Q: I have allocated an instance of NXImage and an instance of NXBitmapImageRep. I then tell the NXImage to use the rep instance, like this:

```
NXRect originalSize;
id myRep, myImage;
int bitsPerPixel;

myRep = [[NXBitmapImageRep alloc] initFromFile: filename];
[myRep getSize: &originalSize];
myImage = [[NXImage alloc] initSize: &originalSize];
[myImage useRepresentation: myRep];
```

Then, later in my application I query the rep instance (as follows) and the query fails because myRep is nil! Why is this?

```
bitsPerPixel = [myRep bitsPerPixel]; /* this fails -- myRep is nil ! */
```

A: This is not a bug. Once you have ªgivenº the NXBitmapImageRep instance to NXImage (by calling useRepresentation:) then the NXImage "owns" that rep and can do what it wishes with it. (This is also true for any class of rep instance, not just NXBitmapImageRep) What the NXimage typically does is to turn

that representation into an NXCachedImageRep and then free the NXBitmapImageRep. To prevent this behavior do a setDataRetained:YES on the NXImage instance. The setDataRetained: method defaults to NO. The NXImage then does not free the NXBitmapImageRep. For example, to correct the above example, add the following line prior to calling useRepresentation:

```
[myImage setDataRetained:YES];
```

QA732

Valid for 2.0, 3.0