Q: How can my program tell whether it is running on a system with a color screen?

A: The PostScript graphics model allows you to do all your rendering in 24 bit color, and have the WindowServer display it in the best way possible (dithering down to 2 bits, if necessary). If there are situations in which you need to use the "inquire and adapt" approach, the AppKit provides you with a method in the Application class called **colorScreen** that returns the most colorful screen available to your application. By examining this data structure, you can determine how deep your color screen is. Below is a code fragment that does this:

```
#import <appkit/Application.h>
#import <appkit/screens.h>
...
{
       const NXScreen *deepestScreen;

       deepestScreen=[NXApp colorScreen];

       switch ( deepestScreen->depth ) {

              case NX_TwelveBitRGBDepth :
                     /* Twelve Bit Color */
```

```
        break;

case NX_TwentyFourBitRGBDepth :
        /* True Color */
        break;

case NX_TwoBitGrayDepth :
default :
        /* Greyscale */
        break;

}
```

```
        }
```

Q: How can I tell whether a View can display color or not?

A: The previous question does not answer which kinds of rendering commands you should be sending for a given View. There are two headed systems, and other circumstances where the window you are drawing in is not based on the value of **colorScreen**. The most common thing you want to do is different rendering based on whether this View can display color or not. One way to do this is to use the

shouldDrawColor method of View. Do the following in your drawSelf:: method for the View:

```
- drawSelf:(NXRect *)rects :(int)count
{
        if ([self shouldDrawColor]) {
        ... color-specific drawing ...
        } else {
                ...
        }
        ...
}
```

If you need to know the actual depth of a given window, use the **depthLimit** method of Window. Here's some code for that:

```
NXWindowDepthLimit  depth;
if ((depth = [myWindow depthLimit]) == NX_DefaultDepth) {
       depth = [Window defaultDepthLimit];
}
```

QA700

Valid for 2.0, 3.0