
NSImageRep

Inherits From:	NSObject
Conforms To:	NSCoding NSCopying NSObject (NSObject)
Declared In:	AppKit/NSImageRep.h

Class Description

NSImageRep is a semi-abstract 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’s typically used through an NSImage object. An NSImage manages a group of representations, choosing the best one for the current output device.

There are four subclasses defined in the Application Kit:

Subclass	Source Data
NSBitmapImageRep	Tag Image File Format (TIFF), Windows bitmap (BMP) and other bitmap data
NSCachedImageRep	A rendered image, usually in an off-screen window
NSCustomImageRep	A delegated method that can draw the image
NSEPSImageRep	Encapsulated PostScript code (EPS)

You can define other NSImageRep subclasses for objects that render images from other types of source information. New subclasses must be added to the NSImageRep class registry by invoking the **registerImageRepClass:** class method. The NSImageRep subclass informs the registry of the data types it can support through its **imageUnfilteredFileTypes**, **imageUnfilteredPasteboardTypes**, and **canInitWithData:** class methods. Once an NSImageRep subclass is registered, an instance of that subclass is created any time NSImage encounters the type of data handled by that subclass.

Subclasses which deal with file and pasteboard types should implement **imageUnfilteredFileTypes**, **imageUnfilteredPasteboardTypes**, **initWithData:**, **canInitWithData:**, and, if they have the ability to read multiple images from a file, **imageRepsWithData:**. These last three should not do any filtering; all filtering is automatic.

Adopted Protocols

NSCoding	– encodeWithCoder: – initWithCoder:
NSCopying	– copyWithZone:

Method Types

Creating an NSImageRep	+ imageRepsWithContentsOfFile: + imageRepsWithPasteboard: + imageRepWithContentsOfFile: + imageRepWithPasteboard:
Checking data types	+ canInitWithData: + canInitWithPasteboard: + imageFileTypes: + imagePasteboardTypes: + imageUnfilteredFileTypes: + imageUnfilteredPasteboardTypes:
Setting the size of the image	– setSize: – size
Specifying information about the representation	– bitsPerSample – colorSpaceName – hasAlpha – isOpaque – pixelsHigh – pixelsWide – setAlpha: – setBitsPerSample: – setColorSpaceName: – setOpaque: – setPixelsHigh: – setPixelsWide:
Drawing the image	– draw – drawAtPoint: – drawInRect:

Managing NSImageRep subclasses

- + `imageRepClassForData:`
- + `imageRepClassForFileType:`
- + `imageRepClassForPasteboardType:`
- + `registeredImageRepClasses`
- + `registerImageRepClass:`
- + `unregisterImageRepClass:`

Notifications

When the NSImageRep class registry changes
NSImageRepRegistryDidChangeNotification

Class Methods

canInitWithData:

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

Overridden in subclasses to return YES if the receiver can initialize itself from *data*, and NO if it cannot. Note that this method doesn't need to do a comprehensive check; it should return NO only if it knows that the receiver can't initialize itself from *data*.

canInitWithPasteboard:

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

Returns YES if the NSImageRep can handle the data represented by pasteboard, otherwise returns NO.

*This method invokes the **imageUnfilteredPasteboardTypes** 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** method to assure that this method returns the correct response.*

imageFileTypes

+ (NSArray *)**imageFileTypes**

*Returns an array of NSStrings representing all file types supported by NSImageRep or one of its subclasses. The list includes both those types returned by the **imageUnfilteredFileTypes** class method and 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 **imageUnfilteredFileTypes** method.*

imagePasteboardTypes

+ (NSArray *)**imagePasteboardTypes**

Returns an array of NSStrings representing all pasteboard types supported by NSImageRep or one of its subclasses. The list includes both those types returned by the **imageUnfilteredPasteboardTypes** class method and those that can be converted by a user-installed filter service to a supported type. Don't override this method when subclassing NSImageRep—it always returns a valid list for any subclass of NSImageRep that correctly overrides the **imageUnfilteredPasteboardTypes** method.

imageRepClassForData:

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

Returns the NSImageRep subclass that handles data of type *data*, or **Nil** if the NSImage class registry contains no subclasses that handle data of the specified type.

imageRepClassForFileType:

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

Returns the NSImageRep subclass that handles files of type *type*, or **Nil** if the NSImage class registry contains no subclasses that handle files of the specified type.

imageRepClassForPasteboardType:

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

Returns the NSImageRep subclass that handles pasteboard data of type *type*, or **Nil** if the NSImage class registry contains no subclasses that handle pasteboard data of the specified type.

imageRepWithContentsOfFile:

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

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:**) that's initialized with the contents of the file *filename*. If sent to a subclass of NSImageRep that recognizes the type of file specified by *filename*, it returns an instance of that subclass initialized with the contents of the file *filename*.

imageRepWithContentsOfFile: 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 indicated by *filename*.

-
- The message is sent to a subclass of `NSImageRep`, and that subclass doesn't handle data of the type indicated by *filename*.
 - The `NSImageRep` subclass is unable to initialize itself with the contents of *filename*.

filename may be a full or relative pathname, and should include an extension that identifies the data type in the file. By default, the files handled are those with the extensions “tiff”, “tif”, “bmp”, and “eps”.

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

See also: + `imageFileTypes`

`imageRepWithPasteboard:`

+ (id)**`imageRepWithPasteboard:`**(`NSPasteboard *`)*pasteboard*

If sent to the `NSImageRep` class object, this method returns a newly-allocated instance of a subclass of `NSImageRep` that's initialized with the data in *pasteboard*. If sent to a subclass of `NSImageRep` that recognizes the type of data contained in *pasteboard*, it returns an instance of that subclass initialized with the data in *pasteboard*.

`imageRepWithPasteboard:` 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 *pasteboard*.
- The message is sent to a subclass of `NSImageRep`, and that subclass doesn't handle data of the type contained in *pasteboard*.
- The `NSImageRep` subclass is unable to initialize itself with the contents of *pasteboard*.

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

See also: + `imagePasteboardTypes`

`imageRepsWithContentsOfFile:`

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

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:`**) that have been initialized with the contents of the file *filename*. If sent to a subclass of `NSImageRep` that recognizes the type of file specified by *filename*, it returns an array of objects (all instances of that subclass) that have been initialized with the contents of the file *filename*.

`imageRepsWithContentsOfFile:` 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 indicated by *filename*.
- The message is sent to a subclass of NSImageRep, and that subclass doesn't handle data of the type indicated by *filename*.
- The NSImageRep subclass is unable to initialize itself with the contents of *filename*.

filename may be a full or relative pathname, and should include an extension that identifies the data type in the file. By default, the files handled are those with the extensions “tiff”, “tif”, “bmp”, and “eps”.

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

See also: + imageFileTypes

imageRepsWithPasteboard:

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

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 *pasteboard*. If sent to a subclass of NSImageRep that recognizes the type of data contained in *pasteboard*, it returns an array of objects (all instances of that subclass) initialized with the data in *pasteboard*

imageRepsWithPasteboard: 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 *pasteboard*.
- The message is sent to a subclass of NSImageRep, and that subclass doesn't handle data of the type contained in *pasteboard*.
- The NSImageRep subclass is unable to initialize itself with the contents of *pasteboard*.

The NSImageRep subclass is initialized by creating an NSData object based on the data in *pasteboard*, then passing it to **imageRepsWithData:**.

See also: + imagePasteboardTypes

imageUnfilteredFileTypes

+ (NSArray *)**imageUnfilteredFileTypes**

Returns an array of NSStrings representing all file types (extensions) supported by the NSImageRep. By default, the returned array is empty.

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

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

    if (!types) types = [[NSArray alloc]
        initWithObjects:@"tiff", @"tif", @"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.

See also: + `imageFileTypes`, + `imageUnfilteredFileTypes`: (`NSImage`)

imageUnfilteredPasteboardTypes

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

Returns an array representing all pasteboard types supported by the `NSImageRep`. By default, the returned array is empty.

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

See also: + `imagePasteboardTypes`, + `imageUnfilteredPasteboardTypes`: (`NSImage`)

registerImageRepClass:

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

Adds *imageRepClass* to the registry of available `NSImageRep` classes. This method posts the `NSImageRepRegistryChangedNotification` notification, 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.

See also: + **load** (NSObject)

registeredImageRepClasses

+ (NSArray *)**registeredImageRepClasses**

Returns an array containing the registered NSImageRep classes.

unregisterImageRepClass:

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

Removes *imageRepClass* from the registry of available NSImageRep classes. This method posts the NSImageRepRegistryChangedNotification notification, along with the receiving object, to the default notification center.

Instance Methods

bitsPerSample

– (int)**bitsPerSample**

Returns the number of bits used to specify a single pixel in each component of the data.

colorSpaceName

– (NSString *)**colorSpaceName**

Returns the name if the image's color space, or NSCalibratedRGBColorSpace if no name has been assigned.

draw

– (BOOL)**draw**

Implemented by subclasses to draw the image at location (0.0, 0.0) in the current coordinate system. Subclass methods return YES if the image is successfully drawn, and NO if it isn't. This version of the method simply returns YES.

drawAtPoint:

– (BOOL)**drawAtPoint:**(NSPoint)*aPoint*

Sets the current coordinates to those indicated by *aPoint*, invokes the receiver's **draw** method draw the image at that point, then restores the current coordinates to their original setting. If *aPoint* is (0.0, 0.0), **drawAtPoint:** simply invokes **draw**.

This method returns NO without translating, scaling, or drawing if the size of the image has not been set. Otherwise it returns the value returned by the **draw** method, which indicates whether the image is successfully drawn.

See also: – **setSize:**

drawInRect:

– (BOOL)**drawInRect:**(NSRect)*rect*

Draws the image so that it fits inside the rectangle referred to by *rect*. The current coordinates are set to the point specified in the rectangle and are scaled so the image will fit within the rectangle. The receiver's **draw** method is then invoked to draw the image. After draw has been invoked, the current coordinates and scale factors are restored to their original settings.

This method returns NO without translating, scaling, or drawing if the size of the image has not been set. Otherwise it returns the value returned by the **draw** method, which indicates whether the image is successfully drawn.

See also: – **setSize:**

hasAlpha

– (BOOL)**hasAlpha**

Returns YES if the receiver has been informed that the image has a coverage component (alpha), and NO if not.

isOpaque

– (BOOL)**isOpaque**

Returns YES if the receiver is opaque; NO otherwise. Use this method to test whether an NSImageRep completely covers the area within the rectangle returned by `size:`. Use the method `setOpaque:` to set the value returned by this method.

pixelsHigh

– (int)**pixelsHigh**

Returns the height of the image in pixels, as specified in the image data.

See also: – size:

pixelsWide

– (int)**pixelsWide**

Returns the width of the image in pixels, as specified in the image data.

See also: – size:

setAlpha:

– (void)**setAlpha:**(BOOL)*flag*

Informs the NSImageRep whether the image has an alpha component. *flag* should be YES if it does, and NO if it doesn't.

setBitsPerSample:

– (void)**setBitsPerSample:**(int)*anInt*

Informs the NSImageRep that the image has *anInt* bits of data for each pixel in each component.

setColorSpaceName:

– (void)**setColorSpaceName:**(NSString *)*string*

Informs the receiver of the image's color space. By default, an NSImageRep'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

setOpaque:

– (void)**setOpaque:(BOOL)***flag*

Sets opacity of the NSImageRep’s image. If flag is YES, the image is opaque.

setPixelsHigh:

– (void)**setPixelsHigh:(int)***anInt*

Informs the NSImageRep that the data specifies an image *anInt* pixels high.

See also: – setSize:

setPixelsWide:

– (void)**setPixelsWide:(int)***anInt*

Informs the NSImageRep that the data specifies an image *anInt* pixels wide.

See also: – setSize:

setSize:

– (void)**setSize:(NSSize)***aSize*

Sets the size of the image in units of the base coordinate system. This determines the size of the image when it’s rendered; it’s not necessarily the same as the width and height of the image in pixels as specified in the image data. You must set the image size before you can render it.

See also: – draw, – setPixelsHigh:, – setPixelsWide:

size

– (NSSize)**size**

Returns the size of the image in units of the base coordinate system. This is the size of the image when it’s rendered; it’s not necessarily the same as the width and height of the image in pixels as specified in the image data.

See also: – pixelsHigh:, – pixelsWide:

Notifications

NSImageRep declares and posts the following notification. In addition, it posts notifications that are declared by its superclass, NSObject. See the NSObject class specification for more information.

NSImageRepRegistryDidChangeNotification

Notification Object The notifying NSImageRep.

Userinfo none

Posted when the NSImageRep class registry changes, either through the use of **registerImageRepClass:** or **unregisterImageRepClass:**.