

Adding an icon

The final change is to add an icon to the button to make things a little more festive. FOX supports all of the popular image file formats (e.g. BMP, GIF, JPEG, PNG and TIFF) and you can use any of them as icons on buttons and labels. For this example, we'll use the one of the "Powered By Ruby" images created by Hal Fulton (and posted at the [Ruby Garden Wiki](#)):

```
require 'fox16'

include Fox

theApp = FXApp.new

theMainWindow = FXMainWindow.new(theApp, "Hello")

theButton = FXButton.new(theMainWindow, "Hello, World!")
theButton.tipText = "Push Me!"
iconFile = File.open("pbr.jpg", "rb")
theButton.icon = FXJPEGIcon.new(theApp, iconFile.read)
iconFile.close
theButton.connect(SEL_COMMAND) { exit }

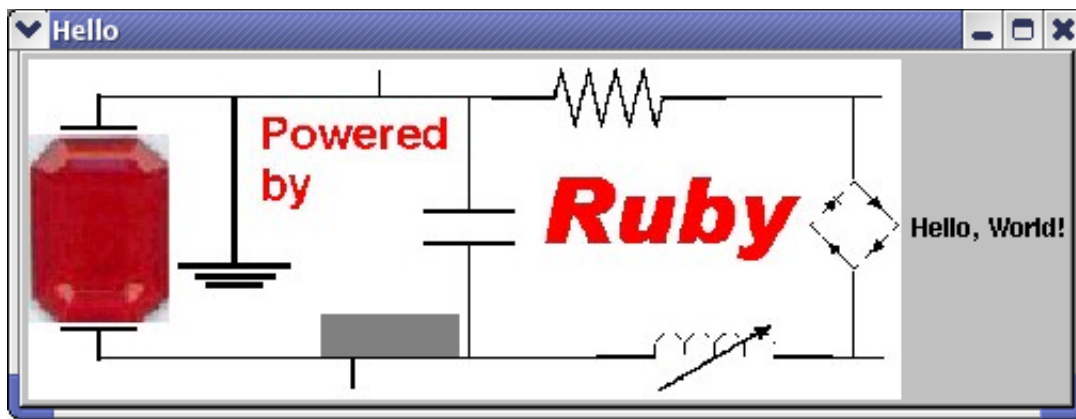
FXToolTip.new(theApp)

theApp.create

theMainWindow.show

theApp.run
```

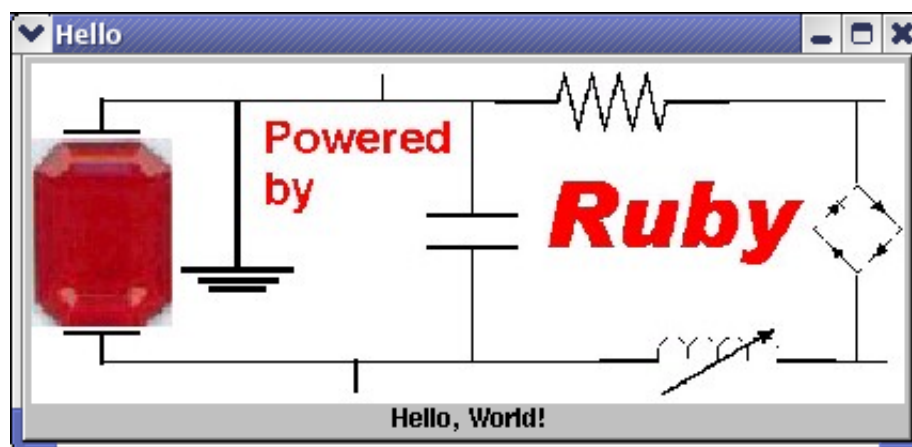
Here, *pbr.jpg* is the file name of the JPEG image file. You want to be sure to open the file in *binary* mode (i.e. including the "b" mode flag), because there is a difference on the Windows platform. Since it's a JPEG image, we need to use the `FXJPEGIcon` class to instantiate this icon. The first argument to `FXJPEGIcon.new` is just a reference to the `FXApp` instance, and the second argument is the contents of the image file. We associate this icon object with our button using the button's `icon` accessor method. If you run this example, you should see:



When you have both text and an icon displayed on a button (or its superclass, `FXLabel`) the default positioning is to display the icon to the left of the text. For this particular example, however, it would probably be more appropriate to display the icon *above* the text. We can achieve this using the button's `iconPosition` accessor method:

```
theButton.iconPosition = ICON_ABOVE_TEXT
```

If you re-run the program after adding this line, you should see:



The last change we're going to make is to make the icon transparent. FOX allows you to specify that some regions of an icon should be treated as "transparent", meaning that whatever's underneath them shows through. FOX distinguishes those transparent regions from the non-transparent ones using a transparency color, and any pixels in the original image that have that color become transparent. In most cases, FOX can determine this transparency color automatically (indeed, for image file formats like GIF it's part of the image information). You can also specify the transparency color explicitly if you like.

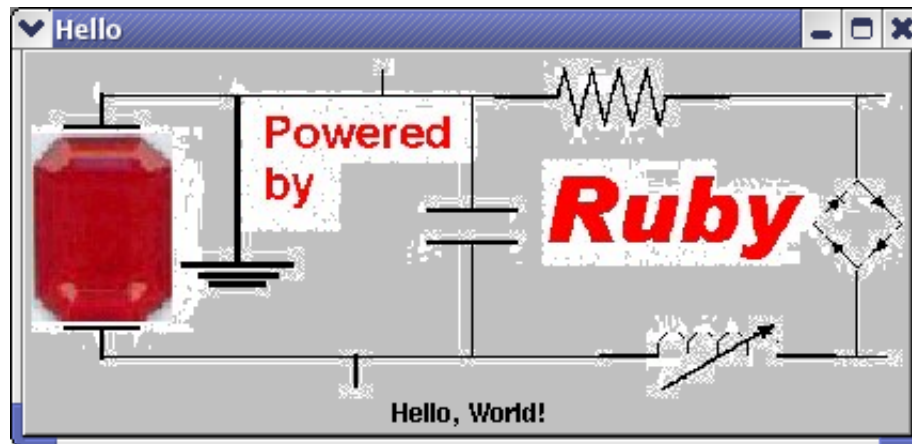
For the icon we've chosen, it's pretty obvious that the transparency color is white, but let's let FOX figure that out for us. We want to activate two options for the icon:

- the `IMAGE_ALPHACOLOR` option, which tells FOX that some regions of this image should be treated as transparent; and,
- the `IMAGE_ALPHAGUESS` option, which tells FOX to guess the appropriate transparency color using the colors used in the four corners of the image.

To set these options, add this line to the program:

```
theButton.icon.options = IMAGE_ALPHACOLOR | IMAGE_ALPHAGUESS
```

and then re-run the program after making this change to see the final result:



[Prev](#)

Adding a tool tip

[Up](#)

[Home](#)

[Next](#)

Chapter 4. Working With the
Clipboard