specified file.

Discussion

If sent to the `NSImageRep` class object, this method returns an array of objects (all newly allocated instances of a subclass of `NSImageRep`, chosen through the use of `imageRepClassForFileType:` (page 10)) that have been initialized with the contents of the file. If sent to a subclass of `NSImageRep` that recognizes the file type, this method returns an array of objects (all instances of that subclass) that have been initialized with the contents of the file.

This method returns `nil` in any of the following cases:

- The message is sent to the `NSImageRep` class object and there are no subclasses in the `NSImageRep` class registry that handle the data in the file.
- The message is sent to a subclass of `NSImageRep` and that subclass cannot handle the data in the file.
- The `NSImageRep` subclass is unable to initialize itself with the contents of *filename*.

The `NSImageRep` subclass is initialized by creating an `NSData` object based on the contents of the file and passing it to the `imageRepsWithData:` method of the subclass. By default, the files handled include those with the extensions “tiff”, “gif”, “jpg”, “pict”, “pdf”, and “eps”.

Availability

Available in Mac OS X v10.0 and later.

See Also

+ [imageFileTypes](#) (page 9)

Declared In

`NSImageRep.h`

imageRepsWithContentsOfURL:

Creates and returns an array of image representation objects initialized using the contents of the specified URL.

```
+ (NSArray *)imageRepsWithContentsOfURL:(NSURL *)aURL
```

Parameters

aURL

The URL pointing to the image data.

Return Value

An array of image representation objects. The array contains one object for each image in the data at the specified URL.

Discussion

If sent to the `NSImageRep` class object, this method returns an array of objects (all newly allocated instances of a subclass of `NSImageRep`) that have been initialized with the contents of the specified URL. If sent to a subclass of `NSImageRep` that recognizes the data at the specified URL, it returns an array of objects (all instances of that subclass) that have been initialized with the contents of that URL.

This method returns `nil` in any of the following cases:

- The message is sent to the `NSImageRep` class object and there are no subclasses in the `NSImageRep` class registry that handle data in the specified URL.
- The message is sent to a subclass of `NSImageRep` and that subclass cannot handle data in the specified URL.
- The `NSImageRep` subclass is unable to initialize itself with the contents of the specified URL.

The `NSImageRep` subclass is initialized by creating an `NSData` object based on the contents of the specified URL and passing it to the `initWithData:` method.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSImageRep.h`

imageRepsWithPasteboard:

Creates and returns an array of image representation objects initialized using the contents of the pasteboard.

```
+ (NSArray *)imageRepsWithPasteboard:(NSPasteboard *)pasteboard
```

Parameters

pasteboard

The pasteboard containing the image data.

Return Value

An array of image representation objects. The array contains one object for each image in the specified pasteboard.

Discussion

If sent to the `NSImageRep` class object, this method returns an array of objects (all newly-allocated instances of a subclass of `NSImageRep`) that have been initialized with the data in the specified pasteboard. If sent to a subclass of `NSImageRep` that recognizes the pasteboard data, it returns an array of objects (all instances of that subclass) initialized with the pasteboard data.

This method returns `nil` in any of the following cases:

- The message is sent to the `NSImageRep` class object and there are no subclasses in the `NSImageRep` class registry that handle the pasteboard data.
- The message is sent to a subclass of `NSImageRep` and that subclass cannot handle the pasteboard data.
- The `NSImageRep` subclass is unable to initialize itself with the contents the pasteboard.

The `NSImageRep` subclass is initialized by creating an `NSData` object based on the data in *pasteboard* and passing it to the `imageRepWithData:` method.

Availability

Available in Mac OS X v10.0 and later.

See Also

+ [imagePasteboardTypes](#) (page 10)

Declared In

`NSImageRep.h`

imageRepWithContentsOfFile:

Creates and returns an image representation object using the contents of the specified file.

```
+ (id)imageRepWithContentsOfFile:(NSString *)filename
```

Parameters

filename

A full or relative pathname specifying the file to open. This string should include the filename extension.

Return Value

An initialized instance of an `NSImageRep` subclass, or `nil` if the image data could not be read.

Discussion

If sent to the `NSImageRep` class object, this method returns a newly allocated instance of a subclass of `NSImageRep` (chosen through the use of [imageRepClassForFileType:](#) (page 10)) initialized with the contents of the specified file. If sent to a subclass of `NSImageRep` that recognizes the type of data in the file, it returns an instance of that subclass initialized with the contents of the file.

This method returns `nil` in any of the following cases:

- The message is sent to the `NSImageRep` class object and there are no subclasses in the `NSImageRep` class registry that handle the type of data in the specified file.
- The message is sent to a subclass of `NSImageRep` and that subclass cannot handle the type of data in the specified file.
- The `NSImageRep` subclass is unable to initialize itself with the contents of the specified file.

The `NSImageRep` subclass is initialized by creating an `NSData` object based on the contents of the file and passing it to the `imageRepWithData:` method. By default, the files handled include those with the extensions “tiff”, “gif”, “jpg”, “pict”, “pdf”, and “eps”.

Availability

Available in Mac OS X v10.0 and later.

See Also

+ [imageFileTypes](#) (page 9)

Related Sample Code

`NSGLImage`

`PDFView`

Declared In

NSImageRep.h

imageRepWithContentsOfURL:

Creates and returns an image representation object using the data at the specified URL.

+ (id)imageRepWithContentsOfURL:(NSURL *)aURL

Parameters*aURL*

The URL pointing to the image data.

Return Value

An initialized instance of an NSImageRep subclass, or nil if the image data could not be read.

Discussion

If sent to the NSImageRep class object, this method returns a newly allocated instance of a subclass of NSImageRep initialized with the contents of the specified URL. If sent to a subclass of NSImageRep that recognizes the data contained in the URL, it returns an instance of that subclass initialized with the data in the URL.

This method returns nil in any of the following cases:

- The message is sent to the NSImageRep class object and there are no subclasses in the NSImageRep class registry that handle the data contained in the specified URL.
- The message is sent to a subclass of NSImageRep and that subclass cannot handle the data contained in the specified URL.
- The NSImageRep subclass is unable to initialize itself with the contents of the specified URL.

The NSImageRep subclass is initialized by creating an NSData object based on the contents of the file, then passing it to the imageRepWithData: method.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSImageRep.h

imageRepWithPasteboard:

Creates and returns an image representation object using the contents of the specified pasteboard.

+ (id)imageRepWithPasteboard:(NSPasteboard *)pasteboard

Parameters*pasteboard*

The pasteboard containing the image data.

Return Value

An initialized instance of an NSImageRep subclass, or nil if the image data could not be read.

Discussion

If sent to the `NSImageRep` class object, this method returns a newly allocated instance of a subclass of `NSImageRep` initialized with the data in the specified pasteboard. If sent to a subclass of `NSImageRep` that recognizes the data on the pasteboard, it returns an instance of that subclass initialized with that data.

This method returns `nil` in any of the following cases:

- The message is sent to the `NSImageRep` class object and there are no subclasses in the `NSImageRep` class registry that handle data of the type contained in the specified pasteboard.
- The message is sent to a subclass of `NSImageRep` and that subclass cannot handle data of the type contained in the specified pasteboard.
- The `NSImageRep` subclass is unable to initialize itself with the contents of the pasteboard.

The `NSImageRep` subclass is initialized by creating an `NSData` object based on the data the specified pasteboard and passing it to the `imageRepWithData:` method.

Availability

Available in Mac OS X v10.0 and later.

See Also

+ [imagePasteboardTypes](#) (page 10)

Declared In

`NSImageRep.h`

imageTypes

Returns an array of UTI strings identifying the image types supported by the receiver, either directly or through a user-installed filter service.

```
+ (NSArray *)imageTypes
```

Return Value

An array of `NSString` objects, each of which contains a UTI identifying a supported image type. Some sample image-related UTI strings include `"public.image"`, `"public.jpeg"`, and `"public.tiff"`. For a list of supported types, see `UTCoreTypes.h`.

Discussion

The returned list includes UTIs all file types supported by this image representation object plus those that can be opened by this image representation after being converted by a user-installed filter service. You can use the returned UTI strings with any method that supports UTIs.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSImageRep.h`

imageUnfilteredFileTypes

Returns the list of file types supported directly by the receiver.


```
+ (NSArray *)imageUnfilteredFileTypes
```

Return Value

An array of `NSString` objects. This array is empty by default. Subclasses must override to return the list of file formats they support.

Discussion

The returned file types can include encoded HFS file types as well as filename extensions. When creating a subclass of `NSImageRep`, override this method to return a list of strings representing the supported file types. For example, the `NSBitmapImageRep` class implements code similar to the following for this method:

```
+ (NSArray *)imageUnfilteredFileTypes {
    static NSArray *types = nil;

    if (!types) types = [[NSArray alloc]
        initWithObjects:@"tiff", @"gif", @"jpg", @"bmp", nil];
    return types;
}
```

If your subclass supports the types supported by its superclass, you must explicitly get the array of types from the superclass and put them in the array returned by this method.

Availability

Available in Mac OS X v10.0 and later.

See Also

+ [imageFileTypes](#) (page 9)

+ [imageUnfilteredFileTypes](#) (`NSImage`)

Declared In

`NSImageRep.h`

imageUnfilteredPasteboardTypes

Returns the list of pasteboard types supported directly by the receiver.

```
+ (NSArray *)imageUnfilteredPasteboardTypes
```

Return Value

An array of `NSString` objects. This array is empty by default. Subclasses must override to return the list of pasteboard formats they support.

Discussion

When creating a subclass of `NSImageRep`, override this method to return a list representing the supported pasteboard types. For example, the `NSBitmapImageRep` class implements code similar to the following for this method:

```
+ (NSArray *)imageUnfilteredPasteboardTypes {
    static NSArray *types = nil;

    if (!types) types = [[NSArray alloc] initWithObjects:NSTIFFPboardType,
        nil];
    return types;
}
```

If your subclass supports the types supported by its superclass, you must explicitly get the list of types from the superclass and add them to the array returned by this method.

Availability

Available in Mac OS X v10.0 and later.

See Also

- + [imagePasteboardTypes](#) (page 10)
- + [imageUnfilteredPasteboardTypes](#) (NSImage)

Declared In

NSImageRep.h

imageUnfilteredTypes

Returns an array of UTI strings identifying the image types supported directly by the receiver.

```
+ (NSArray *)imageUnfilteredTypes
```

Return Value

An array of `NSString` objects, each of which contains a UTI identifying a supported image type. Some sample image-related UTI strings include "public.image", "public.jpeg", and "public.tiff". For a list of supported types, see `UTCoreTypes.h`.

Discussion

The returned list includes UTI strings only for those file types that are supported directly by the receiver. It does not include types that are supported through user-installed filter services. You can use the returned UTI strings with any method that supports UTIs.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSImageRep.h

registeredImageRepClasses

Returns an array containing the registered `NSImageRep` classes.

```
+ (NSArray *)registeredImageRepClasses
```

Return Value

An array of `Class` objects identifying the registered `NSImageRep` subclasses.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSImageRep.h

registerImageRepClass:

Adds the specified class to the registry of available NSImageRep subclasses.

```
+ (void)registerImageRepClass:(Class)imageRepClass
```

Parameters

imageRepClass

The Class object for an NSImageRep subclass.

Discussion

This method posts an [NSImageRepRegistryDidChangeNotification](#) (page 29), along with the receiving object, to the default notification center.

A good place to add image representation classes to the registry is in the `load` class method.

Availability

Available in Mac OS X v10.0 and later.

See Also

+ [unregisterImageRepClass:](#) (page 19)

`load` (NSObject)

Declared In

NSImageRep.h

unregisterImageRepClass:

Removes the specified NSImageRep subclass from the registry of available image representations.

```
+ (void)unregisterImageRepClass:(Class)imageRepClass
```

Parameters

imageRepClass

The Class object for an NSImageRep subclass.

Discussion

This method posts the [NSImageRepRegistryDidChangeNotification](#) (page 29), along with the receiving object, to the default notification center.

Availability

Available in Mac OS X v10.0 and later.

See Also

+ [registerImageRepClass:](#) (page 19)

Declared In

NSImageRep.h

Instance Methods

bitsPerSample

Returns the number of bits per sample in the receiver.

- (NSInteger)bitsPerSample

Return Value

The number of bits used to specify each component of data in a single pixel (for example, a value of 8 for an RGBA image means that each pixel is comprised of four 8-bit values). May also return [NSImageRepMatchesDevice](#) (page 29).

Discussion

If the receiver is a planar image, this method returns the number of bits per sample per plane.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setBitsPerSample:](#) (page 25)
- [bitsPerPixel](#) (NSBitmapImageRep)
- [samplesPerPixel](#) (NSBitmapImageRep)
- [isPlanar](#) (NSBitmapImageRep)

Related Sample Code

Quartz EB

Declared In

NSImageRep.h

colorSpaceName

Returns the name of the receiver's color space.

- (NSString *)colorSpaceName

Return Value

The colorspace name, or `NSCalibratedRGBColorSpace` if no name has been assigned.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setColorSpaceName:](#) (page 25)

Declared In

NSImageRep.h

draw

Implemented by subclasses to draw the image in the current coordinate system.

- (BOOL)draw

Return Value

YES if the image was successfully drawn; otherwise, NO if there was a problem. The default version of this method simply returns YES.

Discussion

Subclass override this method to draw the image using the image data. By the time this method is called, the graphics state is already configured for you to draw the image at location (0.0, 0.0) in the current coordinate system.

The standard Application Kit subclasses all draw the image using the `NSCompositeCopy` composite operation defined in the “Constants” section of `NSImage`. Using the copy operator, the image data overwrites the destination without any blending effects. Transparent (alpha) regions in the source image appear black. To use other composite operations, you must place the representation into an `NSImage` object and use its `drawAtPoint:fromRect:operation:fraction:` or `drawInRect:fromRect:operation:fraction:` methods.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSImageRep.h`

drawAtPoint:

Draws the receiver's image data at the specified point in the current coordinate system.

- (BOOL)drawAtPoint:(NSPoint)aPoint

Parameters

aPoint

The point in the current coordinate system at which to draw the image.

Return Value

YES if the image was successfully drawn; otherwise, NO. If the size of the image has not yet been set, this method returns NO immediately.

Discussion

This method sets the origin of the current coordinate system to the specified point and then invokes the receiver's `draw` method to draw the image at that point. Upon completion, it restores the current coordinates to their original setting. If *aPoint* is (0.0, 0.0), this method simply invokes the `draw` (page 21) method.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setSize:](#) (page 27)
- [drawInRect:](#) (page 22)

Related Sample Code

bMoviePalette

bMoviePaletteCocoa

Declared In

NSImageRep.h

drawInRect:

Draws the image, scaling it (as needed) to fit the specified rectangle.

- (BOOL)drawInRect:(NSRect)rect

Parameters

rect

The rectangle in the current coordinate system in which to draw the image.

Return Value

YES if the image was successfully drawn; otherwise, NO. If the size of the image has not yet been set, this method returns NO immediately

Discussion

This method sets the origin of the current coordinate system to the origin of the specified rectangle before invoking the receiver's [draw](#) (page 21) method. If the rectangle size is different from the image's native size, this method adjusts the coordinate transform, causing the image to be scaled appropriately. After the [draw](#) method returns, the coordinate system changes are undone, restoring the original graphics state.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setSize:](#) (page 27)

- [drawAtPoint:](#) (page 21)

Related Sample Code

PDF Annotation Editor

PDFView

Sketch-112

Declared In

NSImageRep.h

hasAlpha

Returns a Boolean value indicating whether the receiver has an alpha channel.

- (BOOL)hasAlpha

Return Value

YES if the receiver has a known alpha channel; otherwise, NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setAlpha:](#) (page 24)

Related Sample Code

LayerBackedOpenGLView

NSOpenGL Fullscreen

Declared In

NSImageRep.h

isOpaque

Returns a Boolean value indicating whether the receiver is opaque.

- (BOOL)isOpaque

Return Value

YES if the receiver is opaque; otherwise, NO.

Discussion

Use this method to test whether an image representation completely covers the area within the rectangle returned by the [size](#) (page 28) method.

The returned value does not indicate whether the image has an alpha channel or if there is partial or complete transparency when drawing the image rep. Use the [hasAlpha](#) (page 22) method to determine if the image has an alpha channel.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setOpaque:](#) (page 26)

Declared In

NSImageRep.h

pixelsHigh

Returns the height of the image, measured in pixels.

- (NSInteger)pixelsHigh

Return Value

The height of the image, measured in the units of the device coordinate space. This value is usually derived from the image data itself.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setPixelsHigh:](#) (page 26)

- [pixelsWide](#) (page 24)

- [size](#) (page 28)

[NSImageRepMatchesDevice](#) (page 29)

Related Sample Code

OpenGLCompositorLab

Quartz EB

Reducer

Declared In

NSImageRep.h

pixelsWide

Returns the width of the image, measured in pixels.

- (NSInteger)pixelsWide

Return Value

The width of the image, measured in the units of the device coordinate space. This value is usually derived from the image data itself.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setPixelsWide:](#) (page 27)

- [pixelsHigh](#) (page 23)

- [size](#) (page 28)

[NSImageRepMatchesDevice](#) (page 29)

Related Sample Code

OpenGLCompositorLab

Quartz EB

Reducer

Declared In

NSImageRep.h

setAlpha:

Informs the receiver that its image data has an alpha component.

- (void)setAlpha:(BOOL)flag

Parameters

flag

YES if you want the receiver to have an alpha component; otherwise NO.

Discussion

Subclasses should call this method when loading image data to notify the parent class whether that data contains an alpha component. Passing in a value of YES does not add an alpha channel to the image data itself; it merely records the fact that the data has an alpha channel.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [hasAlpha](#) (page 22)

Declared In

NSImageRep.h

setBitsPerSample:

Informs the receiver that its image data has the specified number of bits for each component of a pixel.

```
- (void)setBitsPerSample:(NSInteger)anInt
```

Parameters

anInt

The number of bits used by each component of a pixel, or [NSImageRepMatchesDevice](#) (page 29).

Discussion

Subclasses should call this method when loading image data to notify the parent class of how many bits each sample uses. Specifying a value that differs from the actual image data does not change the bit depth of the image.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [bitsPerSample](#) (page 20)

Declared In

NSImageRep.h

setColorSpaceName:

Informs the receiver of the color space used by the image data.

```
- (void)setColorSpaceName:(NSString *)string
```

Parameters

string

The name of the color space used by the image data.

Discussion

By default, an `NSImageRep` object's color space name is `NSCalibratedRGBColorSpace`. Color space names are defined as part of the `NSColor` class, in `NSGraphics.h`. The following are valid color space names:

```
NSCalibratedWhiteColorSpace
NSCalibratedBlackColorSpace
NSCalibratedRGBColorSpace
NSDeviceWhiteColorSpace
NSDeviceBlackColorSpace
NSDeviceRGBColorSpace
```

NSDeviceCMYKColorSpace
NSNamedColorSpace
NSCustomColorSpace

Availability

Available in Mac OS X v10.0 and later.

See Also

- [colorSpaceName](#) (page 20)

Declared In

NSImageRep.h

setOpaque:

Sets whether the receiver's image is opaque.

- (void)setOpaque:(BOOL)flag

Parameters

flag

YES if the image should be treated as fully opaque; otherwise, NO to indicate the image may include some transparent regions.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [isOpaque](#) (page 23)

Declared In

NSImageRep.h

setPixelsHigh:

Informs the receiver of the image data height.

- (void)setPixelsHigh:(NSInteger)anInt

Parameters

anInt

The height of the image, measured in pixels.

Discussion

Subclasses should call this method when loading image data to notify the parent class of the image height. You cannot use this method to change the actual number of pixels in the image.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [pixelsHigh](#) (page 23)

- [setPixelsWide:](#) (page 27)

- [setSize:](#) (page 27)
- [NSImageRepMatchesDevice](#) (page 29)

Declared In

NSImageRep.h

setPixelsWide:

Informs the receiver of the image data width.

```
- (void)setPixelsWide:(NSInteger)anInt
```

Parameters*anInt*

The width of the image, measured in pixels.

Discussion

Subclasses should call this method when loading image data to notify the parent class of the image width. You cannot use this method to change the actual number of pixels in the image.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [pixelsWide](#) (page 24)
- [setPixelsHigh:](#) (page 26)
- [setSize:](#) (page 27)
- [NSImageRepMatchesDevice](#) (page 29)

Declared In

NSImageRep.h

setSize:

Sets the size of the image representation to the specified value.

```
- (void)setSize:(NSSize)aSize
```

Parameters*aSize*

The new size of the image representation, measured in points in the user coordinate space.

Discussion

This method determines the size of the image when it's rendered. It is not necessarily the same as the width and height of the image in pixels as specified by the image data, nor must it be equal to the size set for the `NSImage` object that wraps this image representation. You must set the image size before you can render it.

The size of an image representation combined with the physical dimensions of the image data determine the resolution of the image.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [size](#) (page 28)
- [draw](#) (page 21)
- [setPixelsHigh:](#) (page 26)
- [setPixelsWide:](#) (page 27)

Declared In

NSImageRep.h

size

Returns the size of the image representation.

- (NSSize) size

Return Value

The size of the image representation, measured in points in the user coordinate space.

Discussion

This size is the size of the image representation when it's rendered. It is not necessarily the same as the width and height of the image in pixels as specified by the image data, nor must it be equal to the size set for the `NSImage` object that wraps this image representation.

The size of an image representation combined with the physical dimensions of the image data determine the resolution of the image.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setSize:](#) (page 27)
- [pixelsHigh](#) (page 23)
- [pixelsWide](#) (page 24)

Related Sample Code

bMoviePaletteCocoa
NURBSSurfaceVertexProg
OpenGL Screensaver
Reducer
Vertex Optimization

Declared In

NSImageRep.h

Constants

Display Device Matching

The following constant is used by `NSImageRep` to denote an attribute whose value changes to match the display device.

```
enum {
    NSImageRepMatchesDevice
};
```

Constants

`NSImageRepMatchesDevice`

Indicates that the value of certain attributes, such as the number of colors or bits per sample, will change to match the display device.

This value can be passed in (or received back) as the value of `bitsPerSample` (page 20), `pixelsWide` (page 24), and `pixelsHigh` (page 23).

Available in Mac OS X v10.0 and later.

Declared in `NSImageRep.h`.

Declared In

`NSImageRep.h`

Obsolete Change Notification Name

The following constant maps to the new notification and is for legacy code only.

```
#define NSImageRepRegistryChangedNotification NSImageRepRegistryDidChangeNotification
```

Constants

`NSImageRepRegistryChangedNotification`

An older name for the `NSImageRepRegistryDidChangeNotification` (page 29) notification. Do not use.

Available in Mac OS X v10.0 and later.

Declared in `NSImageRep.h`.

Notifications

NSImageRepRegistryDidChangeNotification

Posted whenever the `NSImageRep` class registry changes.

The notification object is the image class that is registered or unregistered. This notification does not contain a `userInfo` dictionary.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSImageRep.h

Document Revision History

This table describes the changes to *NSImageRep Class Reference*.

Date	Notes
2009-01-06	Added a definition for the <code>NSImageRepRegistryChangedNotification</code> macro.
2007-10-31	Updated for Mac OS X v10.5.
2006-10-03	Updated descriptions for <code>imageFileTypes</code> , <code>imageUnfilteredFileTypes</code> , <code>imagePasteboardTypes</code> , and <code>imageUnfilteredPasteboardTypes</code> .
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

B

bitsPerSample **instance method** [20](#)

C

canInitWithData: **class method** [8](#)
canInitWithPasteboard: **class method** [9](#)
colorSpaceName **instance method** [20](#)

D

Display Device Matching [29](#)
draw **instance method** [21](#)
drawAtPoint: **instance method** [21](#)
drawInRect: **instance method** [22](#)

H

hasAlpha **instance method** [22](#)

I

imageFileTypes **class method** [9](#)
imagePasteboardTypes **class method** [10](#)
imageRepClassForData: **class method** [10](#)
imageRepClassForFileType: **class method** [10](#)
imageRepClassForPasteboardType: **class method** [11](#)
imageRepClassForType: **class method** [11](#)
imageRepsWithContentsOfFile: **class method** [11](#)
imageRepsWithContentsOfURL: **class method** [12](#)
imageRepsWithPasteboard: **class method** [13](#)
imageRepWithContentsOfFile: **class method** [14](#)
imageRepWithContentsOfURL: **class method** [15](#)

imageRepWithPasteboard: **class method** [15](#)
imageTypes **class method** [16](#)
imageUnfilteredFileTypes **class method** [16](#)
imageUnfilteredPasteboardTypes **class method** [17](#)
imageUnfilteredTypes **class method** [18](#)
isOpaque **instance method** [23](#)

N

NSImageRepMatchesDevice **constant** [29](#)
NSImageRepRegistryChangedNotification **constant** [29](#)
NSImageRepRegistryDidChangeNotification **notification** [29](#)

O

Obsolete Change Notification Name [29](#)

P

pixelsHigh **instance method** [23](#)
pixelsWide **instance method** [24](#)

R

registeredImageRepClasses **class method** [18](#)
registerImageRepClass: **class method** [19](#)

S

setAlpha: **instance method** [24](#)
setBitsPerSample: **instance method** [25](#)
setColorSpaceName: **instance method** [25](#)

setOpaque: [instance method 26](#)
setPixelsHigh: [instance method 26](#)
setPixelsWide: [instance method 27](#)
setSize: [instance method 27](#)
size [instance method 28](#)

U

unregisterImageRepClass: [class method 19](#)