Javascript can be written to directly call into a Java Applet’s methods ([example here](http://docs.oracle.com/javase/tutorial/deployment/applet/invokingAppletMethodsFromJavaScript.html))

 and, with a little love, a Java Applet can execute Javascript code in the context of the browser ([example here](http://docs.oracle.com/javase/6/docs/technotes/guides/plugin/developer_guide/java_js.html)).

To control JavaScript from Java, LiveConnect uses the netscape.javascript.JSObject class.  All the interactions with JavaScript are accomplished through this class.  Note that because all JavaScript objects must appear in a hierarchical manner in the browser window, the class must also appear in the hierarchy.

The class has the following methods:

* call : calls a JavaScript method
* eval : evaluates a JavaScript expression
* getMember : retrieves a named member of a JavaScript object
* getSlot: retrieves an indexed member of a JavaScript object
* setMember : sets a named member of a JavaScript object
* setSlot: sets an indexed member of a JavaScript object
* toString : converts a JSObject to a string
* removeMember : removes a named member of a JavaScript object

The class also has a static method:

* getWindow : gets a JSObject for the window containing the given applet

To see how to use any of these methods, let's take the call method as an example.  This method does the same as the statement this.methodName(args[0], args[1], ...) in JavaScript. To use the method, you would follow the following instruction in Java:

public Object call(String methodName, Object args[])

**The Mysterious plugin.jar**

When I first discovered the JSObject I could not find where it lived! Buried in the [example](http://docs.oracle.com/javase/6/docs/technotes/guides/plugin/developer_guide/java_js.html) document is the hint that it belongs to a JAR called “plugin.” Now, where does this bad boy live? It can be found within the “lib” folder of your JRE installation. For me it was: C:\Program Files\Java\jdk1.6.0\_32\jre\lib. If you’re using Eclipse, like me, it does not add it automatically to your Applet project. Make sure you add the external reference to the JAR in your build path. You should be all set afterwards.

s