Sending Data to the Clipboard

Whenever some other window requests the clipboard's contents (e.g. as a result of a "paste" operation) FOX will send a `SEL_CLIPBOARD_REQUEST` message to the current clipboard owner. Remember, the clipboard owner is the window that called `acquireClipboard()`. For our example, the main window is acting as the clipboard owner and so it needs to handle the `SEL_CLIPBOARD_REQUEST` message:

```
# Handle clipboard request
self.connect(SEL_CLIPBOARD_REQUEST) do
  setDNDData(FROM_CLIPBOARD, FXWindow.stringType, Fox.fxencodeStringData(@clippedCustomer.to_s))
end
```

The `setDNDData()` method takes three arguments. The first argument tells FOX which kind of data transfer we're trying to accomplish; as it turns out, this method can be used for drag-and-drop (`FROM_DRAGNDROP`) and X11 selection (`FROM_SELECTION`) data transfer as well. The second argument to `setDNDData()` is the drag type for the data and the last argument is the data itself, a binary string.

If you're wondering why we need to call the `fxencodeStringData()` module method to preprocess the string returned by the call to `Customer#to_s`, that's a reasonable thing to wonder about. In order for FOX to play nice with other clipboard-aware applications, it must be able to store string data on the clipboard in the format expected by those applications. Unfortunately, that expected format is platform-dependent and does not always correspond directly to the format that Ruby uses internally to store its string data. The `fxencodeStringData()` method (and the corresponding `fxdecodeStringData()` method) provide you with a platform-independent way of sending (or receiving) string data with the `stringType` drag type.

If you run the program as it currently stands, you should now be able to select a customer from the list, click the "Copy" button and then paste the selected customer data (as a string) into some other application. For example, if you're trying this tutorial on a Windows machine, try pasting into a copy of Notepad or Microsoft Word. The pasted text should look something like:

```
#<struct Struct::Customer name="Joe Smith", address="123 Maple, Anytown, NC", zip=12345>
```