Q: In my application I am reading in an NXImage. A nil is never returned, even if I read in a bogus file. Is this a bug? Here is my code:

```
id myNXImage = [NXImage alloc];
if ([myNXImage initFromFile: "dummyName.tiff"] == nil)
{
        /* this is never getting called! */
        fprintf(stderr,"dummyName.tiff doesn't exist!\n");
}
```

A: This is not a bug. The **initFromFile:** method is lazy and does not catch *all* the errors that might happen when loading an image. Your application should be prepared to check for errors later on down the line either through delegation or by checking the **composite:** or **lockFocus** return values. If you wish, you can force the image to be rendered immediately:

```
id myNXImage = [[NXImage alloc] initFromFile: filename];
if ([myNXImage lockFocus])
        [image unlockFocus];
else
        fprintf(stderr,"%s doesn't exist\n", filename);
```

Although this behavior might seem confusing it allows for more optimal performance: the image isn't rendered into the cache until it is needed. Rendering a large or complex file can be slowÐparticularly for a complex EPS file.

Note: Another good approach for determining whether an image can be successfully rendered is the NXImage delegate method **imageDidNotDraw:inRect:**. If you have assigned a delegate for the image and implemented this method, it gets called when compositing fails for whatever reason. See the documentation on NXImage for more information about this method. Also note that this method of delegation may be the only way to catch a drawing error for an image which is being "handed" to the AppKitÐan icon on a button, for example.

There is a known bug in Release 2 where **imageDidNotDraw:inRect:** fails to be called when encountering an error from within the method **composite:toPoint:**. This bug can be avoided by using the NXImage method **composite:fromRect:toPoint:**. This bug has been fixed in Release 3.

QA730

Valid for 2.0, 3.0