ObjCandCandC++ ReadMe.txt

This tutorial is about calling C++ (and C) from Obj/C (very easy)
And about calling Obj/C from C++ (not difficult)

There really isn't anything very interesting to say about this matter.
However if you want to do this, I think you'll find this rather helpful.

Calling C++ and C is very simple. Nothing special is required.

When you compile and .m file, it knows about C and Obj/C
.mm file, it knows about C, Obj/C and C++
.c file, it knows about C
.cpp file, C++

You can really mix them altogether.

You can of course tell the compiler to treat .m (or .c or anything really) as an Obj/C, C++ file and you'll be fine.

Calling Obj/C from C and C++

```
#include <objc/message.h>

SEL sel = NSSelectorFromString(@"myMethod:")
objc_msgSend(object,sel,arg, ...) ;
```

If your method has muliple inputs, no problem:

```
SEL sel2 = NSSelectorFromString(@"myMethod:withArgument:andSomethingElse:);
objc_msgSend(object,sel2,argument1,argument2,argument3);
```

There's a lot more information at this address:
http://developer.apple.com/library/ios/#documentation/Cocoa/Reference/ObjCRuntimeRef/Reference/reference.html

How does it work?

The Obj/C compiler converts messages into objc_msgSend calls. I think it has an inside track on getting the selector without having to call NSSelectorFromString, maybe not. It doesn't matter.

This is all very like the IDispatch method in COM. At run time, you have to know the object you want to call and it's selector (un unsigned integer) and obcj_sendMessage will do the business. Presumably he maintains a sparse array of the addresses of the methods and jumps through there to call the object's method.

For sure you can extend the dispatch table of every member of a class using the Category feature of Obj/C. I'm not so sure that you can add a method to an object at run-time. So I suspect that all members of a class have the same methods. COM (and JavaScript) are more flexible because they can add methods to objects individually (as well as 'static' class-wide methods).

In reality, I don't know if adding methods to objects individually is incredibly useful - however I have encountered a use case. When I worked on the E4X implementation in ExtendScript, the XML object is required to expose properties based on the content of the XML. So although there are class-wide member function, individual objects expose the properites of the XML hidden inside. Very nice. Very elegant.

If you know how to achieve this with Obj/C, I'll be delighted to learn how. I'll even update this tutorial and acknowledge your contribution.