

Freeing an NXImage object

`initWithPasteboard:`

`copyFromZone:`

the file name on the pasteboard, either directly or through a user-installed filter service. If no registered filter handles data of type, returns nil.

If you create a subclass of `NXImageRep` that supports other pasteboard types, you must override the method `imageUnfilteredPasteboardTypes` to return the extensions representing those pasteboard types. Your subclass using `NXImage`'s class method `registerImageRep:`.

`imagePasteboardTypes`, `imageUnfilteredPasteboardTypes` (`NXImageRep` class)

`imageRepForStream:`

+ (Class)imageRepForStream:(NXStream *)stream

Returns the `NXImageRep` subclass that can be instantiated from the data in stream. By default, this method returns `NXBitmapImageRep` for a stream containing TIFF data, `NXEPSImageRep` for a stream containing EPS data, and `NXImageRep` for a stream containing an unsupported data type. stream must be seekable.

If you create a subclass of `NXImageRep` that supports other data types, you must override the `NXImageRep` class method `canLoadFromStream:` to determine whether your subclass can be instantiated from data in stream. You must also register your subclass using `NXImage`'s `registerImageRep:` class method.

`canLoadFromStream:` (`NXImageRep`)

`imageUnfilteredFileTypes`

+ (const char *const *)imageUnfilteredFileTypes

This method returns a NULL-terminated list of file types which all the image reps registered with `NXImage` can load from. This list belongs to the `NXImage` and should not be freed or changed:

`imageUnfilteredPasteboardTypes`

+ (const NXAtom *)imageUnfilteredPasteboardTypes

This method returns a NULL-terminated list of pasteboard types which all the image reps registered with `NXImage` can load from. This list belongs to the `NXImage` and should not be freed or changed:

`registerImageRep:`

+ (void)registerImageRep:imageRepClass

Notifies `NXImage` of the existence of a subclass of `NXImageRep`.

This method adds the class to `NXImage`'s list of image reps without verifying the behavior of image reps. When an application calls `NXImage` class methods such as `imageFileTypes`, `imagePasteboardTypes`, and `imageRepForStream:`, the class checks this list to find the types of image data it can support. `imageRepClass` must implement `imageUnfilteredFileTypes`, `imageUnfilteredPasteboardTypes`, and `canLoadFromStream:` to inform `NXImage` of the types it supports.

This method may be invoked multiple times `imageRepClass` is added to `NXImage`'s list of image reps. `registerImageRep:` is invoked.

copyFromZone:(NXZone *)zone

Returns a new instance of NXImage that's an exact copy of the receiver. Memory for the new instance is allocated in the specified zone. Cached image reps are copied fully, the new NXImage has its own copy of the image data. Image reps are copied with only source information (for example, file names) the image can be recreated. When the NXImage is asked to composite itself.

copyFromZone: (NXCachedImageRep)

setSize:

init

loadFromFile:, loadFromFile:, + imageRepForFileType:, + registerImageRep

initWithImage:rect:

initWithPasteboard:(Pasteboard *)pasteboard

Initializes and returns the receiver, a newly allocated NXImage instance, from pasteboard. pasteboard must be a Pasteboard object returned by one of the registered NXImageRep's imageUnfilteredPasteboardTypes methods the default types supported are NXPostscriptPboardType (NXEPSImageRep) and NXTIFFPboardType (NXBitmapImageRep). If NXFilenamePboardType, the file name should have an extension returned by one of the registered imageUnfilteredFileTypes methods the default types supported are "tiff", "tif", (NXBitmapImageRep) and "eps" (NXEPSImageRep).

If the data type on the pasteboard isn't supported by a registered NXImageRep, this method frees the pasteboard and returns nil. Otherwise this method invokes initWithStream to initialize the image.

(BOOL)loadFromFile:(const char *)filename

Creates an image representation from the data read from filename and adds it to the receiving NXImageRep representations. This method is equivalent to mapping the file to memory, then invoking loadFromData:withName:.

filename may be a full or relative pathname, and should include an extension that identifies the data type. This method looks for an NXImageRep subclass that handles that data type from among those registered with the system. By default, the files handled are those with the extensions "tiff", "tif", and "eps". If the file name has an extension that is one not recognized by one of the registered subclasses of NXImageRep, this method will attempt to load the file and attempts to find a registered subclass of NXImageRep that can handle the data in the stream. If the TIFF or EPS data includes more than one image, a separate representation is created for each one.

If the NXImage object doesn't retain image data (isDataRetained returns NO), the image will be represented in the screen window and the representations will be of type NXCachedImageRep. If the data is retained in memory, the representations will be of type NXBitmapImageRep or NXEPSImageRep, depending on the data.

If successful in creating at least one representation, this method returns YES. If not, it returns NO.

