

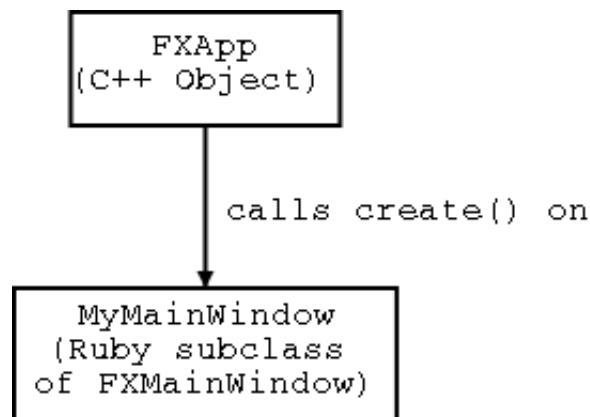
Virtual Functions

One of the design requirements for FXRuby was to ensure that any virtual function call made on a FOX object (from the C++ library layer) is routed to the proper Ruby instance method, even if that method has been overridden in a derived class.

For example, many of the FXRuby example programs are structured with a main window class that subclasses `FXMainWindow` and then overrides its `create` instance method:

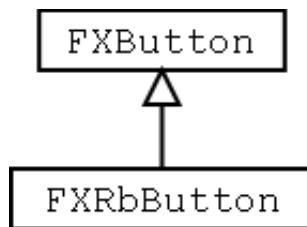
```
class MyMainWindow < Fox::FXMainWindow
  # overrides the base class version, FXMainWindow#create
  def create
    end
end
```

This `create` method isn't called directly from Ruby code, however; it's called as a side effect of calling `FXApp#create`. Inside the C++ FOX library, the `FXApp::create()` member function recursively calls `create()` on all of the application's windows and eventually calls the virtual `FXMainWindow::create()` member function as well:



To ensure that our main window's overridden `create` method is invoked instead of the base class implementation, we need some special machinery in place.

For every C++ class that declares virtual member functions, we derive a new C++ class that overrides all of those virtual functions with special stubs that know how to make method calls on Ruby instances. The naming convention for these classes is that the "FX" prefix is replaced with "FXRb"; for example, our `FXRbButton` C++ class is derived from FOX's `FXButton`:



Although the implementation of these "stubs" varies from function to function, the basic requirements are always the same:

1. Look up the Ruby object associated with this C++ object.
2. Convert each of the function's arguments to their Ruby representation. For example, arguments of type `FXint` are converted to Ruby `Fixnums`.
3. Call the Ruby object's method, using the `rb_funcall()` function from Ruby's C API, and capture its result (which is itself a Ruby object).
4. Convert the method call's result back to the virtual function's C++ return type, and then return that result from the C++ function. For example, if the C++ function has a `FXbool` return type, we would expect the corresponding Ruby method to return either `Qtrue` or `Qfalse`.

[Prev](#)

Object Life Cycles and Garbage
Collection

[Up](#)

[Home](#)

[Next](#)

Appendix F. Getting the Sources
from GitHub