Chapter 6. Unicode and FXRuby

Table of Contents

Beginning with version 1.6, FOX and FXRuby provide support for the display of Unicode strings in FOX widgets. For some excellent discussion about how to use Unicode in Ruby, I recommend Patrick Hall's article, "Ruby and Unicode" and why the lucky stiff's follow-up article, "Closing in on Unicode with Jcode". Here, we're going to make use of the ideas in those articles to give a quick demonstration of how to use FXRuby's support for Unicode.

Basic Application

Here's the original version of our "Hello, World!" program:

```
require 'fox16'

include Fox

application = FXApp.new("Hello", "FoxTest")
main = FXMainWindow.new(application, "Hello", nil, nil, DECOR_ALL)
FXButton.new(main, "&Hello, World!", nil, application, FXApp::ID_QUIT)
application.create()
main.show(PLACEMENT_SCREEN)
application.run()
```

and here's the modified version:

```
require 'fox16'
require 'jcode'

$KCODE = 'u'

class UString < String
  # Show u-prefix as in Python
  def inspect; "u#{ super }" end

  # Count multibyte characters
  def length; self.scan(/./).length end

  # Reverse the string
  def reverse; self.scan(/./).reverse.join end
end
```

```ruby
module Kernel
  def u( str )
    UString.new str.gsub(/U\+([0-9a-fA-F]{4,4})/u){["#$1".hex ].pack('U*')}
  end
end


include Fox


question = u'U+00bfHabla espaU+00f1ol?'


application = FXApp.new("Hello", "FoxTest")
main = FXMainWindow.new(application, "Hello", nil, nil, DECOR_ALL)
FXButton.new(main, question, nil, application, FXApp::ID_QUIT)
application.create()
main.show(PLACEMENT_SCREEN)
application.run()
```

The **jcode** library (part of the standard Ruby library) provides a number of extensions to Ruby's String class, to ensure that its methods work properly for non-ASCII character encodings. By setting the $KCODE global variable to "u", we're telling Ruby which character encoding it is that we're using (UTF-8).