

Exploring Rhino

Introduction

Rhino is a JavaScript interpreter written in Java and developed under the banner of the Mozilla organization. Rhino has some interesting features. Prime among these is the ability to have Java and JavaScript interact: each can access variables and call methods written in the other language. This offers a lot of potential for making otherwise 'closed' applications scriptable with minimal effort.

Setting Up

This exercise involves both server-side and client-side software.

Installing the Software

There are a number of prerequisite pieces of software that need to be installed to establish a development framework for this exercise.

You will require the following:

- Sun JDK 1.3.1
- Rhino 1.5 Release 2

You have been given these on your CD-ROM.

Configuration

A bit of housekeeping is needed before development can begin.

Install the JDK

If you have not already done so, you should install the Sun Java Development Kit (this should be provided to you on your course CD-ROM). Note where it is installed.

Create and Populate the Rhino Home Directory

Open a new Command Prompt window and issue the following command to copy the directory Z:\Exercises\9 Rhino\Rhino to C:\Rhino:

```
C:\> xcopy /ieq "Z:\Exercises\9 Rhino" C:\Rhino
```

Notes:

- This assumes that your CD-ROM device is Z:, you should use your real device letter instead
- Do not simply drag&drop from the CD-ROM; if you do, you will end up with a read-only directory structure and this will complicate things later on

You should end up with the following directory structure on your workstation's hard disk:

```
C:\Rhino
  bin
  Rhino
  Rhino1_5R2
```

Edit Utility Script

You have been provided with a useful utility script to make your life easier later on. Using a text editor such as notepad, edit the supplied **C:\Rhino\bin\setup.bat**¹ file as follows: (for *path-to-jdk* use the place where you installed the Java Development Kit earlier):

```
@echo off
set JAVA_HOME=path-to-jdk
set PATH=%PATH%;%JAVA_HOME%\bin
set CLASSPATH=C:\Rhino\rhino1_5R2\js.jar;%CLASSPATH%
```

A Simple Scripted Application

This example will show how simple Rhino is to use.

Create a Java Application

This application will act as a simple framework for JavaScript scripts.

Edit the skeleton file **C:\Rhino\Rhino\TwoWorlds.java** as shown below:

```
import org.mozilla.javascript.*;
import java.io.*;

public class TwoWorlds
{
    public static void main(String args[])
        throws Exception
    {
        Context cx = Context.enter();
        try
        {
            Scriptable scope = cx.initStandardObjects(null);

            String java_var = args[0],
                parameter = args[1];

            BufferedReader r = new BufferedReader(new FileReader(args[2]));
            String js = r.readLine();
```

¹ If you are using notepad, be aware that notepad will append a '.txt' extension to anything it saves. To stop this behaviour you should enclose the full desired filename in double quote marks when you type it into notepad's "Save As..." dialog box.

```

// create a JavaScript variable called java_var with value
// taken from the Java version of java_var. This will be in
// scope of the function just read in from a file
scope.put("java_var", scope, java_var);

// Now evaluate the string we've collected. We'll ignore the result.
cx.evaluateString(scope, js, "<cmd>", 1, null);

// Print the value of variable "java_var"
Object x = scope.get("java_var", scope);
if (x == Scriptable.NOT_FOUND)
    System.out.println("java_var is not defined.");
else
    System.out.println("java_var = " + Context.toString(x));

// Call function "f(parameter)" and print its result.
Object fObj = scope.get("f", scope);
if (!(fObj instanceof Function))
    System.out.println("f is undefined or not a function.");
else
{
    Object functionArgs[] = { parameter };
    Function f = (Function)fObj;
    Object result = f.call(cx, scope, scope, functionArgs);
    System.out.println("f('" + parameter + "') = " +
        Context.toString(result));
}
}
finally
{
    Context.exit();
}
}
}

```

Points to Note

- This is *not* an "industrial strength" application! It does, however, show the concepts and mechanisms involved.

Create a JavaScript Function

Edit the supplied skeleton file to become:

```
function f(p) { return (p*p) + (java_var*1.0); }
```

Points to Note

- As well as accessing a parameter passed in from Java and returning a value to Java, the script accesses a JavaScript global variable created 'on-the-fly' by the Java framework application.

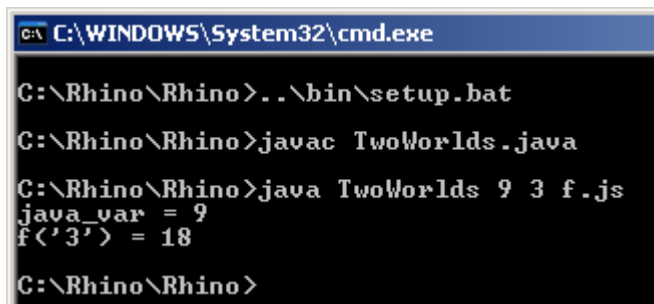
- Note the way the value `java_var` is multiplied by 1.0: this ensures that it takes part in the overall expression as a numeric type rather than as a string

Build and Execute the Application

Open a Command Prompt and execute the following sequence of commands:

```
C:\> cd Rhino\Rhino
C:\Rhino\Rhino> ..\bin\setpath.bat
C:\Rhino\Rhino> javac TwoWorlds.java
C:\Rhino\Rhino> java TwoWorlds 9 3 f.js
```

You should see a result similar to that shown in the screendump below:



```
C:\WINDOWS\System32\cmd.exe
C:\Rhino\Rhino>..\bin\setup.bat
C:\Rhino\Rhino>javac TwoWorlds.java
C:\Rhino\Rhino>java TwoWorlds 9 3 f.js
java_var = 9
f('3') = 18
C:\Rhino\Rhino>
```

Optional Activity

Edit the `f.js` file to become:

```
function f(p) { return (p*p) + java_var; }
```

Re-execute the Java application as before. You should see a different result caused by JavaScript interpreting the `+` operator as string concatenation, rather than as numeric addition.

Another Interesting Feature

Rhino includes a feature known as 'LiveConnect' that makes it easy to use Java APIs directly from within JavaScript. A simple but fairly impressive (in terms of the potential expressed in the demonstration) demonstration whereby JavaScript is used to create a Java 'Swing' GUI-based application can be accessed with the following command:

```
C:\Rhino\Rhino> java org.mozilla.javascript.tools.shell.Main
    ..\rhinol_5R2\examples\SwingApplication.js
```

Note that this command should be given on a single line.

See Also

The definitive Rhino site is <http://www.mozilla.org/rhino>.