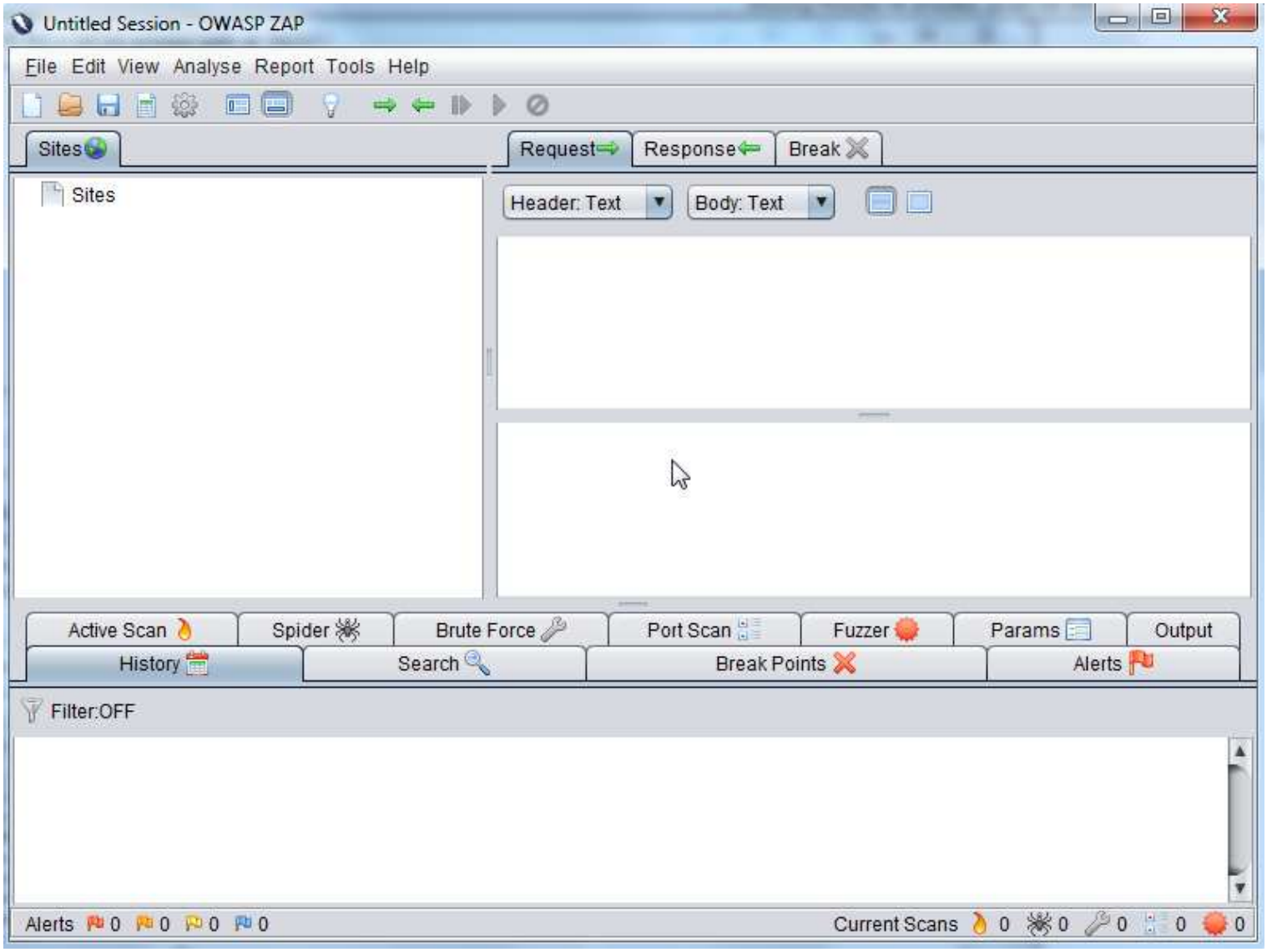
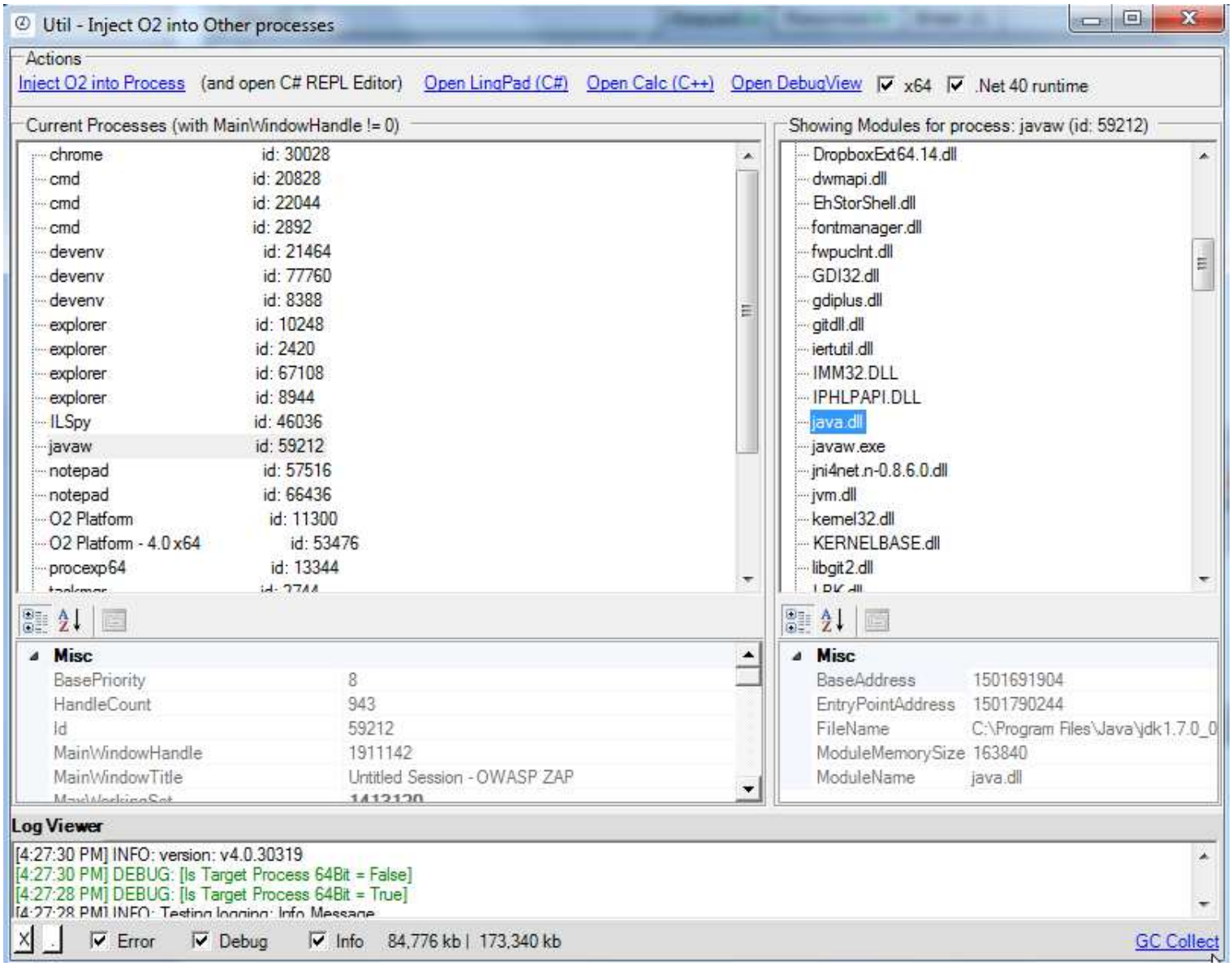
Using Jni4Net (Part 1) - To C# REPL a java process (ZAP Proxy)

Injecting O2 into a Java Process

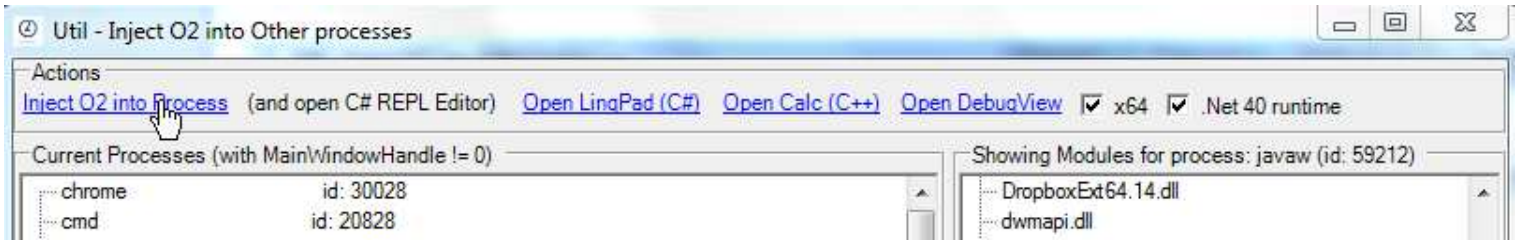
First open ZAP (a java application):



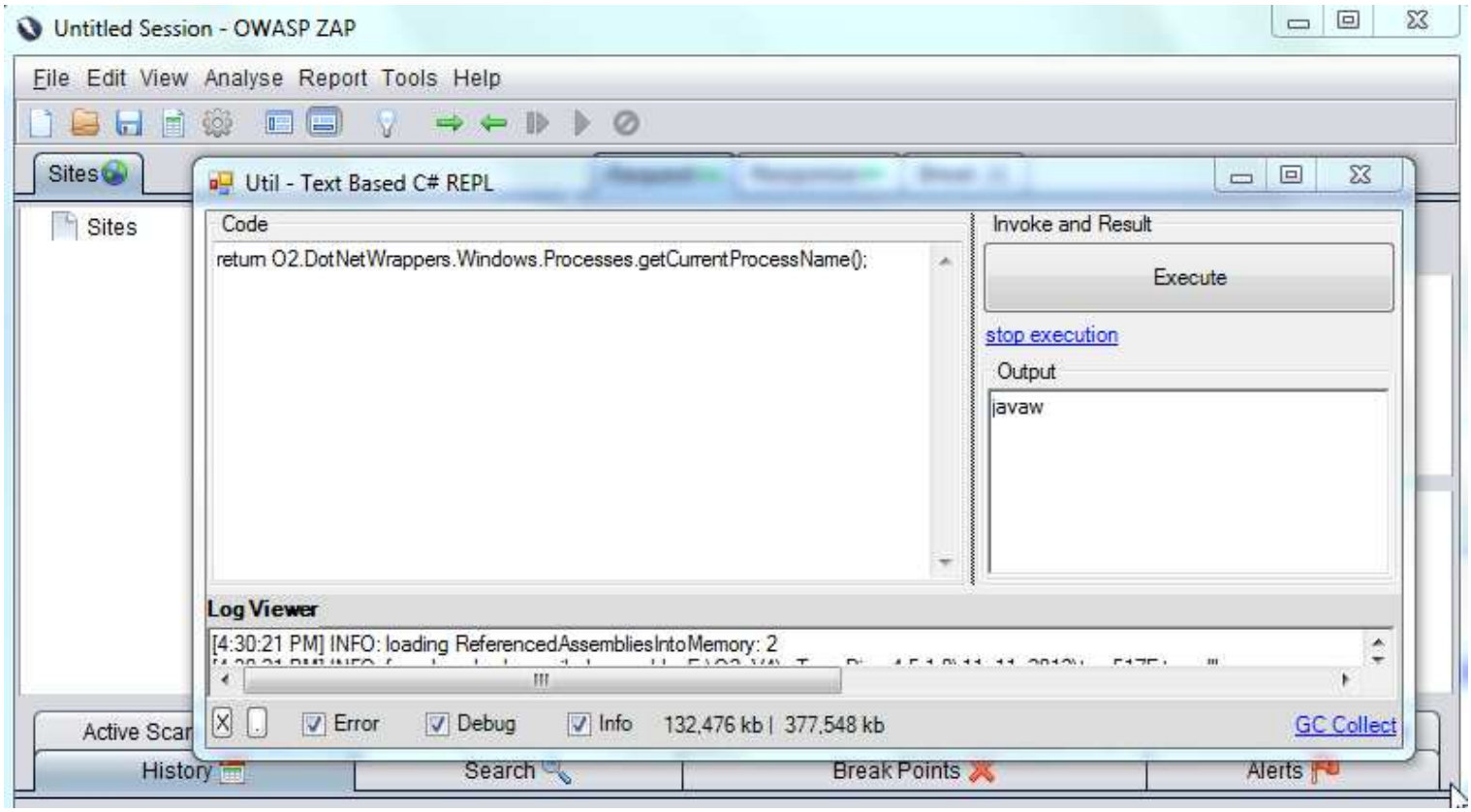
Then on the '***Util - Inject O2 into Other processes***' tool chose the javaw process:



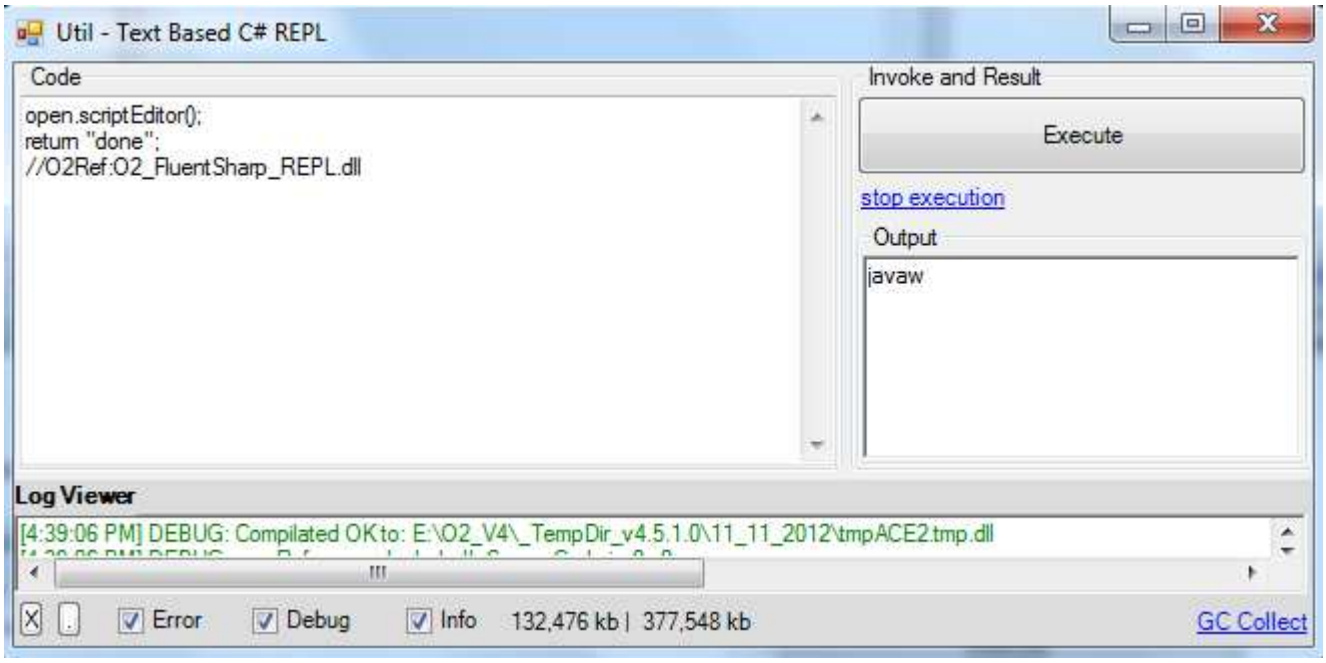
use the ***Inject o2 into Process*** to trigger the injection (not the use of x64 and .net 4.0 runtime options)



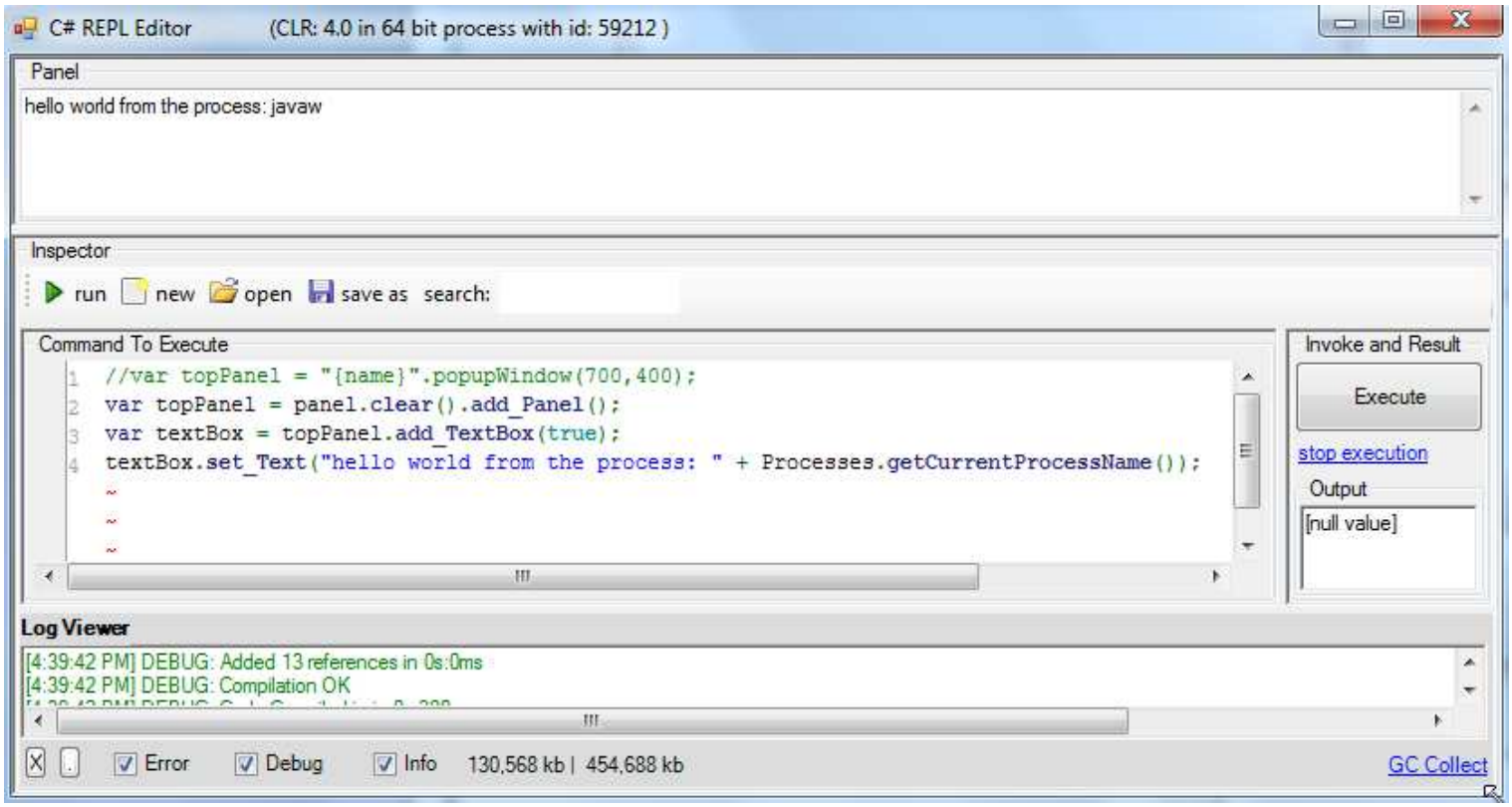
after injection a simple ***'Text Based C# REPL'*** Gui should popup (which is now running under the same process as ZAP)



to ease with scripting it is better to open the full REPL editor (from the O2_FluentSharp_REPL.dll)



which will also be running under the ZAP process

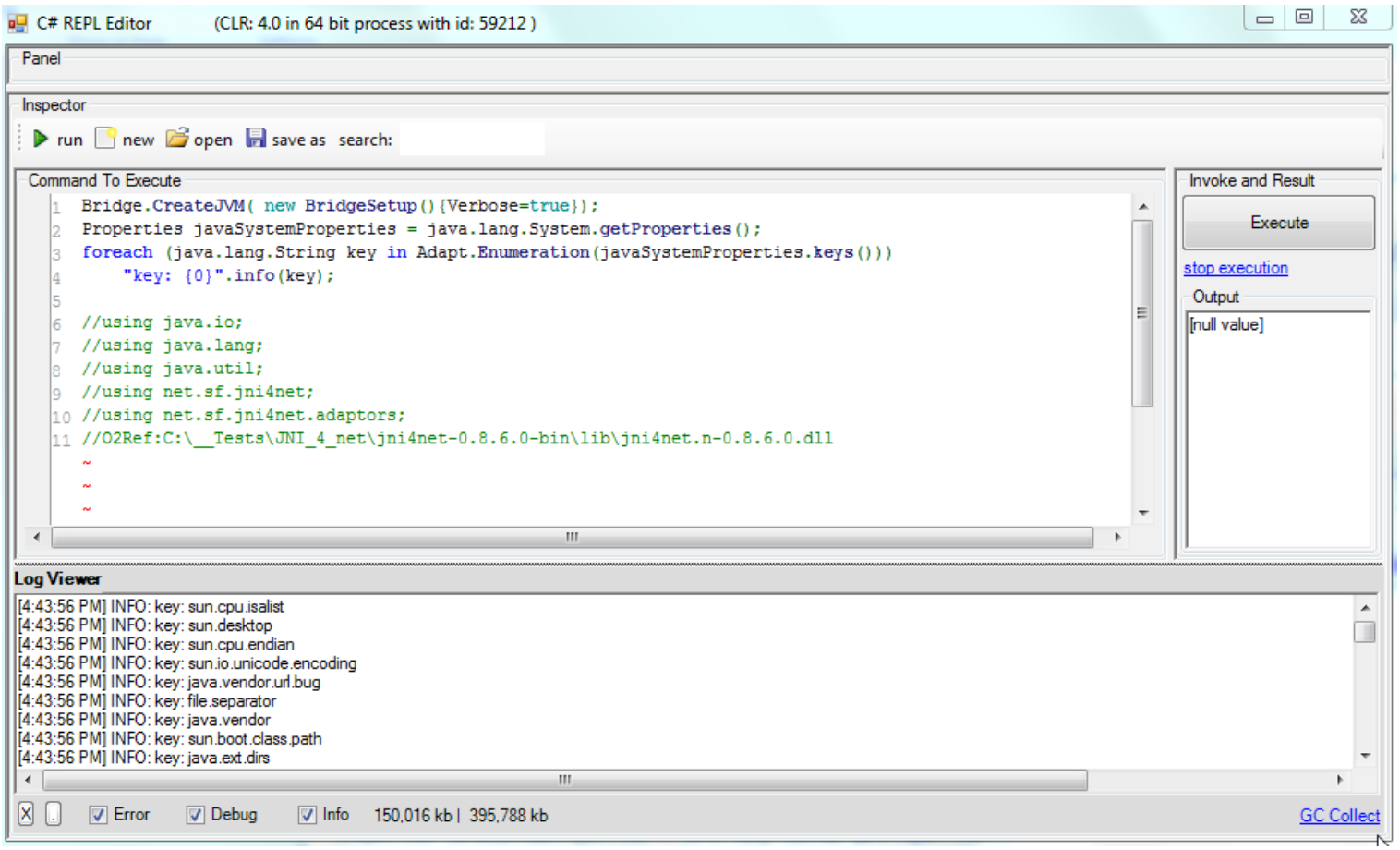


At this stage we have a CLR and JVM in the same process :)

Using Jni4Net inside C# REPL enviroment (inside ZAP process)

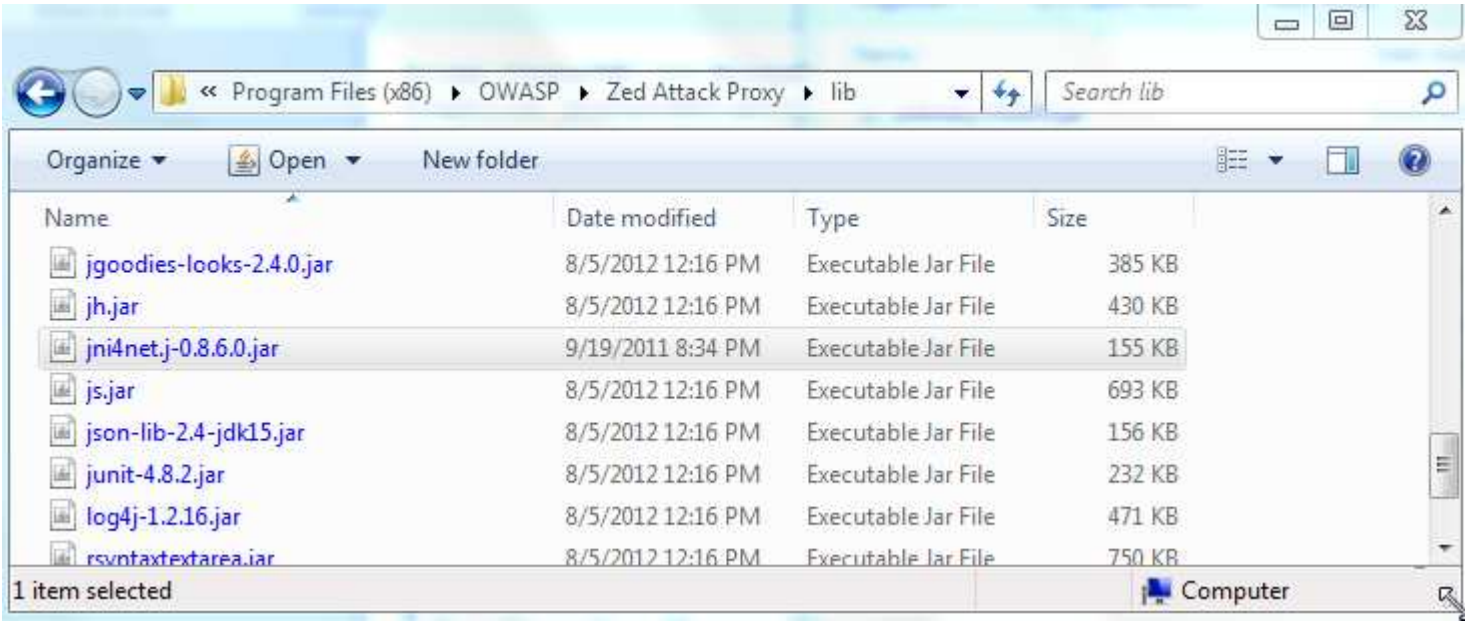
The next step is to access the JVM from the C# REPL environment, which we will using the Jni4Net module:

```
Bridge.CreateJVM( new BridgeSetup(){Verbose=true});
Properties javaSystemProperties = java.lang.System.getProperties();
foreach (java.lang.String key in Adapt.Enumeration(javaSystemProperties.keys()))
       "key: {0}".info(key);

//using java.io;
//using java.lang;
//using java.util;
//using net.sf.jni4net;
//using net.sf.jni4net.adaptors;
//O2Ref:C:\__Tests\JNI_4_net\jni4net-0.8.6.0-bin\lib\jni4net.n-0.8.6.0.dll
```



Note how in the screenshot above we are accessing the System properties from a C# script

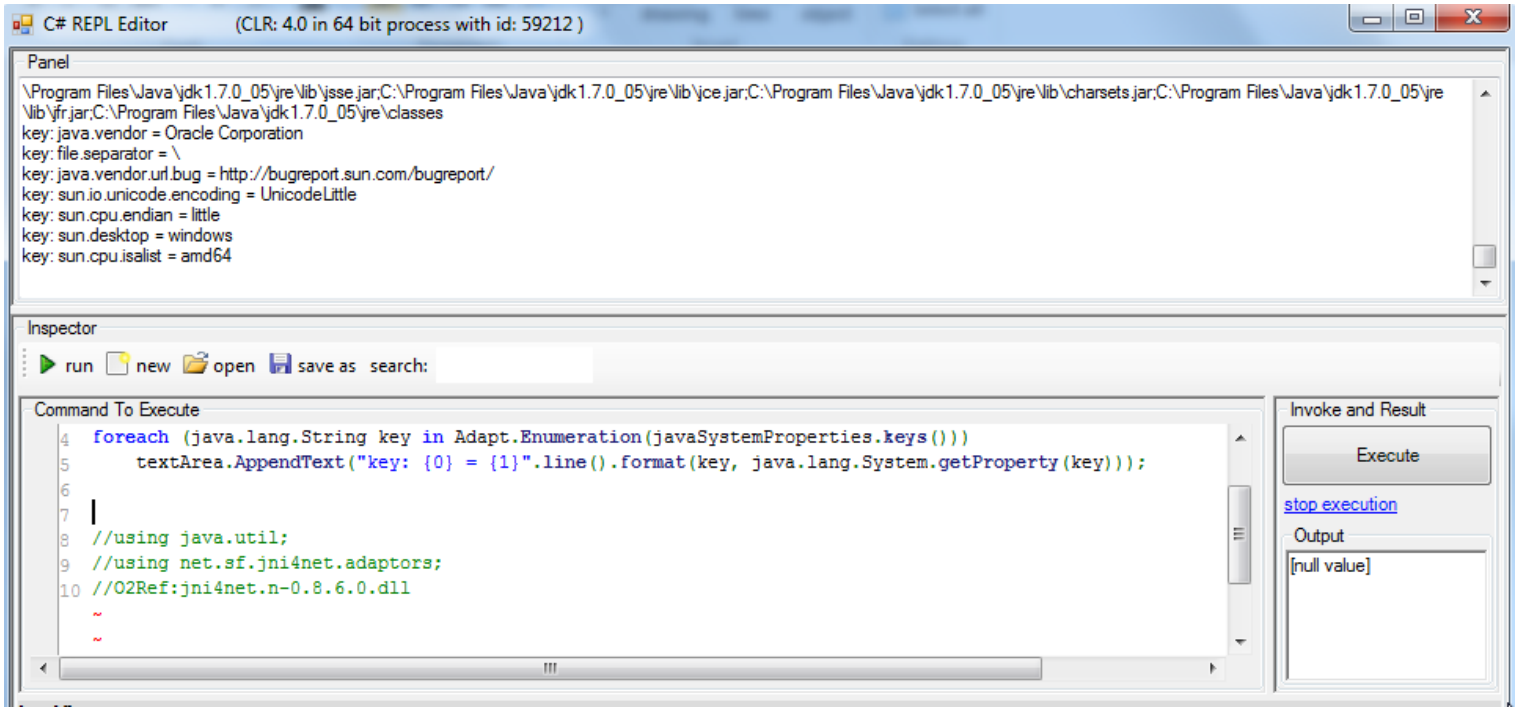
At the moment this is still working with some hardcoded paths (at the top to **jni4net.n-0.8.6.0.dll** and below the **jni4net.j-8.6.0.jar**)



Viewing the JavaProperties (inside a .NET TextBox)

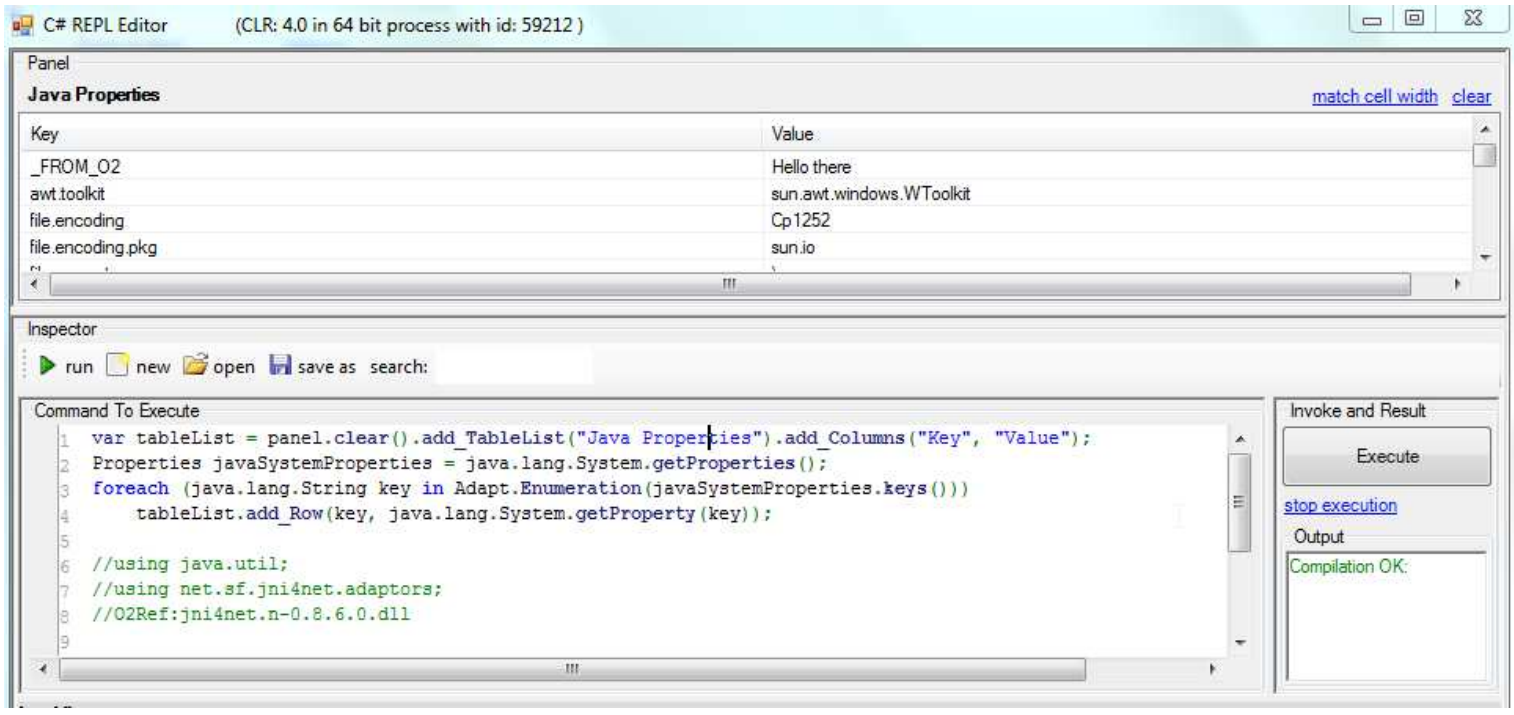
```
//Bridge.CreateJVM( new BridgeSetup(){Verbose=true});  //only needs to be done once
var textArea = panel.clear().add_TextArea();
Properties javaSystemProperties = java.lang.System.getProperties();
foreach (java.lang.String key in Adapt.Enumeration(javaSystemProperties.keys()))
      textArea.AppendText("key: {0} = {1}".line().format(key, java.lang.System.getProperty(key)));


//using java.util;
```

```
//using net.sf.jni4net.adaptors;
//O2Ref:jni4net.n-0.8.6.0.dll
```



Viewing the JavaProperties (inside a .NET TableList)

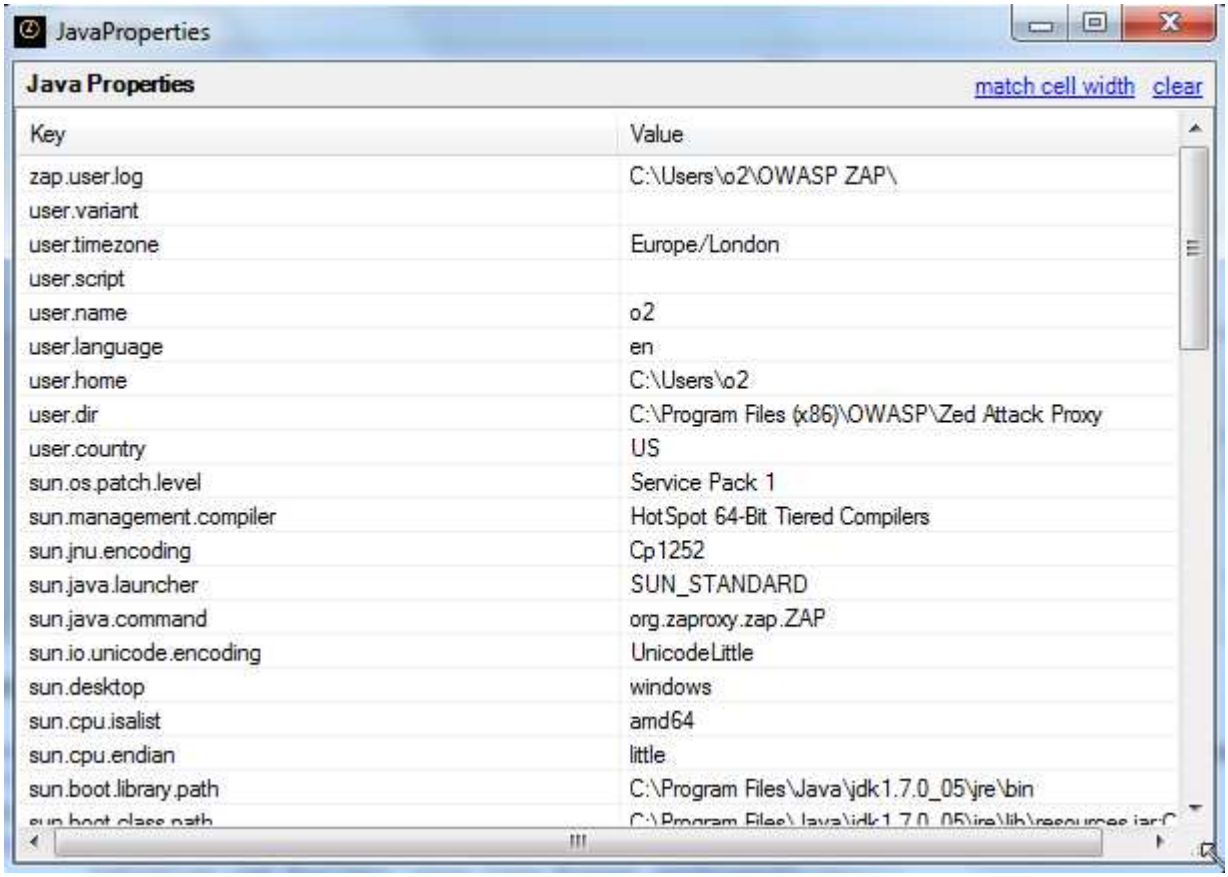
```
var tableList = panel.clear().add_TableList("Java Properties").add_Columns("Key", "Value");
Properties javaSystemProperties = java.lang.System.getProperties();
foreach (java.lang.String key in Adapt.Enumeration(javaSystemProperties.keys()))
       tableList.add_Row(key, java.lang.System.getProperty(key));

//using java.util;
//using net.sf.jni4net.adaptors;
//O2Ref:jni4net.n-0.8.6.0.dll
```



Viewing the JavaProperties (inside a .NET TableList on a stand-alone Form window)

```
var tableList = "JavaProperties".popupWindow().add_TableList("Java Properties").add_Columns("Key",
"Value");
Properties javaSystemProperties = java.lang.System.getProperties();
foreach (java.lang.String key in Adapt.Enumeration(javaSystemProperties.keys()))
```

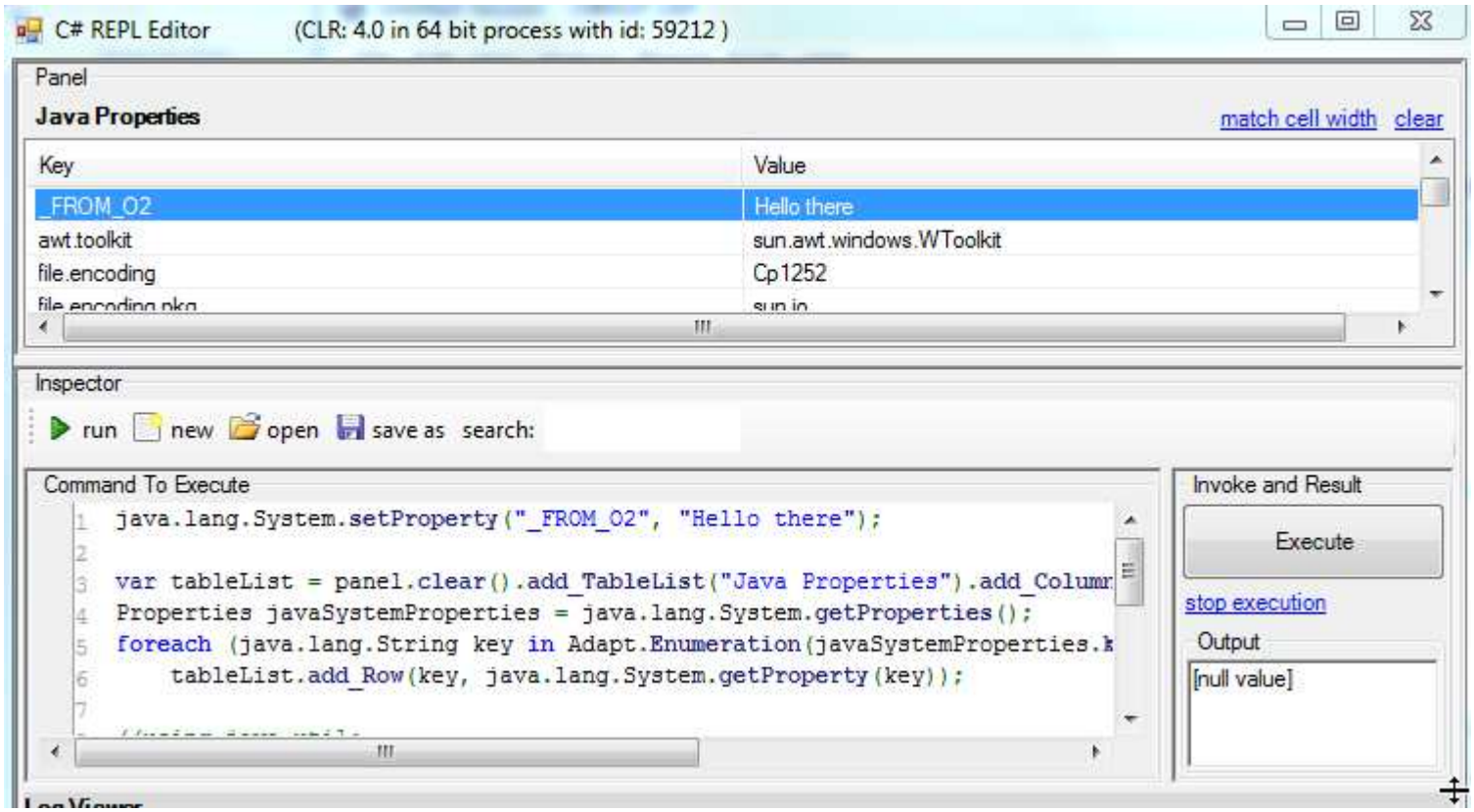
```
      tableList.add_Row(key, java.lang.System.getProperty(key));
```



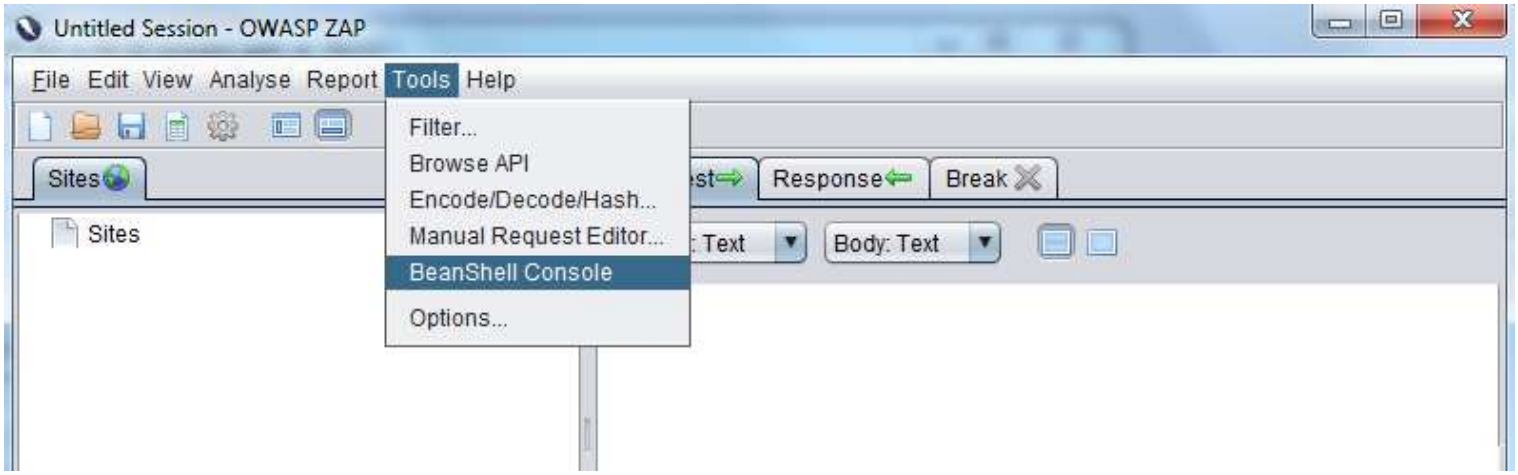
Setting a 'JVM System Property' value

```
java.lang.System.setProperty("_FROM_O2", "Hello there");
```

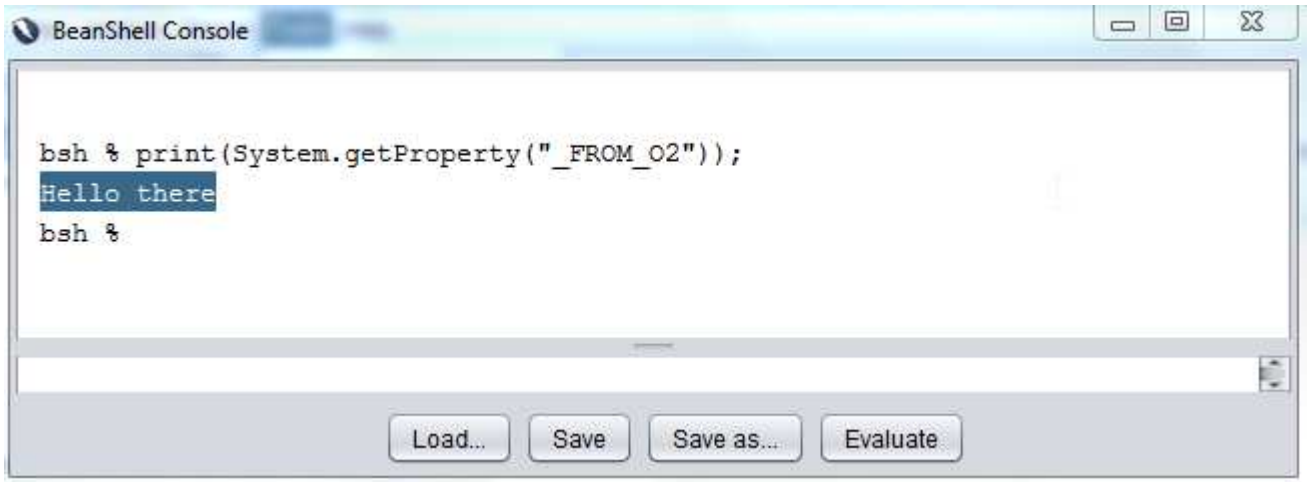
```
var tableList = panel.clear().add_TableList("Java Properties").add_Columns("Key", "Value");
Properties javaSystemProperties = java.lang.System.getProperties();
foreach (java.lang.String key in Adapt.Enumeration(javaSystemProperties.keys()))
      tableList.add_Row(key, java.lang.System.getProperty(key));
```



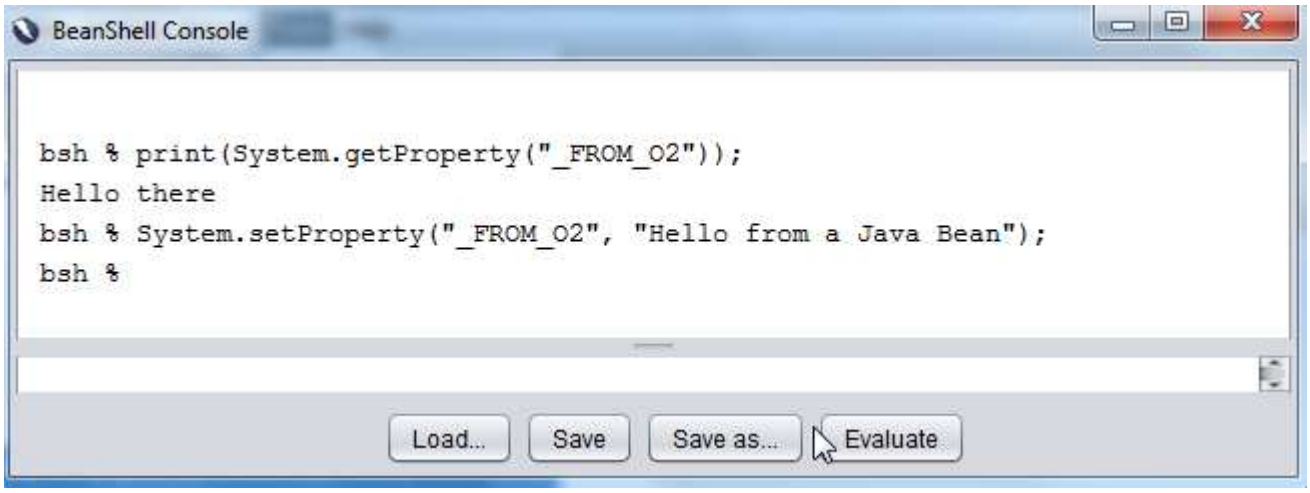
In ZAP open a BeanShell Console



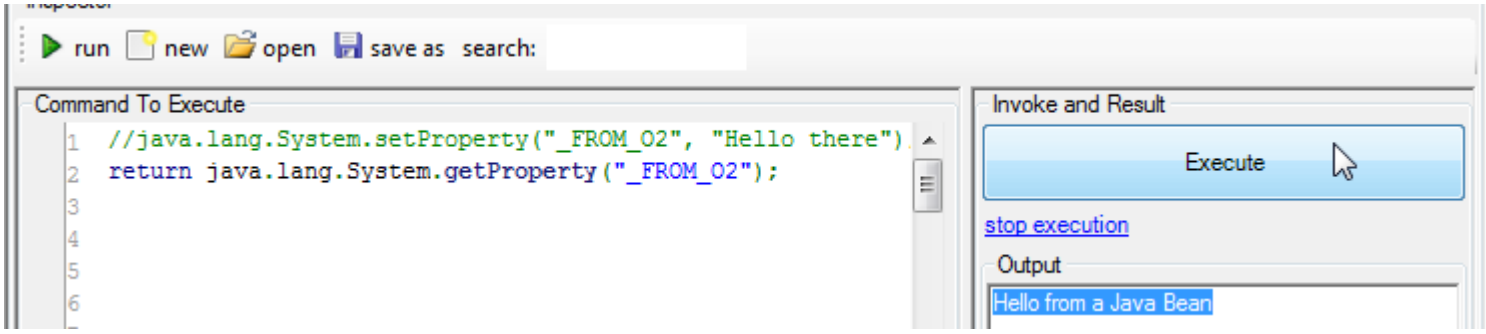
The property value (set from the .NET side) can be read



Setting the System Property value in the Java Side

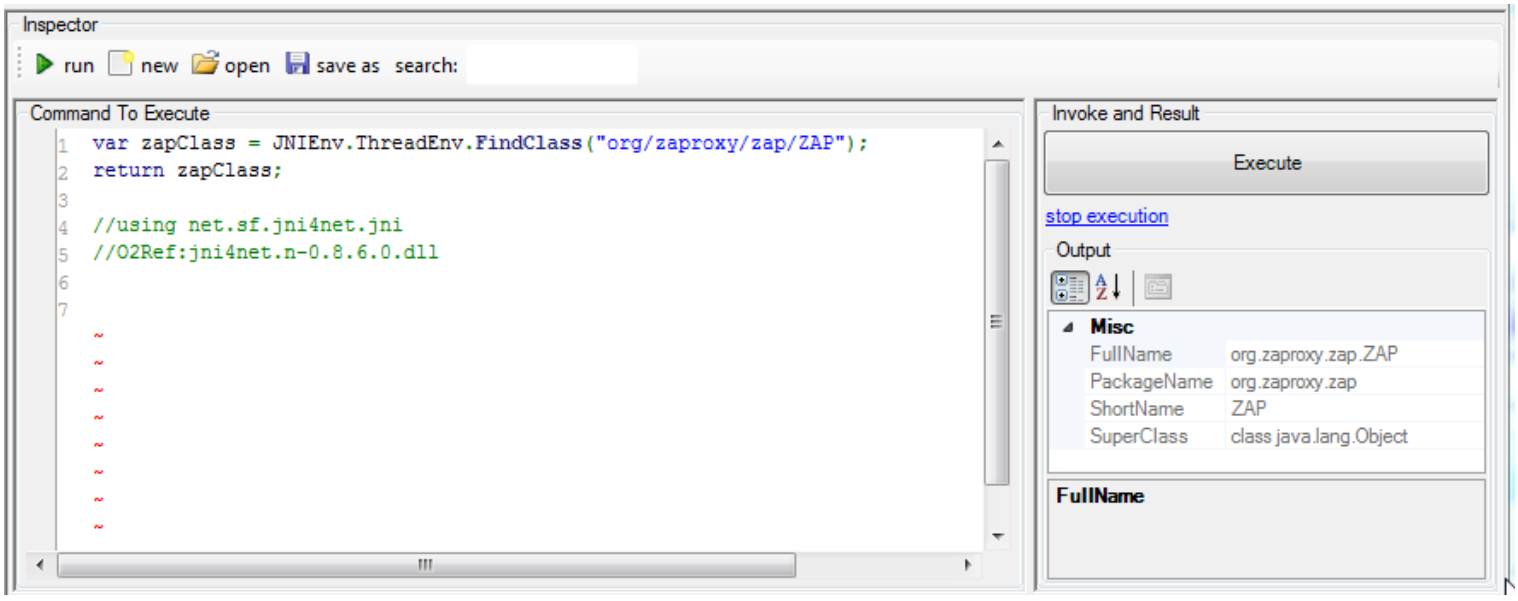


Reading that System Property from the .Net side



Getting a Class Object using Reflection:

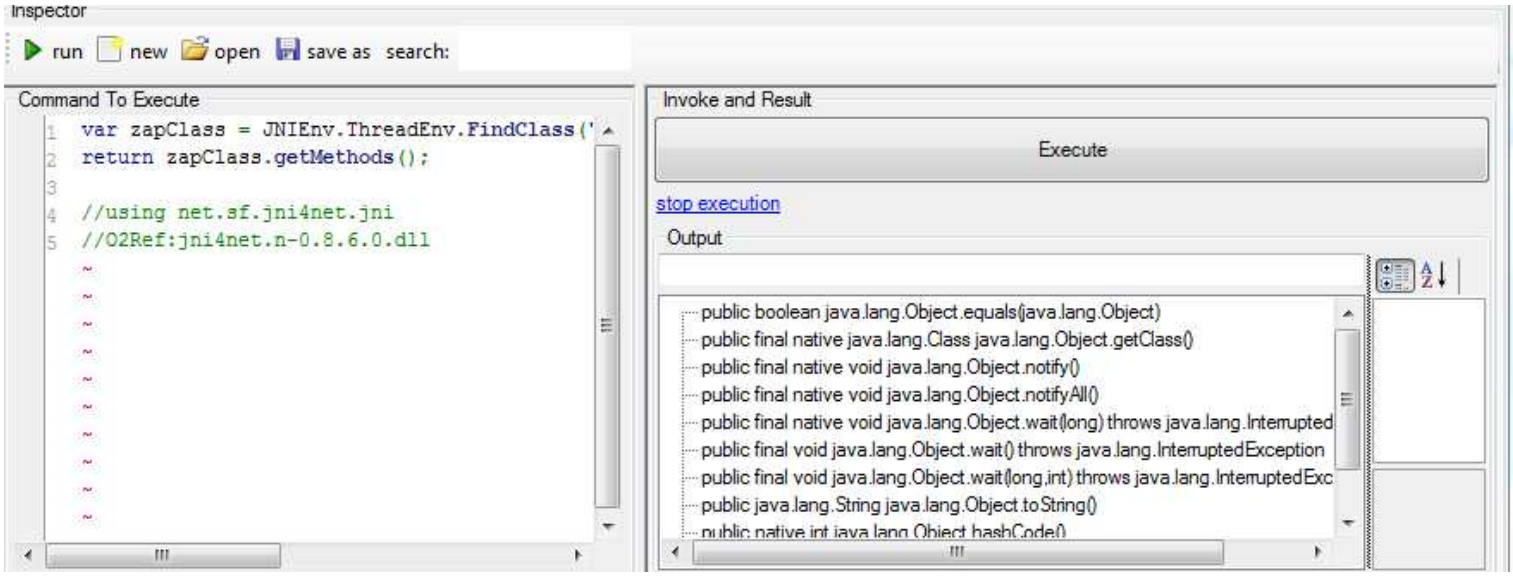
```
var zapClass = JNIEnv.ThreadEnv.FindClass("org/zaproxy/zap/ZAP");
return zapClass;

//using net.sf.jni4net.jni
//O2Ref:jni4net.n-0.8.6.0.dll
```



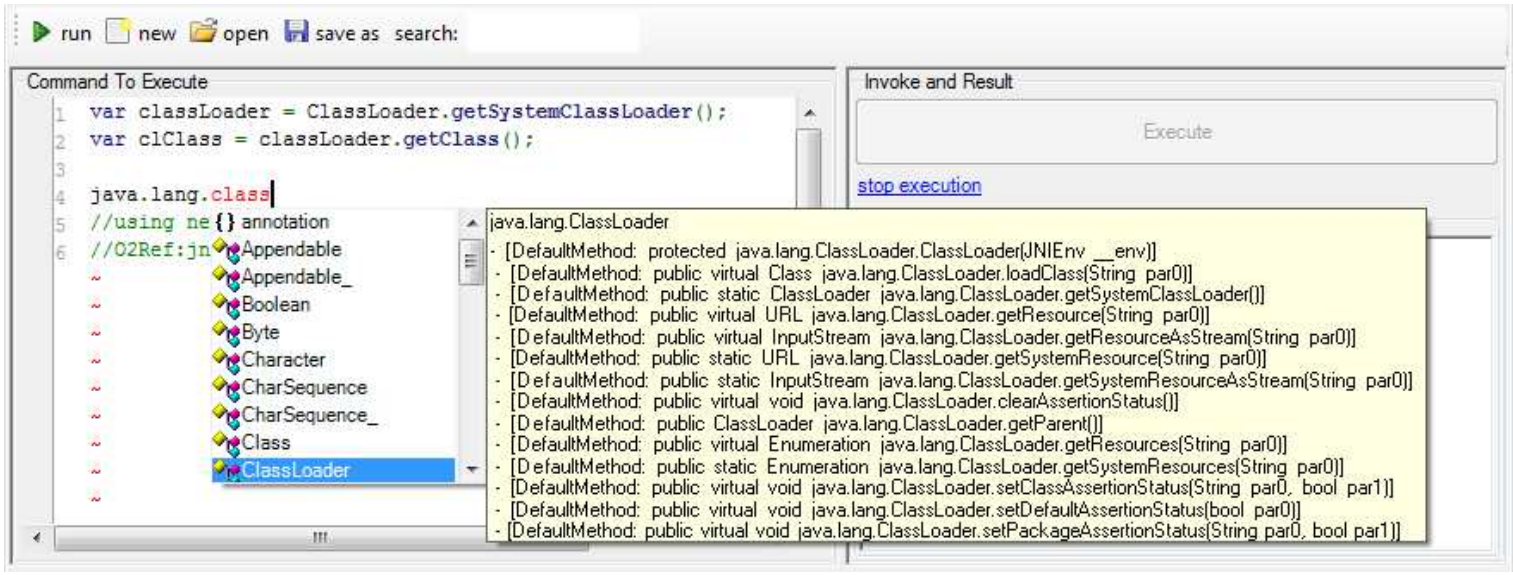
Getting the methods from a class:

```
var zapClass = JNIEnv.ThreadEnv.FindClass("org/zaproxy/zap/ZAP");
return zapClass.getMethods();

//using net.sf.jni4net.jni
```

```
//O2Ref:jni4net.n-0.8.6.0.dll
```

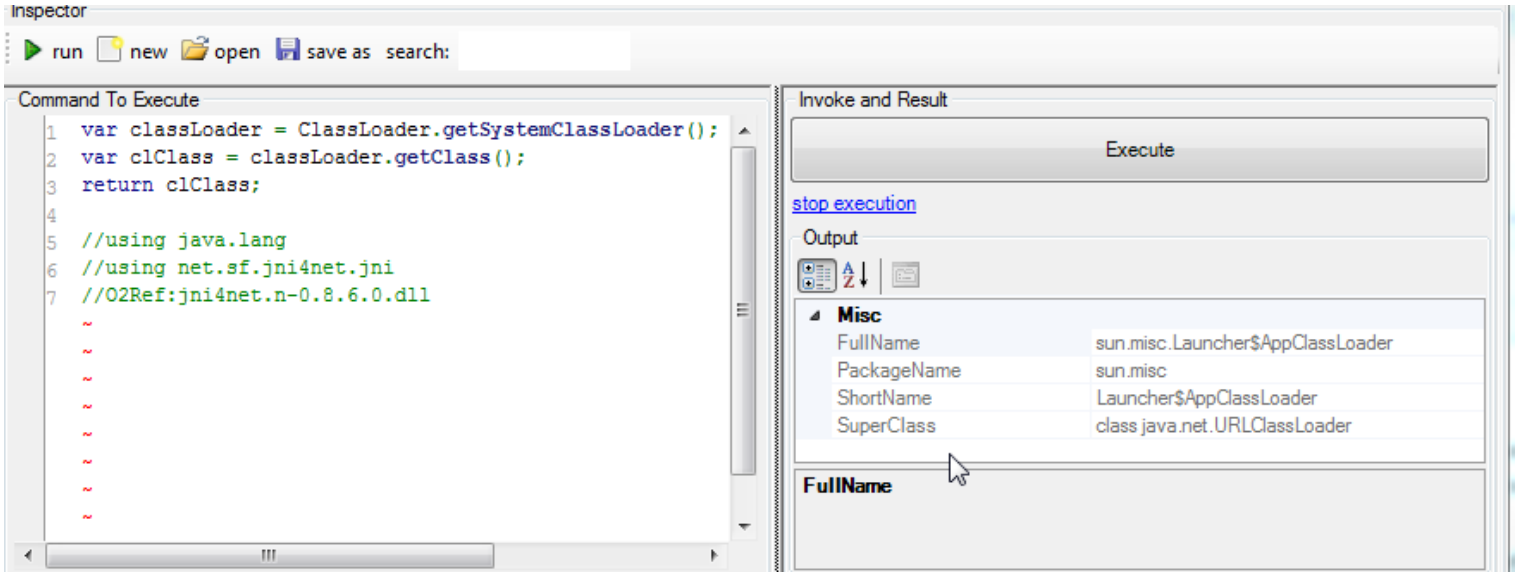


Intelisense (code complete) works on the .Net side



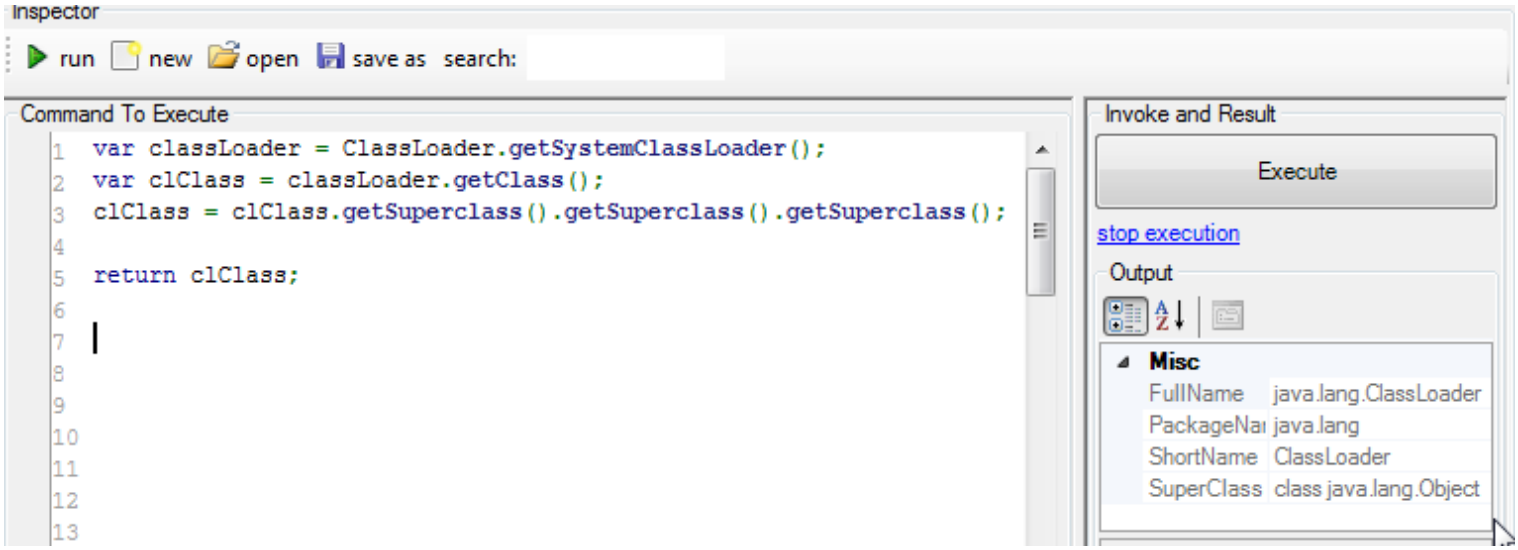
getting the Java System class loader

```
var classLoader = ClassLoader.getSystemClassLoader();
var clClass = classLoader.getClass();

java.lang.class
//using net.sf.jni4net.jni
//O2Ref:jni4net.n-0.8.6.0.dll
```



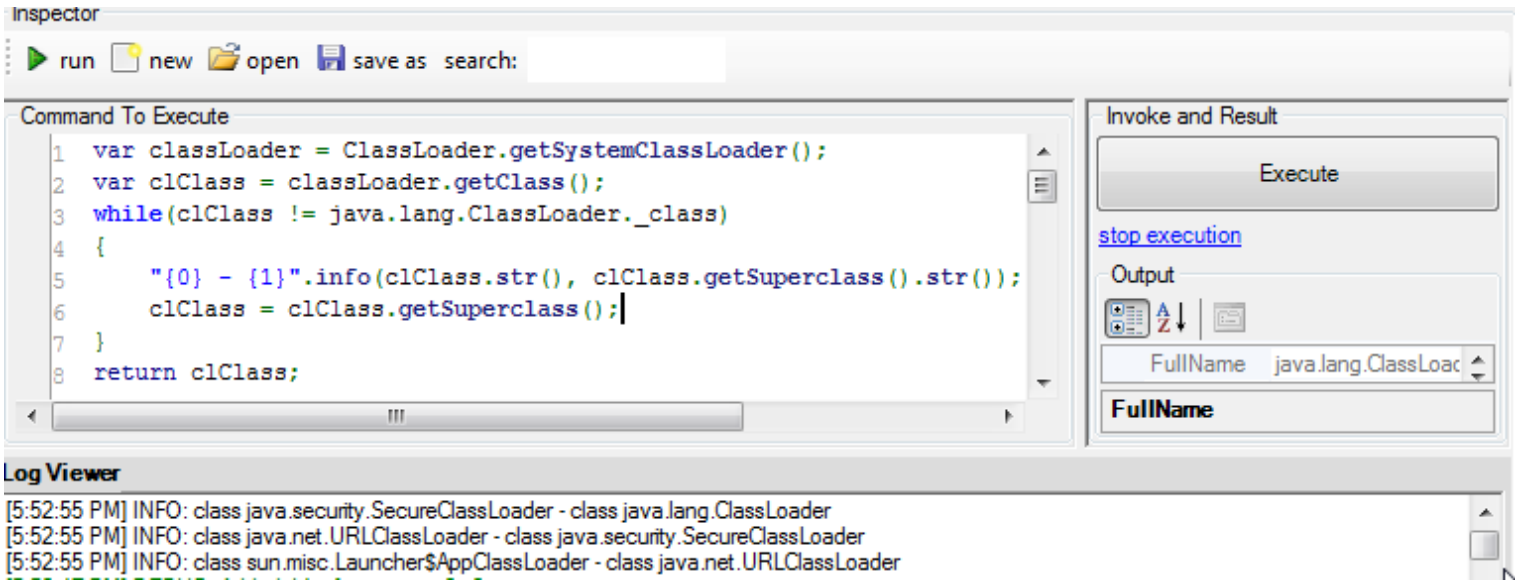
getting the top SuperClass class loader (manually)

```
var classLoader = ClassLoader.getSystemClassLoader();
var clClass = classLoader.getClass();
clClass = clClass.getSuperclass().getSuperclass().getSuperclass();

return clClass;
```



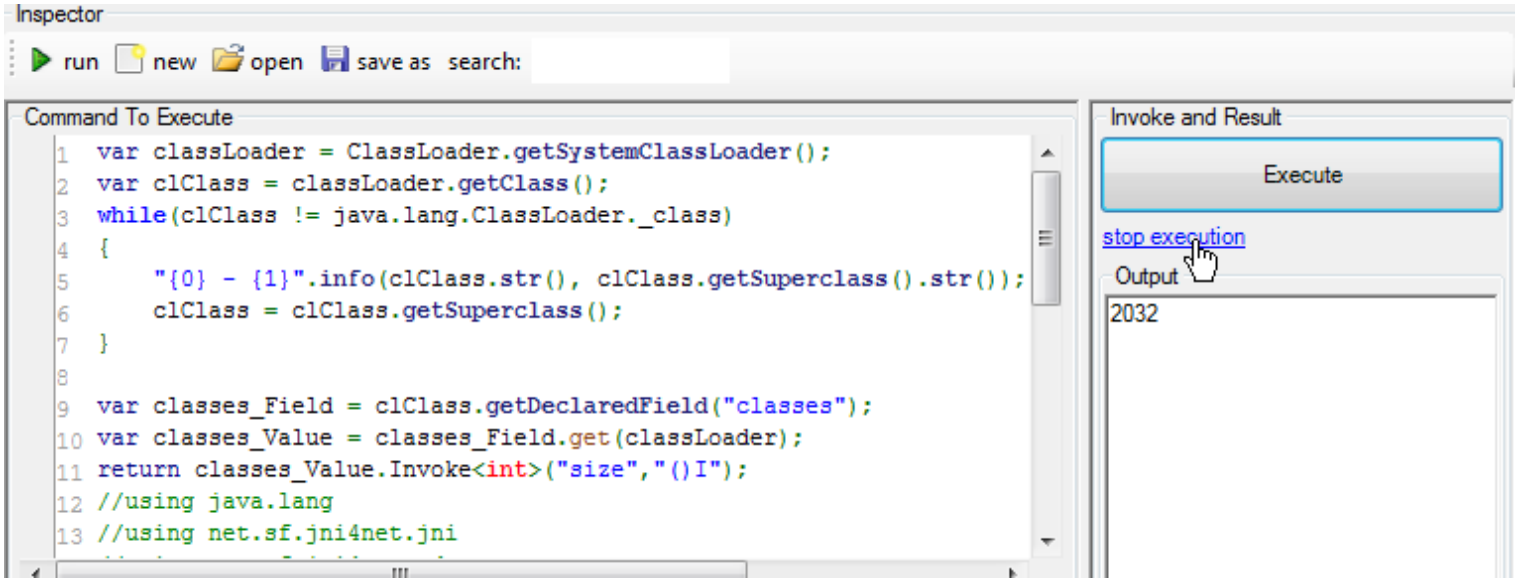
getting the top SuperClass class loader (using while loop)

```
var classLoader = ClassLoader.getSystemClassLoader();
var clClass = classLoader.getClass();
while(clClass != java.lang.ClassLoader._class)
{
	"{0} - {1}".info(clClass.str(), clClass.getSuperclass().str());
	clClass = clClass.getSuperclass();
}
return clClass;
```



Invoking the 'size' method from the 'classes' field from the 'ClassLoader' live object

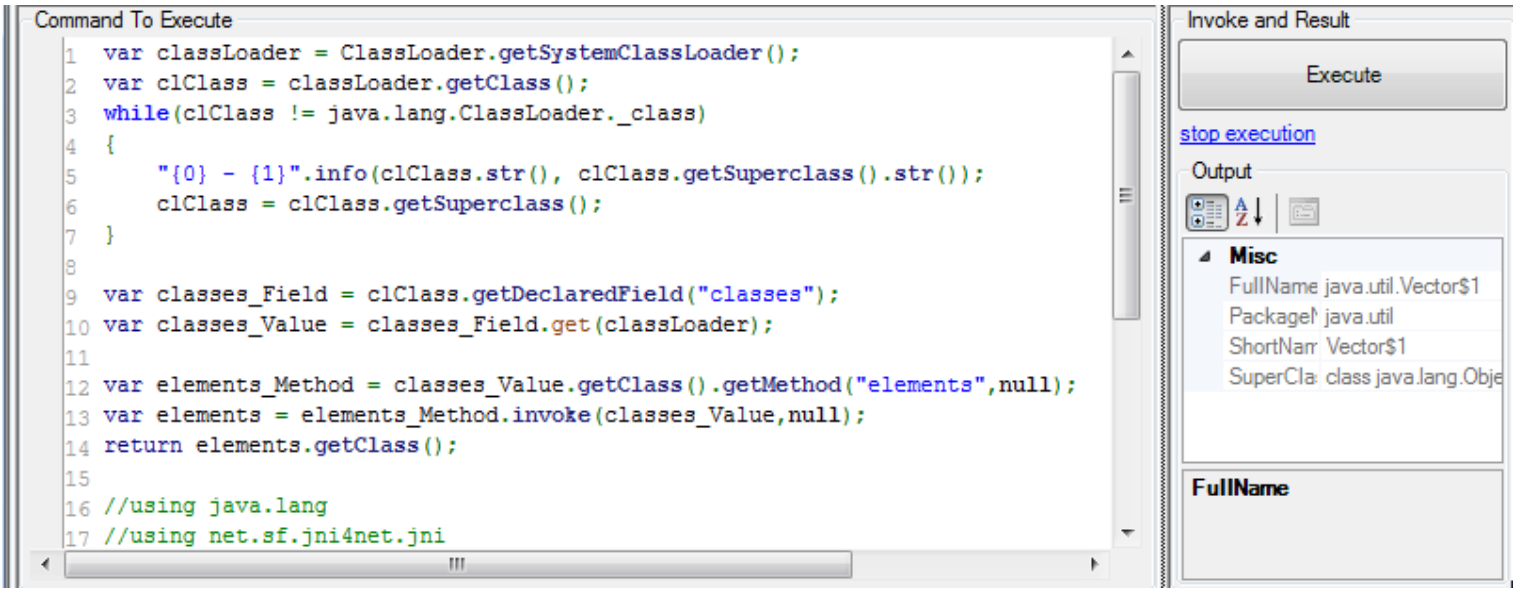
```
var classLoader = ClassLoader.getSystemClassLoader();
var clClass = classLoader.getClass();
while(clClass != java.lang.ClassLoader._class)
{
	"{0} - {1}".info(clClass.str(), clClass.getSuperclass().str());
	clClass = clClass.getSuperclass();
}

var classes_Field = clClass.getDeclaredField("classes");
var classes_Value = classes_Field.get(classLoader);
return classes_Value.Invoke<int>("size","()I");
//using java.lang
//using net.sf.jni4net.jni
//using net.sf.jni4net.adaptors
//O2Ref:jni4net.n-0.8.6.0.dll
```



invoking the 'elements' method from the 'classes' field from the 'ClassLoader' live object

```
var classLoader = ClassLoader.getSystemClassLoader();
var clClass = classLoader.getClass();
while(clClass != java.lang.ClassLoader._class)
{
	"{0} - {1}".info(clClass.str(), clClass.getSuperclass().str());
	clClass = clClass.getSuperclass();
}
```

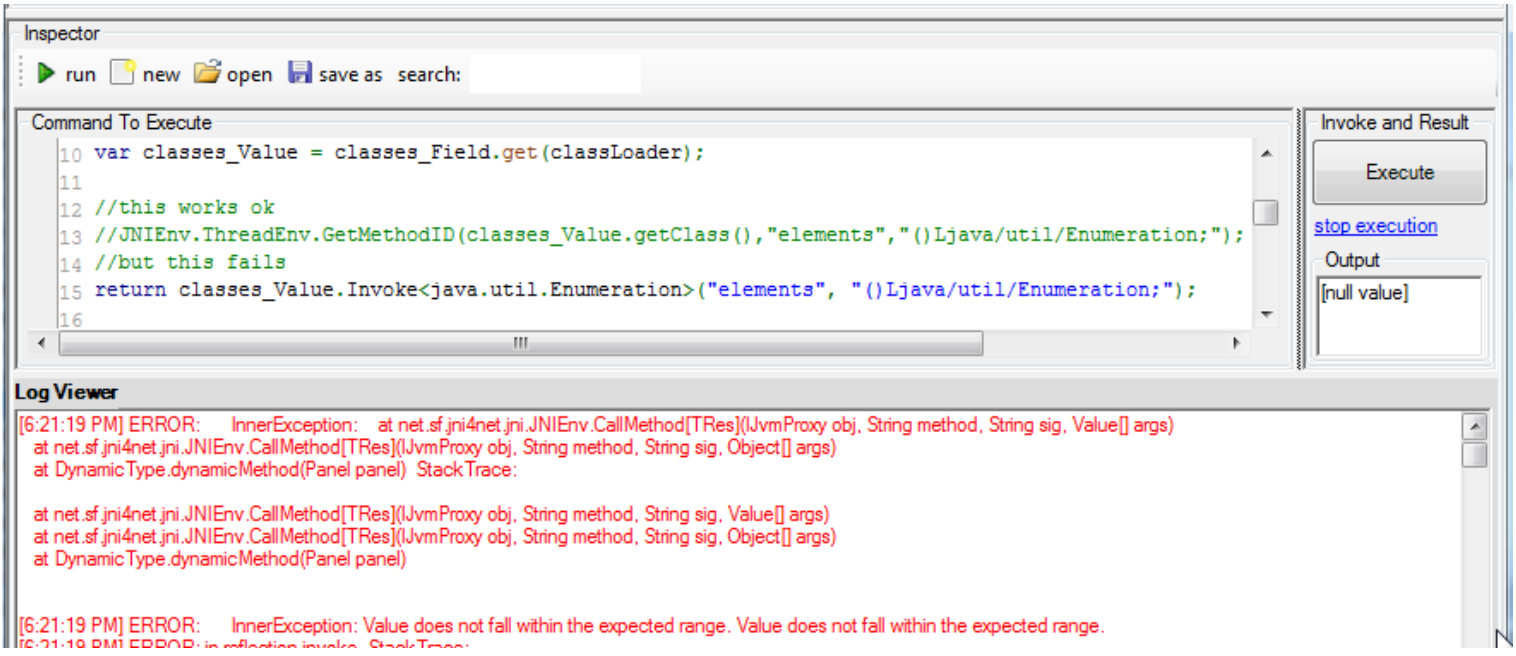
```
var classes_Field = clClass.getDeclaredField("classes");
var classes_Value = classes_Field.get(classLoader);

var elements_Method = classes_Value.getClass().getMethod("elements",null);
var elements = elements_Method.invoke(classes_Value,null);
return elements.getClass();

//using java.lang
//using net.sf.jni4net.jni
//using net.sf.jni4net.adaptors
//O2Ref:jni4net.n-0.8.6.0.dll
```



note: the classes field is a java.util.Vector whose details can be seen here
http://docs.oracle.com/javase/1.4.2/docs/api/java/util/Vector.html

Enumerating classes list (not working: trying to cast the object into a Jni4Net java.util.Enumeration)

```
var classes_Field = clClass.getDeclaredField("classes");
var classes_Value = classes_Field.get(classLoader);

//this works ok
//JNIEnv.ThreadEnv.GetMethodID(classes_Value.getClass(),"elements","()Ljava/util/Enumeration;");
//but this fails
return classes_Value.Invoke<java.util.Enumeration>("elements", "()Ljava/util/Enumeration;");
```

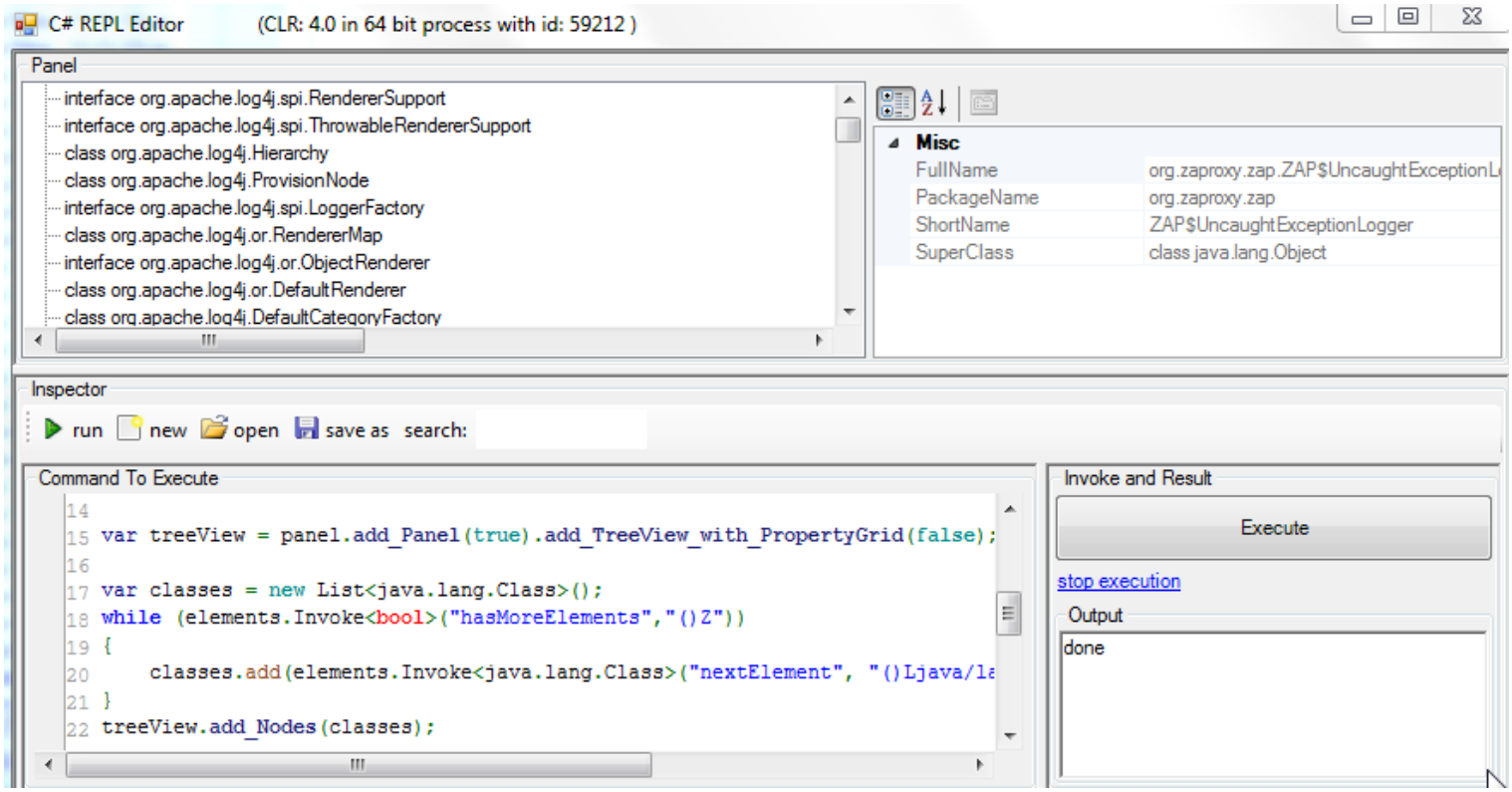


Enumerating classes list (working using reflection to invoke hasMoreElements and nextElement)

```
var classes_Field = clClass.getDeclaredField("classes");
var classes_Value = classes_Field.get(classLoader);

var elements_Method = classes_Value.getClass().getMethod("elements",null);
var elements = elements_Method.invoke(classes_Value,null);
while (elements.Invoke<bool>("hasMoreElements","()Z"))
	elements.Invoke<java.lang.Object>("nextElement", "()Ljava/lang/Object;").str().info();
```

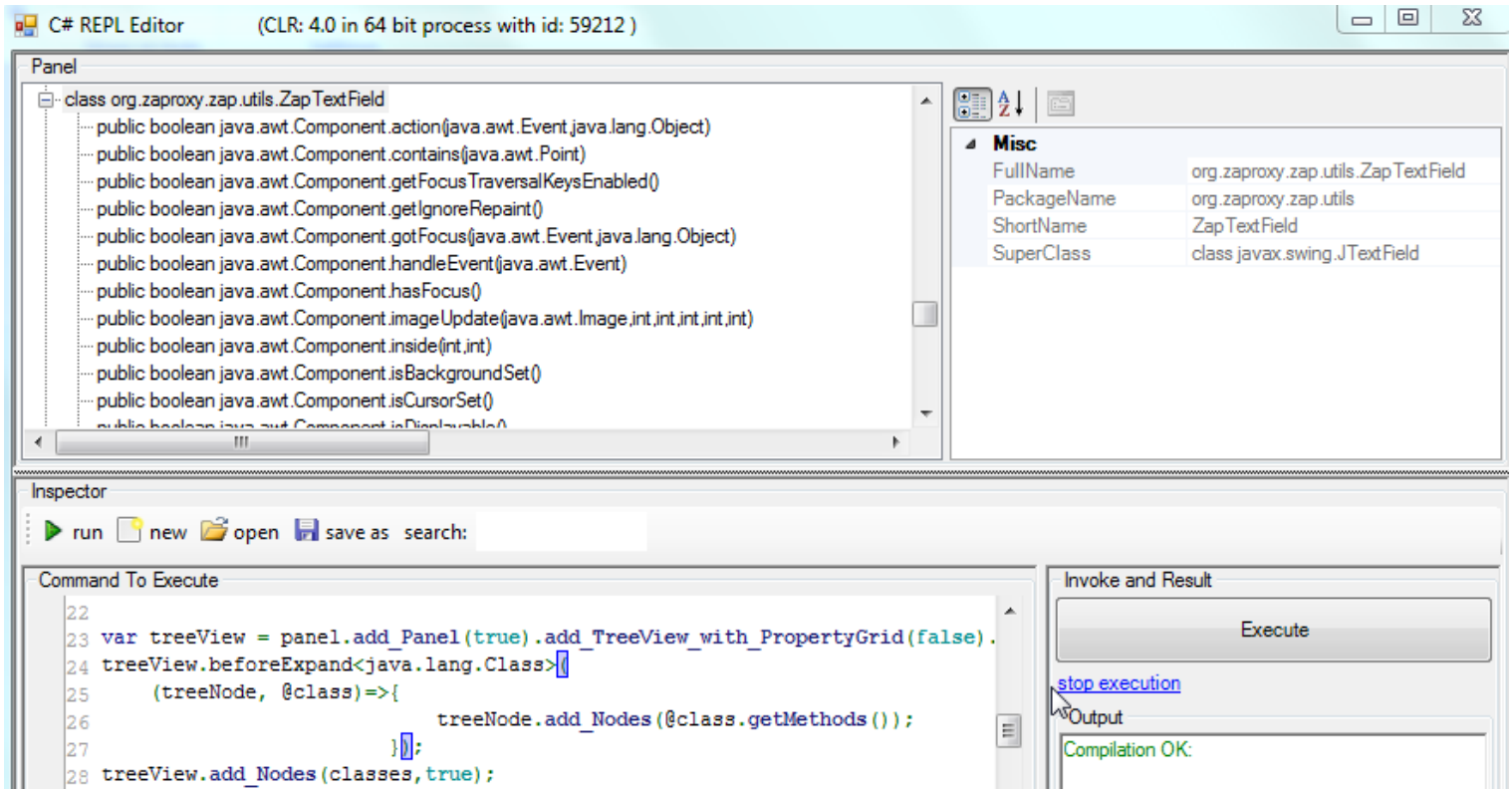
Viewing the classes in a TreeView

```
var treeView = panel.add_Panel(true).add_TreeView_with_PropertyGrid(false);

var classes = new List<java.lang.Class>();
while (elements.Invoke<bool>("hasMoreElements","()Z"))
{
	classes.add(elements.Invoke<java.lang.Class>("nextElement", "()Ljava/lang/Object;"));
}
treeView.add_Nodes(classes);
```



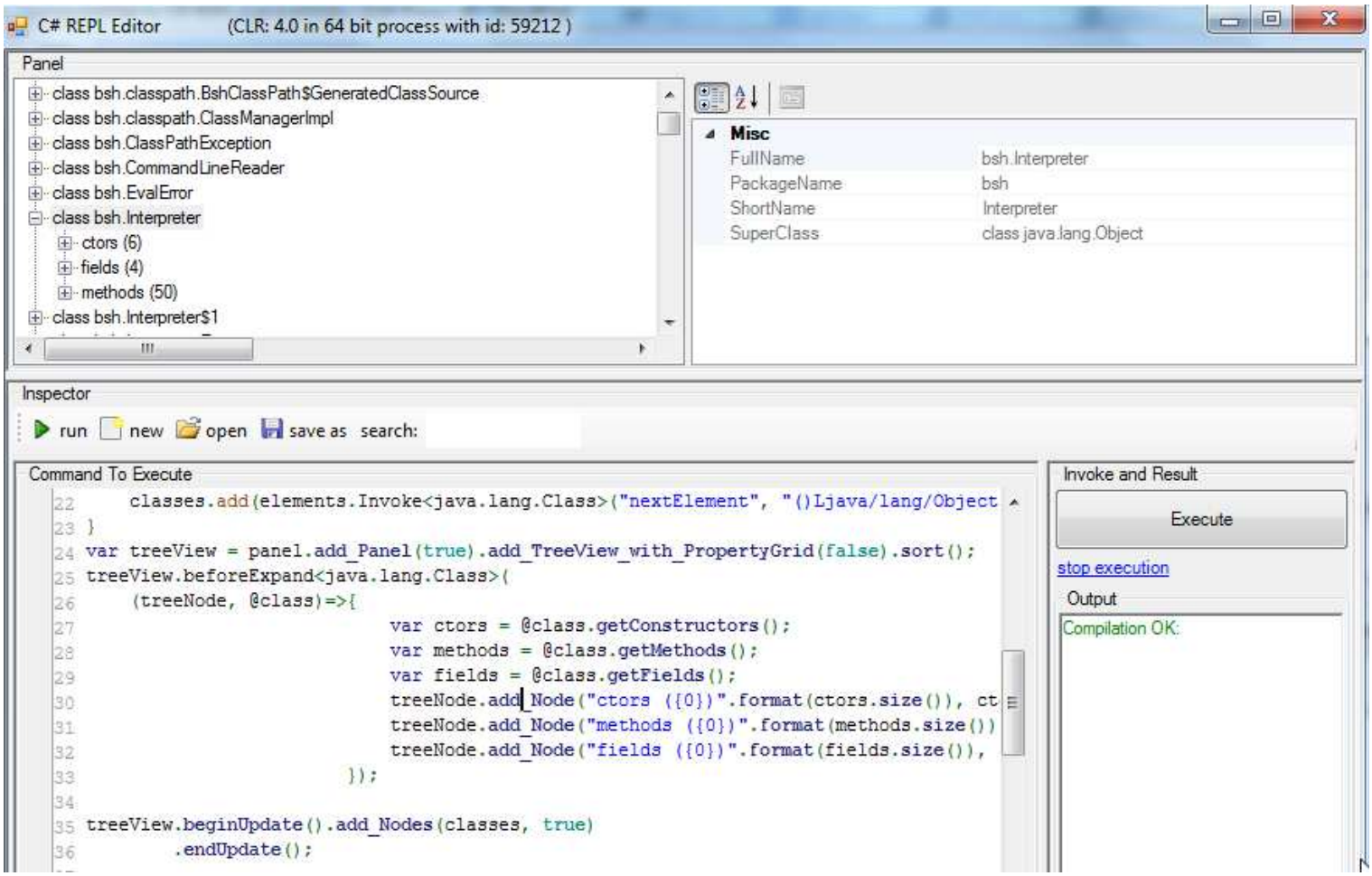
Showing the methods and classes on a treeView:

```
var treeView = panel.add_Panel(true).add_TreeView_with_PropertyGrid(false).sort();
treeView.beforeExpand<java.lang.Class>(
       (treeNode, @class)=>{
                                                  treeNode.add_Nodes(@class.getMethods());
                                           });
treeView.add_Nodes(classes,true);
```



Showing the classes, constructors, methods and fields in the treeview

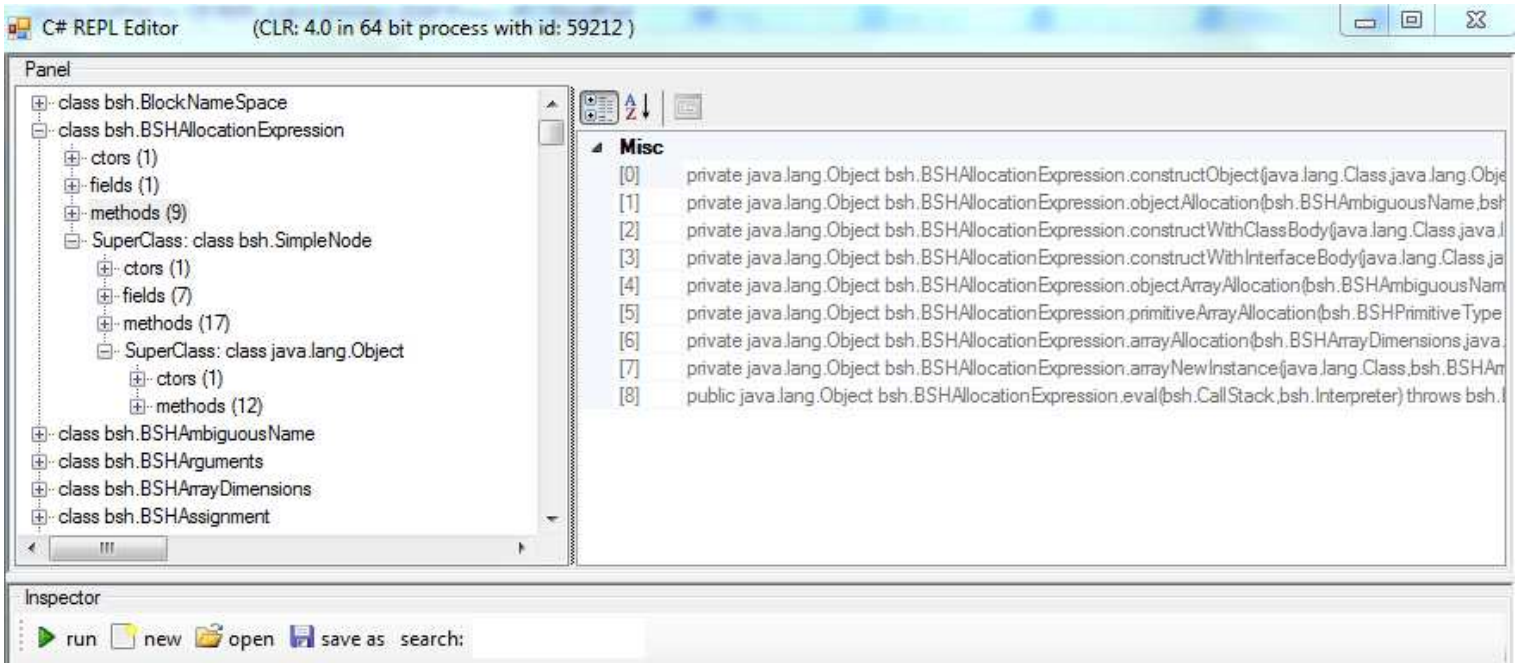
```
var treeView = panel.add_Panel(true).add_TreeView_with_PropertyGrid(false).sort();
treeView.beforeExpand<java.lang.Class>(
   (treeNode, @class)=>{
                           var ctors = @class.getConstructors();
                           var methods = @class.getMethods();
                           var fields = @class.getFields();
                           treeNode.add_Node("ctors ({0})".format(ctors.size()), ctors, true);
                           treeNode.add_Node("methods ({0})".format(methods.size()), methods, true);
                           treeNode.add_Node("fields ({0})".format(fields.size()), fields, true);
                       });

treeView.beginUpdate().add_Nodes(classes, true)
          .endUpdate();
```



Showing the classes, constructors, methods and fields and SuperClass in the treeview (only if they exist)

```
var treeView = panel.add_Panel(true).add_TreeView_with_PropertyGrid(false).sort();
treeView.beforeExpand<java.lang.Class>(
       (treeNode, @class)=>{
                           var ctors = @class.getDeclaredConstructors();
                           var methods = @class.getDeclaredMethods();
                           var fields = @class.getDeclaredFields();
                           var superClass = @class.getSuperclass();
                           //var annotations = @class.getDeclaredAnnotations();
                           if (superClass.notNull())   treeNode.add_Node("SuperClass: {0}".format
(superClass), superClass, true);
                           if (ctors.size() > 0)       treeNode.add_Node("ctors ({0})".format(ctors.size
()), ctors,true);
                           if (methods.size() > 0)      treeNode.add_Node("methods ({0})".format
(methods.size()), methods,  ctors.size()>0);
                           if (fields.size() > 0)      treeNode.add_Node("fields ({0})".format
(fields.size()), fields,  ctors.size()>0);
                                       });

treeView.beginUpdate().add_Nodes(classes, true)
```

```
        .endUpdate();
```



Showing method's name, full signature and parameters

```
var treeView = panel.add_Panel(true).add_TreeView_with_PropertyGrid(false).sort();
treeView.beforeExpand<java.lang.Class>(
     (treeNode, @class)=>{
                              var ctors = @class.getDeclaredConstructors();
                              var methods = @class.getDeclaredMethods();
                              var fields = @class.getDeclaredFields();
                              var superClass = @class.getSuperclass();
                              //var annotations = @class.getDeclaredAnnotations();
                              if (superClass.notNull()) treeNode.add_Node("SuperClass: {0}".format
(superClass), superClass, true);
                              if (ctors.size() > 0)
                                     treeNode.add_Node("ctors ({0})".format(ctors.size()), ctors,true);
                              if (methods.size() > 0)
                                     treeNode.add_Node("methods ({0})".format(methods.size()),
methods,true);
                              if (fields.size() > 0)
                                     treeNode.add_Node("fields ({0})".format(fields.size()), fields,true);
                        });

treeView.beforeExpand<java.lang.reflect.Method[]>(
       (treeNode,methods)=>{
                                     treeNode.add_Nodes(methods, (method)=>method.getName(), true);
                              });
treeView.beforeExpand<java.lang.reflect.Method>(
       (treeNode,method)=>{
                                     treeNode.add_Node("Signature: {0}".format(method), method);
                                     treeNode.add_Node("Parameters")
                                            .add_Nodes(method.getParameterTypes(),true);
                              });
```

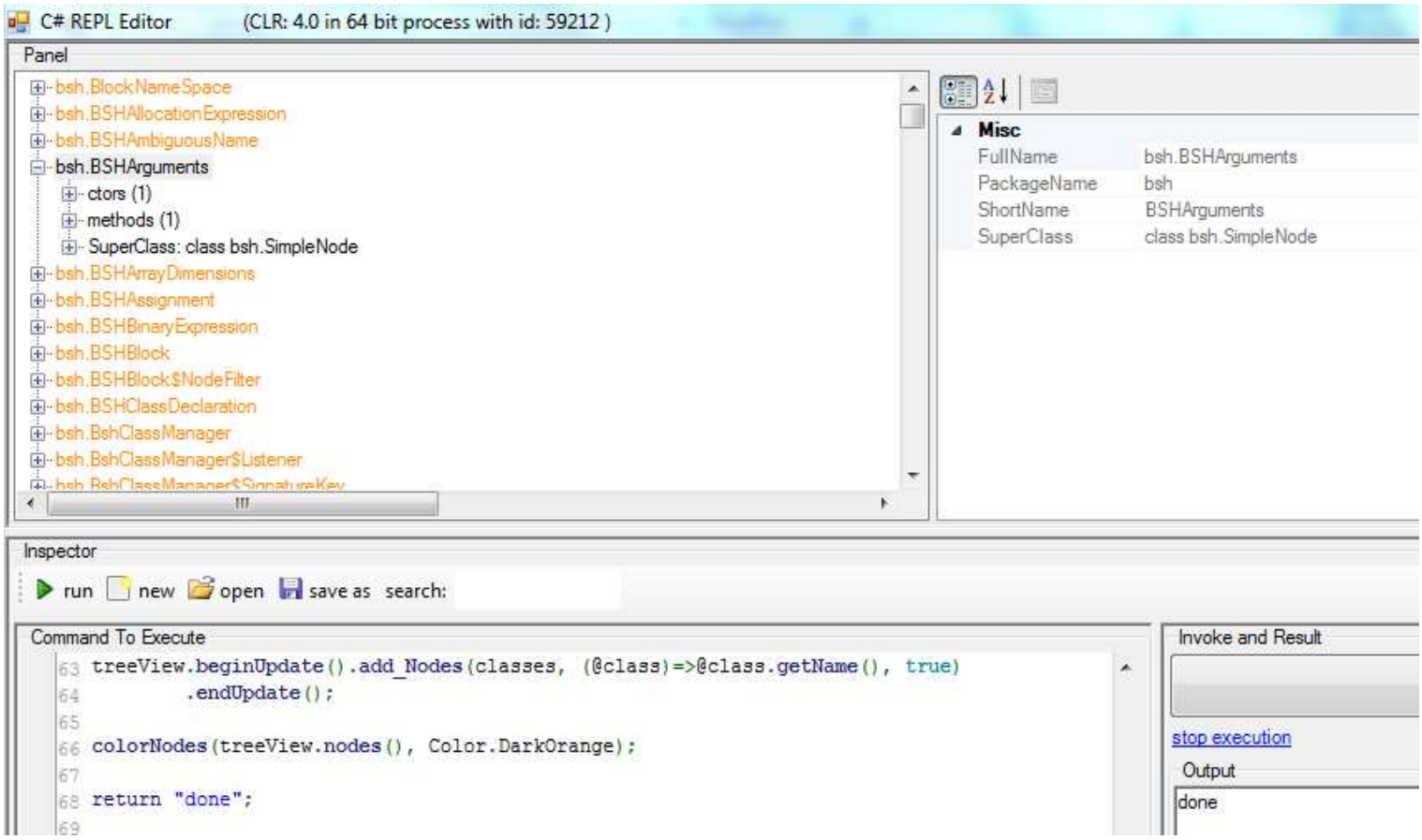
Adding missing (in O2's API) method to color all TreeNodes

sometimes the helper method we want is not available, so we can just add a temp one to the current script:

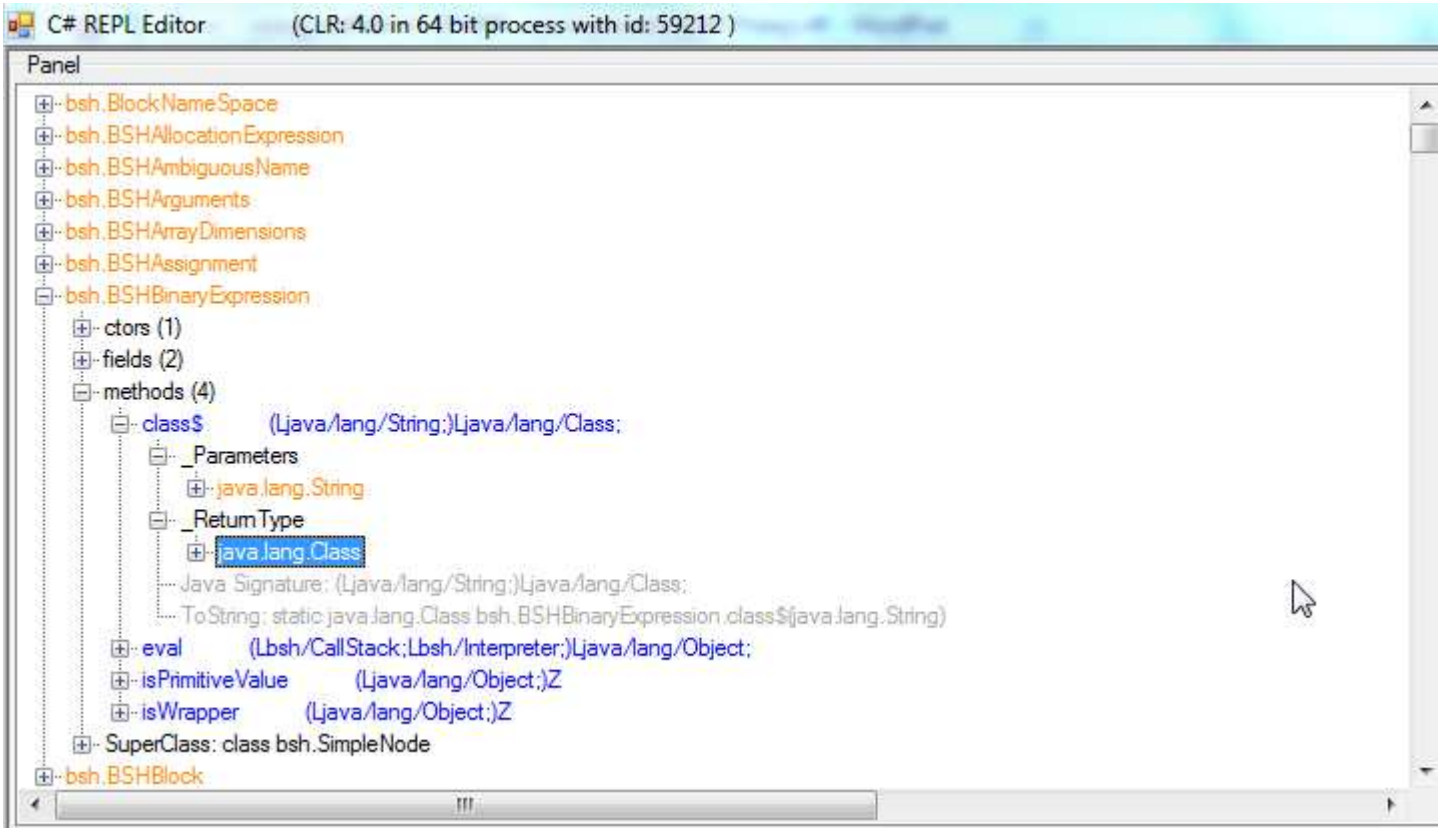
```
Action<List<TreeNode>, Color> colorNodes =
       (nodes, color)=>{
                              treeView.beginUpdate();
                              foreach(var node in nodes)
                                     node.foreColor(color);
                              treeView.endUpdate();
                        };
```

Color coding the classes (and showing class name in TreeNode text)

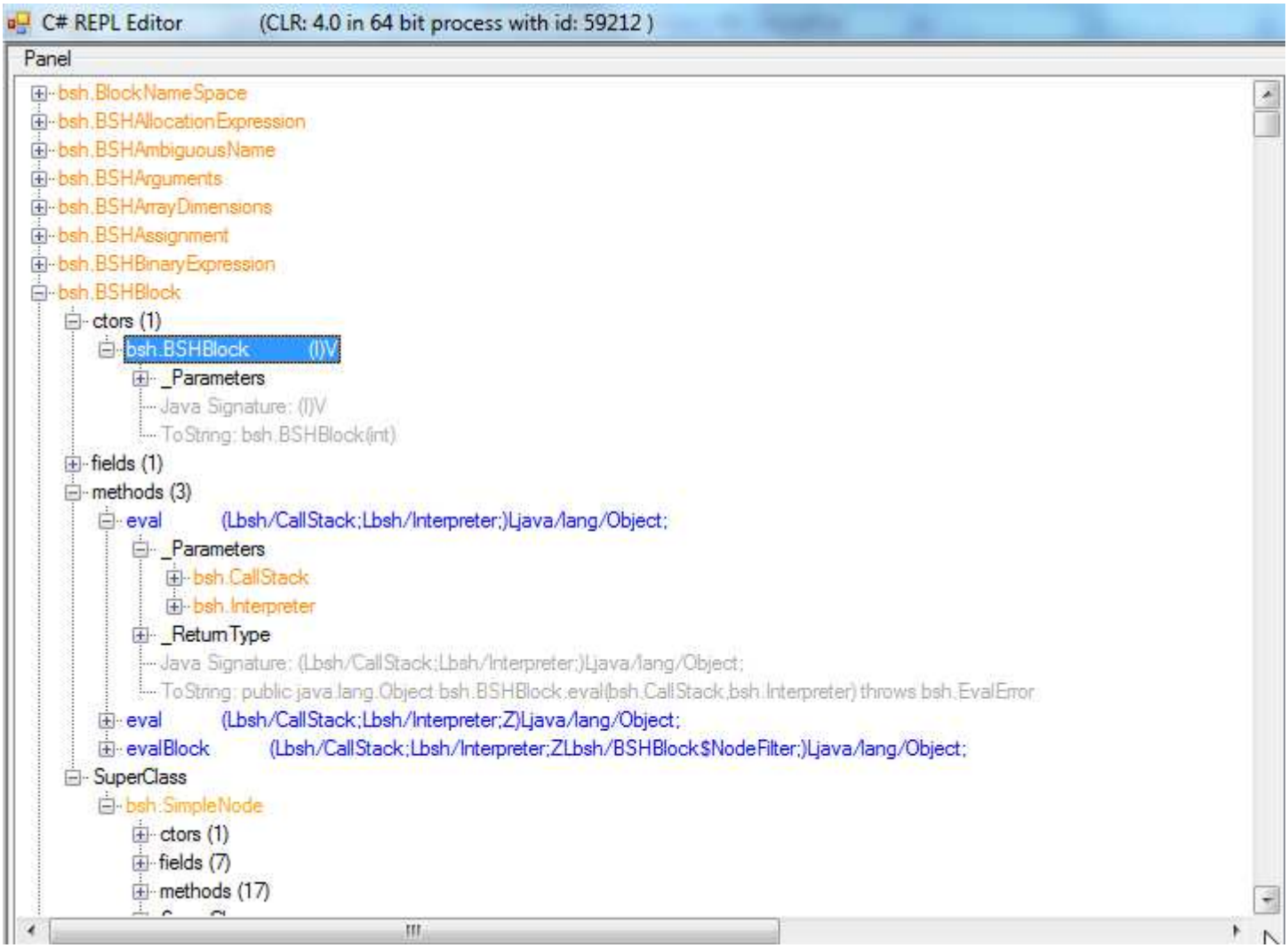
```
treeView.beginUpdate().add_Nodes(classes, (@class)=>@class.getName(), true)
          .endUpdate();

colorNodes(treeView.nodes(), Color.DarkOrange);
```



Color Coding methods and also showing ReturnType



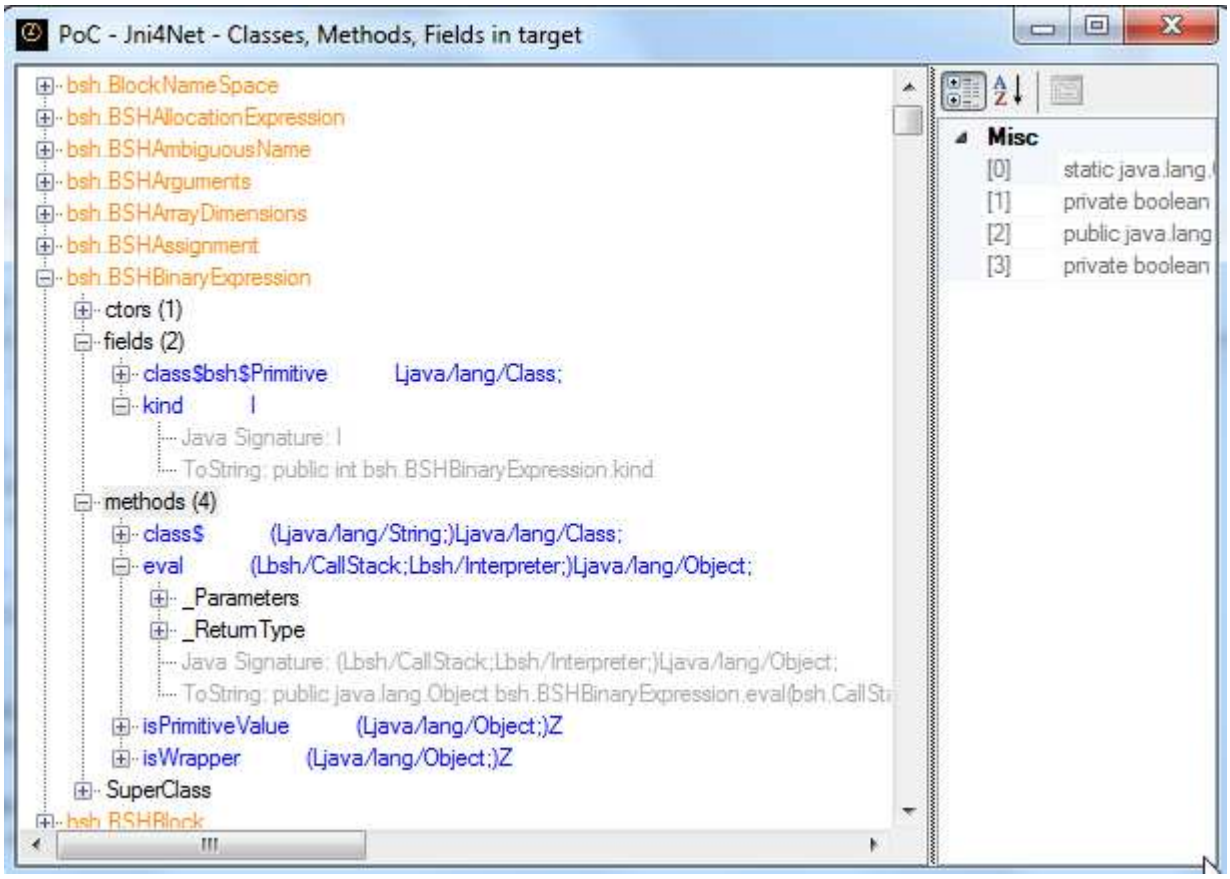
Showing and Color Coding the Constructors and Fields:



Making the last Gui a stand alone Form (full code below)

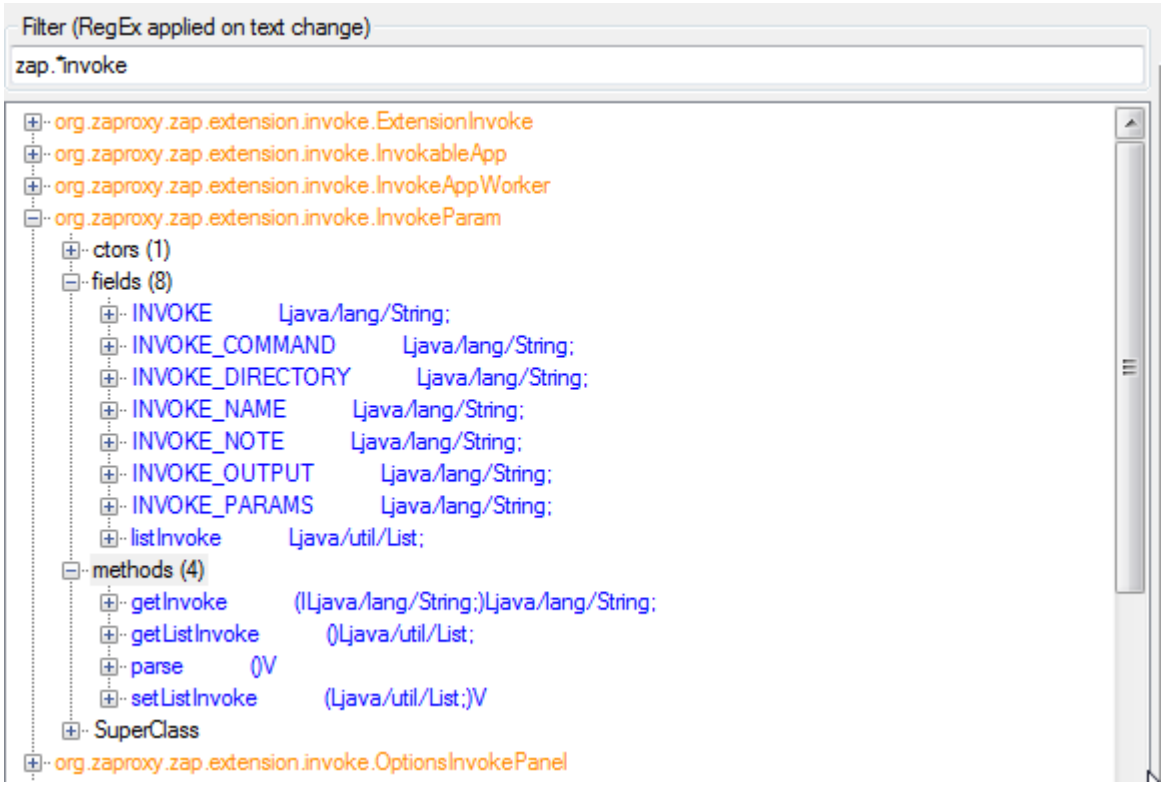
```
var topPanel = "Jni4Net - Classes, Methods, Fields in target".popupWindow();

var classLoader = ClassLoader.getSystemClassLoader();
var clClass = classLoader.getClass();
while(clClass != java.lang.ClassLoader._class)
{
	"{0} - {1}".info(clClass.str(), clClass.getSuperclass().str());
	clClass = clClass.getSuperclass();
}

var classes_Field = clClass.getDeclaredField("classes");
var classes_Value = classes_Field.get(classLoader);

var elements_Method = classes_Value.getClass().getMethod("elements",null);
var elements = elements_Method.invoke(classes_Value,null);


var classes = new List<java.lang.Class>();
while (elements.Invoke<bool>("hasMoreElements","()Z"))
{
	classes.add(elements.Invoke<java.lang.Class>("nextElement", "()Ljava/lang/Object;"));
}
var treeView = topPanel.add_Panel(true).add_TreeView_with_PropertyGrid(false).sort();
Action<List<TreeNode>, Color> colorNodes =
	(nodes, color)=>{
					treeView.beginUpdate();
					foreach(var node in nodes)
						node.foreColor(color);
					treeView.endUpdate();
				};

treeView.beforeExpand<java.lang.Class>(
	(treeNode, @class)=>{
						var ctors = @class.getDeclaredConstructors();
						var methods = @class.getDeclaredMethods();
						var fields = @class.getDeclaredFields();
						var superClass = @class.getSuperclass();
						//var annotations = @class.getDeclaredAnotations();

						if (superClass.notNull()) treeNode.add_Node("SuperClass")

.add_Node(superClass.getName(), superClass, true).foreColor(Color.Orange);;

						if (ctors.size() > 0)
							treeNode.add_Node("ctors ({0})".format(ctors.size
()), ctors,true);
						if (methods.size() > 0)
							treeNode.add_Node("methods ({0})".format
(methods.size()), methods,true);
						if (fields.size() > 0)
							treeNode.add_Node("fields ({0})".format(fields.size
()), fields,true);
					});

treeView.beforeExpand<java.lang.reflect.Method[]>(
	(treeNode,methods)=>{
						treeNode.add_Nodes(methods, (method)=>"{0}
{1}".format(method.getName(),method.GetSignature()), true);
						colorNodes(treeNode.nodes(), Color.Blue);
					});
treeView.beforeExpand<java.lang.reflect.Constructor[]>(
	(treeNode,ctors) => {
						treeNode.add_Nodes(ctors, (ctor)=>"{0}
{1}".format(ctor.getName(),ctor.GetSignature()), true);
						colorNodes(treeNode.nodes(), Color.Blue);
					});

treeView.beforeExpand<java.lang.reflect.Field[]>(
	(treeNode,fields) => {
						treeNode.add_Nodes(fields, (field)=>"{0}
{1}".format(field.getName(),field.GetSignature()), true);
						colorNodes(treeNode.nodes(), Color.Blue);
					});

treeView.beforeExpand<java.lang.reflect.Method>(
	(treeNode,method)=>{
						treeNode.add_Node("ToString: {0}".format
```

```
(method.toGenericString()), method).foreColor(Color.DarkGray);
                                                    treeNode.add_Node("Java Signature: {0}".format
(method.GetSignature()), method).foreColor(Color.DarkGray);;
                                                    var parameterNode = treeNode.add_Node("_Parameters");
                                                    parameterNode.add_Nodes(method.getParameterTypes(),
(@class)=>@class.getName(),true);
                                                    colorNodes(parameterNode.nodes(), Color.DarkOrange);
                                                    treeNode.add_Node("_ReturnType").add_Node
(method.getReturnType().getName(), method.getReturnType(),true)

.foreColor(Color.DarkOrange);
                                        });

treeView.beforeExpand<java.lang.reflect.Constructor>(
       (treeNode,ctor)=>{
                                                    treeNode.add_Node("ToString: {0}".format
(ctor.toGenericString()), ctor).foreColor(Color.DarkGray);
                                                    treeNode.add_Node("Java Signature: {0}".format
(ctor.GetSignature()), ctor).foreColor(Color.DarkGray);;
                                                    var parameterNode = treeNode.add_Node("_Parameters");
                                                    parameterNode.add_Nodes(ctor.getParameterTypes(), (@class)
=>@class.getName(),true);
                                                    colorNodes(parameterNode.nodes(), Color.DarkOrange);


                                        });

treeView.beforeExpand<java.lang.reflect.Field>(
       (treeNode,field)=>{
                                                    treeNode.add_Node("ToString: {0}".format
(field.toGenericString()), field).foreColor(Color.DarkGray);
                                                    treeNode.add_Node("Java Signature: {0}".format
(field.GetSignature()), field).foreColor(Color.DarkGray);;
                                        });

treeView.beginUpdate().add_Nodes(classes, (@class)=>@class.getName(), true)
           .endUpdate();

colorNodes(treeView.nodes(), Color.DarkOrange);

return "done";


//using java.lang
//using net.sf.jni4net.jni
//using net.sf.jni4net.adaptors
//O2Ref:jni4net.n-0.8.6.0.dll
```



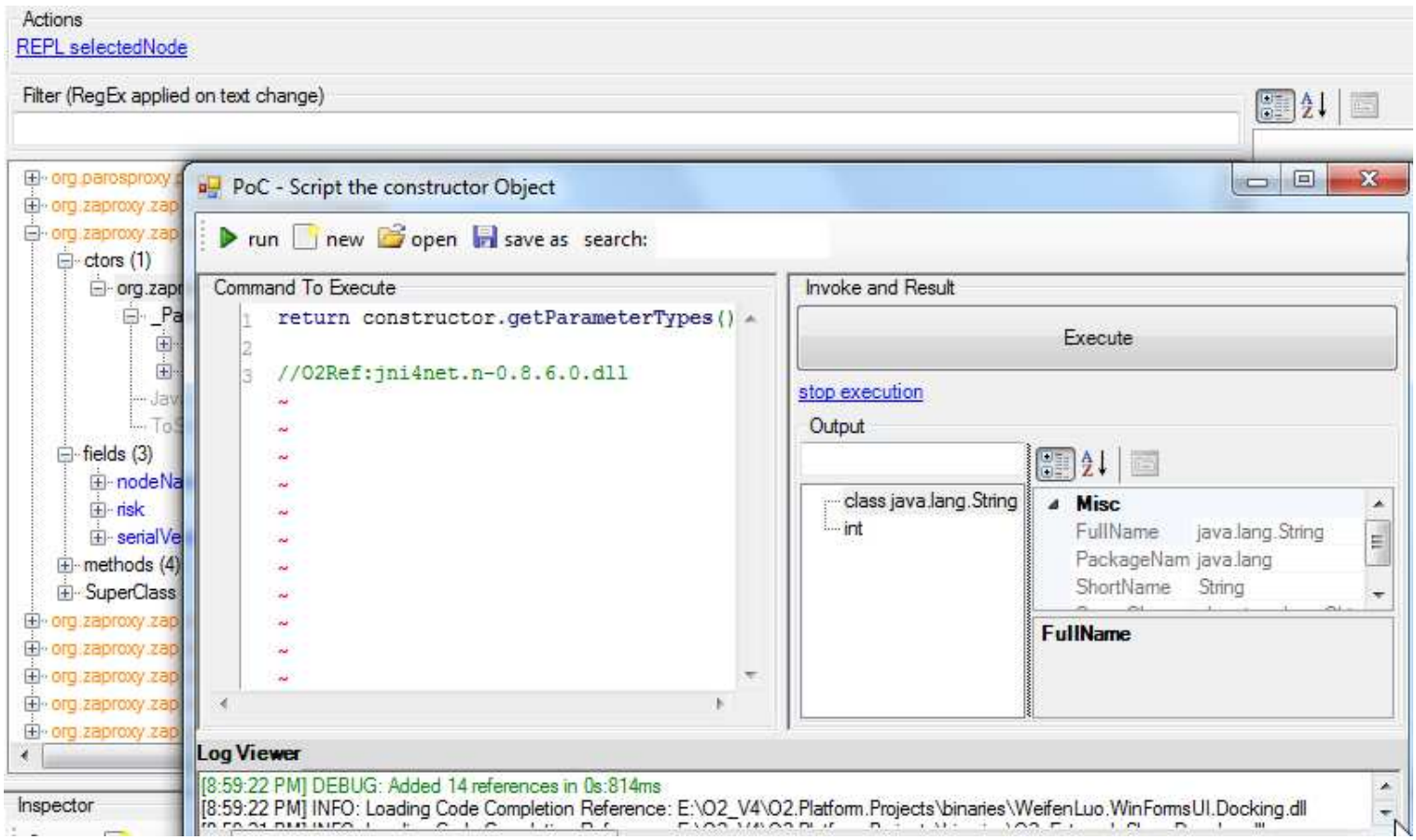
Adding a class filter

```
Action<string> showClasses =
       (filter)=>{
                     var classesToShow =  (filter.notValid())
                                              ? classes
                                              : classes.Where((@class)=>@class.getName().str().regEx
(filter));
                     treeView.beginUpdate()
                               .clear()
                               .add_Nodes(classesToShow, (@class)=>@class.getName(), true)
                              .endUpdate();

                     colorNodes(treeView.nodes(), Color.DarkOrange);
                 };

treeView.insert_Above(40,"Filter (RegEx applied on text change)")
                     .add_TextBox().fill().onTextChange(showClasses);
showClasses("");
```



Adding a link to open a REPL C# script editor on the current Node:

```
var actionPanel = topPanel.insert_ActionPanel();
var treeView = topPanel.add_Panel(true).add_TreeView_with_PropertyGrid(false).sort();
actionPanel.add_Link("REPL selectedNode", ()=> treeView.selected().Tag.script_Me());
```

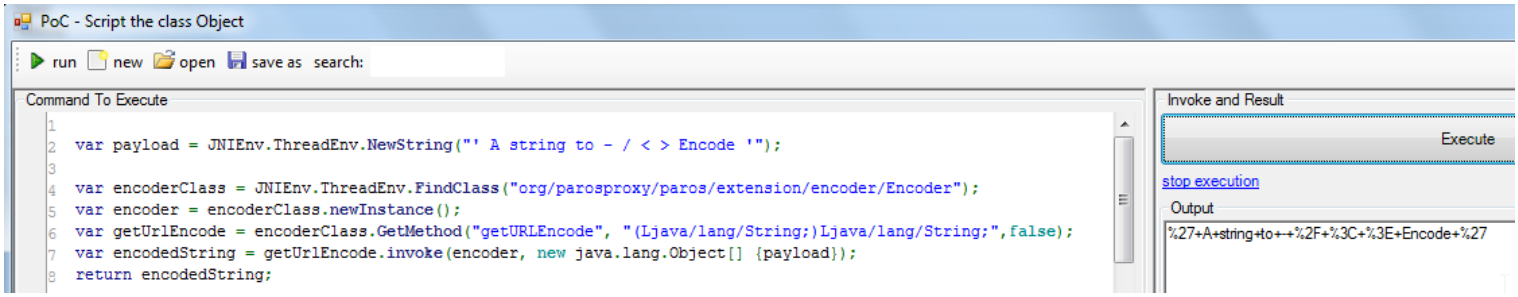


Creating a Java Class and invoking its method from C# REPL

The code below will :

- get the class ***org.parosproxy.paros.extension.encoder.Encoder***
- create a new instance of it (as the ***encoder*** variable)
- get the ***getUrlEncode*** method
- invoke the ***getUrlEncode*** with the ***payload*** string (which will execute on the JVM)

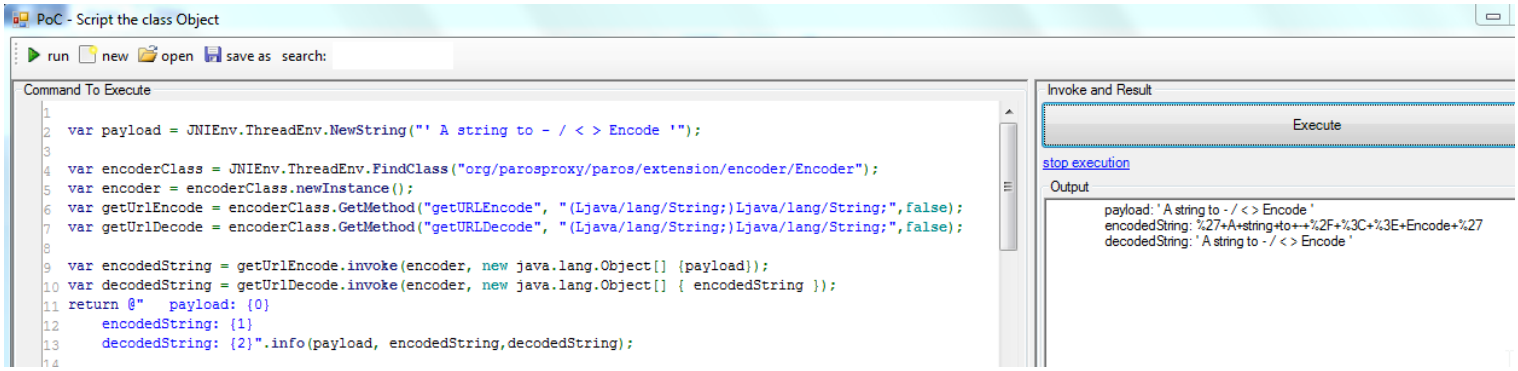
```
var payload = JNIEnv.ThreadEnv.NewString("' A string to - / < > Encode '");

var encoderClass = JNIEnv.ThreadEnv.FindClass("org/parosproxy/paros/extension/encoder/Encoder");
var encoder = encoderClass.newInstance();
var getUrlEncode = encoderClass.GetMethod("getURLEncode", "(Ljava/lang/String;)
Ljava/lang/String;",false);
var encodedString = getUrlEncode.invoke(encoder, new java.lang.Object[] {payload});
return encodedString;
```



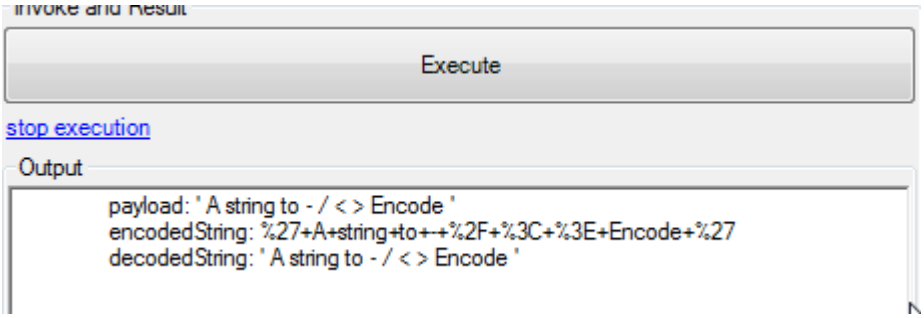
Creating a Java Class and invoking two methods from C# REPL

The code below will call first the ***getUrlEncode*** JVM method with the ***payload*** value and feed its result to the ***getUrlDecode*** JVM method:

```
var payload = JNIEnv.ThreadEnv.NewString("' A string to - / < > Encode '");

var encoderClass = JNIEnv.ThreadEnv.FindClass("org/parosproxy/paros/extension/encoder/Encoder");
var encoder = encoderClass.newInstance();
var getUrlEncode = encoderClass.GetMethod("getURLEncode", "(Ljava/lang/String;)
Ljava/lang/String;",false);
var getUrlDecode = encoderClass.GetMethod("getURLDecode", "(Ljava/lang/String;)
Ljava/lang/String;",false);

var encodedString = getUrlEncode.invoke(encoder, new java.lang.Object[] {payload});
var decodedString = getUrlDecode.invoke(encoder, new java.lang.Object[] { encodedString });
return @"    payload: {0}
      encodedString: {1}
      decodedString: {2}".info(payload, encodedString,decodedString);

//O2Ref:jni4net.n-0.8.6.0.dll
//using net.sf.jni4net.jni;
//using net.sf.jni4net.utils;
```

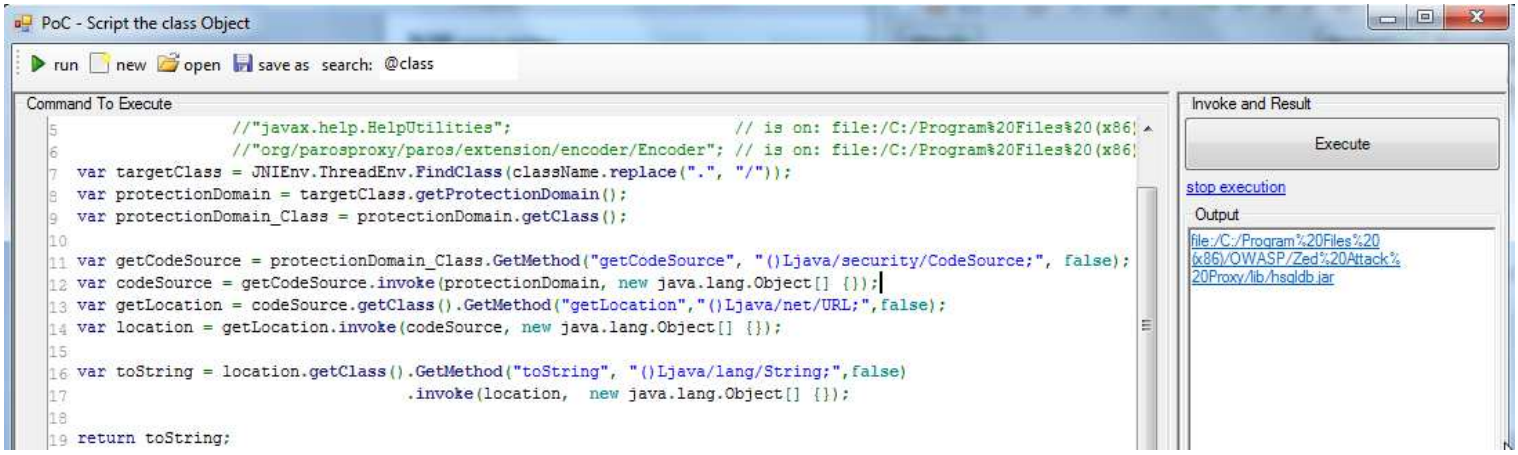


the end result is the same value :



Finding the source location of a particular class:

```
//var protectionDomain = @class.getProtectionDomain();
//var targetClass = JNIEnv.ThreadEnv.FindClass();
var className = "org.hsqldb.persist.Log";                                    // is on:
file:/C:/Program%20Files%20(x86)/OWASP/Zed%20Attack%20Proxy/lib/hsqldb.jar
                        //"system.String";                                                  //
is on: file:/C:/Program%20Files%20(x86)/OWASP/Zed%20Attack%20Proxy/lib/jni4net.j-0.8.6.0.jar
                        //"javax.help.HelpUtilities";                          // is on:
file:/C:/Program%20Files%20(x86)/OWASP/Zed%20Attack%20Proxy/lib/jh.jar
                        //"org/parosproxy/paros/extension/encoder/Encoder"; // is on:
file:/C:/Program%20Files%20(x86)/OWASP/Zed%20Attack%20Proxy/zap.jar
var targetClass = JNIEnv.ThreadEnv.FindClass(className.replace(".", "/"));
var protectionDomain = targetClass.getProtectionDomain();
var protectionDomain_Class = protectionDomain.getClass();

var getCodeSource = protectionDomain_Class.GetMethod("getCodeSource", "()Ljava/security/CodeSource;",
false);
var codeSource = getCodeSource.invoke(protectionDomain, new java.lang.Object[] {});
var getLocation = codeSource.getClass().GetMethod("getLocation","()Ljava/net/URL;",false);
var location = getLocation.invoke(codeSource, new java.lang.Object[] {});

var toString = location.getClass().GetMethod("toString", "()Ljava/lang/String;",false)
                                                    .invoke(location,  new java.lang.Object[] {});

return toString;

//how I found (for example) the signature of the getCodeSource method
//var _method = protectionDomain_Class.getMethods().Where((method)=>method.getName()
=="getCodeSource").first();
//return _method.GetSignature();


//using net.sf.jni4net.jni;

//O2Ref:jni4net.n-0.8.6.0.dll
```



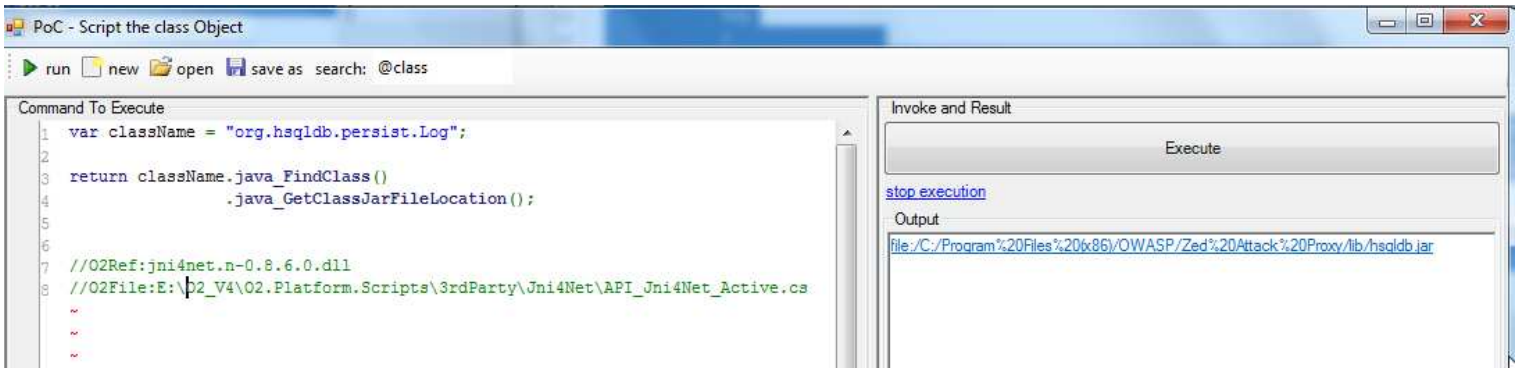
The code above can be refactored into a code-behind file

```
// This file is part of the OWASP O2 Platform (http://www.owasp.org/index.php/OWASP_O2_Platform) and is
released under the Apache 2.0 License (http://www.apache.org/licenses/LICENSE-2.0)
using System;
using O2.Kernel;
using O2.DotNetWrappers.ExtensionMethods;
```

```
using java.lang;
using net.sf.jni4net;
using net.sf.jni4net.jni;
//Installer:Jni4Net_Installer.cs!Jni4Net/bin/proxygen.exe
//O2Ref:Jni4Net/lib/jni4net.n-0.8.6.0.dll

namespace O2.XRules.Database.APIs
{
	public class API_Jni4Net_Active
	{

	}

	public static class API_Jni4Net_Active_ExtensionMethods
	{
		public static Class java_FindClass(this string className)
		{
			try
			{
				return JNIEnv.ThreadEnv.FindClass(className.replace(".", "/"));
			}
			catch(System.Exception ex)
			{
				ex.log();
				return null;
			}
		}

		public static string java_GetClassJarFileLocation(this Class targetClass)
		{
			try
			{
				var protectionDomain = targetClass.getProtectionDomain();
				var protectionDomain_Class = protectionDomain.getClass();

				var getCodeSource = protectionDomain_Class.GetMethod("getCodeSource", "()
Ljava/security/CodeSource;", false);
				var codeSource = getCodeSource.invoke(protectionDomain, new java.lang.Object[]
{});
				var getLocation = codeSource.getClass().GetMethod("getLocation","()
Ljava/net/URL;",false);
				var location = getLocation.invoke(codeSource, new java.lang.Object[] {});

				var toString = location.getClass().GetMethod("toString", "()
Ljava/lang/String;",false)
                                                                    .invoke(location,  new
java.lang.Object[] {});
				return toString.str();
			}
			catch(System.Exception ex)
			{
				ex.log("in getClassJarFileLocation");
				return null;
			}
		}
	}
}
```

which can be consumed like this:

```
var className = "org.hsqldb.persist.Log";

return className.java_FindClass()
                .java_GetClassJarFileLocation();


//O2Ref:jni4net.n-0.8.6.0.dll
//O2File:E:\O2_V4\O2.Platform.Scripts\3rdParty\Jni4Net\API_Jni4Net_Active.cs
```



After some refactoring, the size of the tool is greatly reduced:

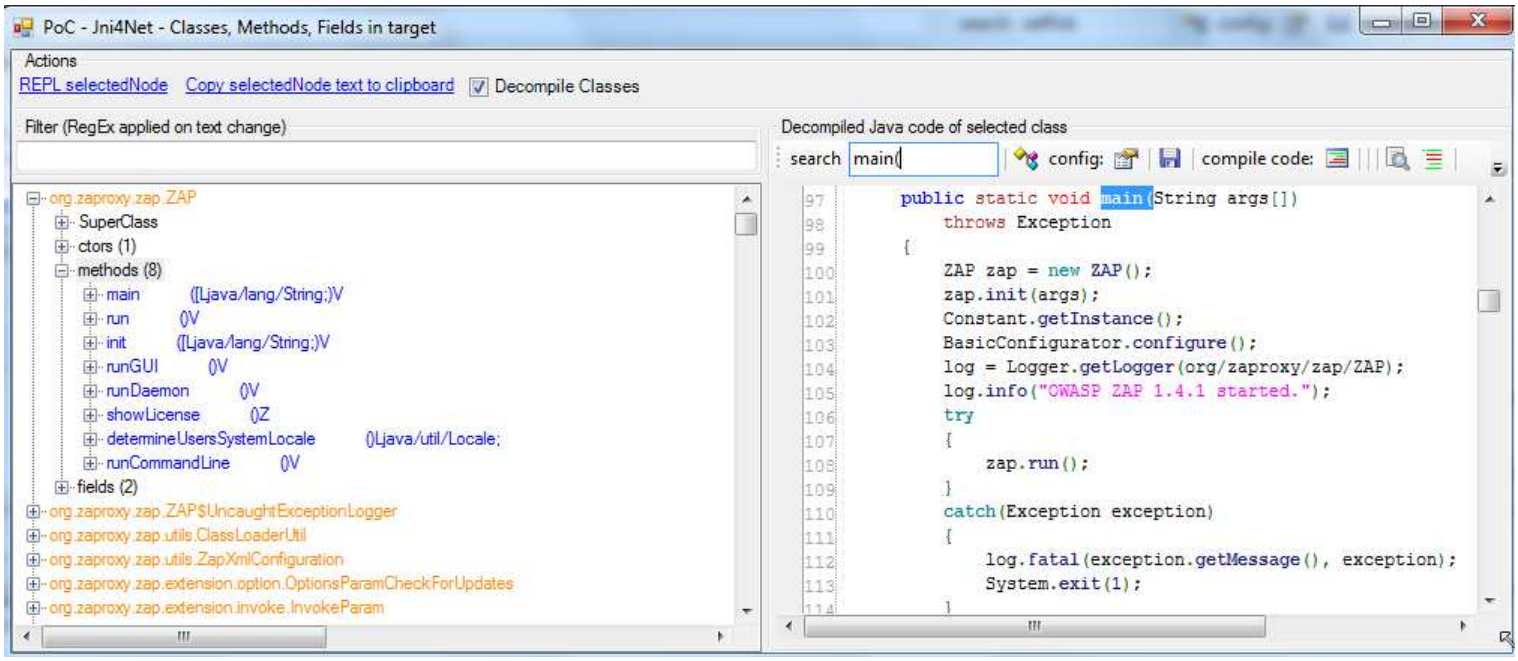
```
Bridge.CreateJVM( new BridgeSetup(){Verbose=true}); //only needs to be called once

//var topPanel = "PoC - Jni4Net - Classes, Methods, Fields in target".popupWindow();
var topPanel = panel.add_Panel(true);

var actionPanel = topPanel.insert_ActionPanel();
var decompileClasses = false;
var treeView = topPanel.insert_Left().add_TreeView_with_PropertyGrid().sort();
var codeViewer = topPanel.title("Decompiled Java code of selected class").add_SourceCodeViewer();

actionPanel.add_Link("REPL selectedNode", ()=> treeView.selected().Tag.script_Me())
           .append_Link("Copy selectedNode text to clipboard", ()=> treeView.selected().str
().clipboardText_Set())
           .append_CheckBox("Decompile Classes", (value)=> decompileClasses = value).autoSize().tick();

treeView.java_SetTreeView_To_Show_Jni4Net_Reflection_Data();

var jni4Net_Active = new API_Jni4Net_Active();

var classes = jni4Net_Active.java_From_ClassLoader_get_Loaded_Classes();

Action<string> showClasses =
       (filter)=>{
                     var classesToShow =  (filter.notValid())
                           ? classes
                           : classes.Where((@class)=>@class.getName().str().regEx(filter));
                     treeView.beginUpdate()
                              .clear()
                              .add_Nodes(classesToShow, (@class)=>@class.getName(), true)
                            .endUpdate();

                     treeView.nodes().colorNodes(Color.DarkOrange);
                  };

treeView.insert_Above(40,"Filter (RegEx applied on text change)")
                     .add_TextBox().fill().onTextChange(showClasses);

showClasses("zap");

return "done";


//using java.lang
//using net.sf.jni4net;
//using net.sf.jni4net.jni
//O2Ref:jni4net.n-0.8.6.0.dll

//O2File:E:\O2_V4\O2.Platform.Scripts\3rdParty\Jni4Net\API_Jni4Net_Active.cs
```

Adding java decompilation support via the API_Jad.cs

```
treeView.afterSelect<Class>(
       (@class) => {
                                    codeViewer.enabled(false);
                                    O2Thread.mtaThread(
                                           ()=>{
                                                        var jad = new API_Jad();
```

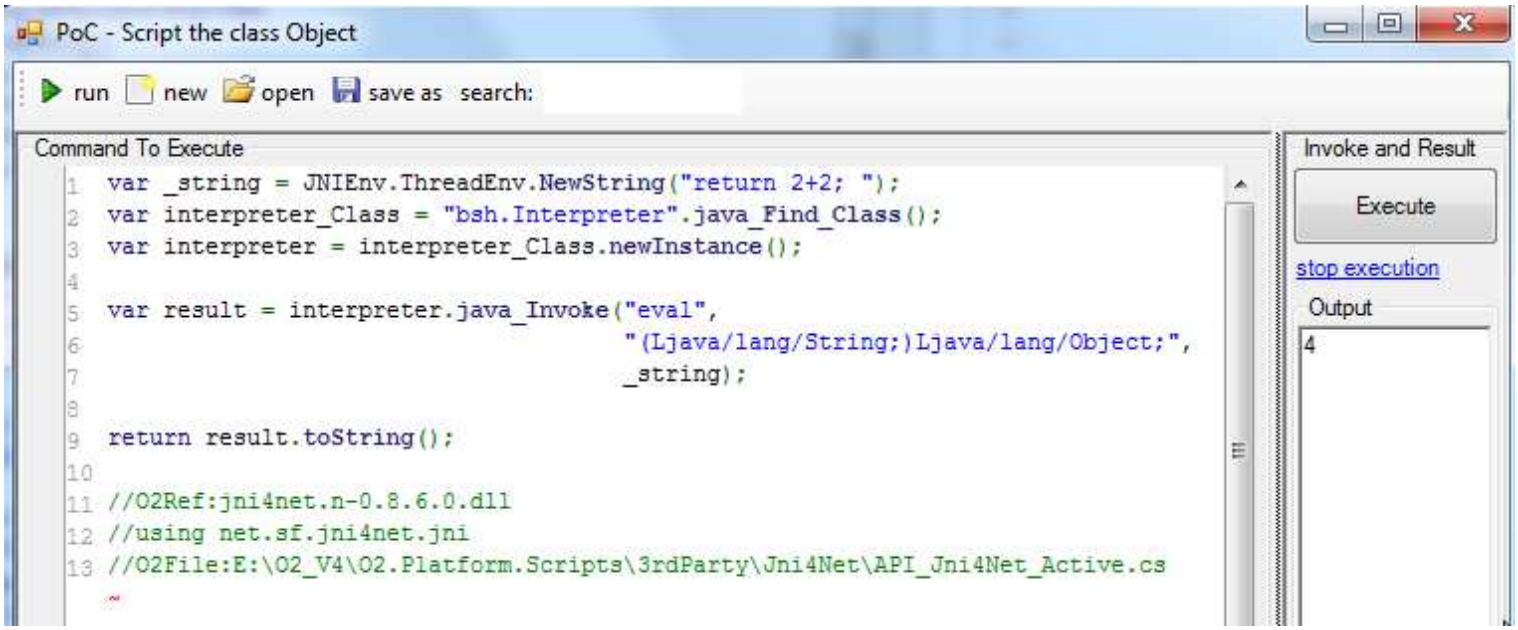
```
                                                        if (decompileClasses)
                                                        {
                                                                var pathToJar =
@class.java_Get_Class_Jar_File_Location();
                                                                if (pathToJar.valid())
                                                                {
                                                                        var extractedFiles =
jad.extractJarIntoTempFolder(pathToJar);
                                                                        var classFile =
extractedFiles.pathCombine(@class.FullName.replace(".","\\") + ".class");
                                                                        var javaCode = jad.decompile
(classFile);
                                                                        codeViewer.set_Text(javaCode,
".java");
                                                                }
                                                                else
                                                                {
                                                                        codeViewer.set_Text("Error: Could not
get location for class: {0}".format(@class.FullName));
                                                                }

                                                        }
                                                        codeViewer.enabled(true);
                                                });

                                });
```

Which when executed looks like this:



Invoking BeanShell from C# REPL

```
var _string = JNIEnv.ThreadEnv.NewString("return 2+2; ");
var interpreter_Class = "bsh.Interpreter".java_Find_Class();
var interpreter = interpreter_Class.newInstance();

var result = interpreter.java_Invoke("eval",
                                                        "(Ljava/lang/String;)Ljava/lang/Object;",
                                                        _string);

return result.toString();

//O2Ref:jni4net.n-0.8.6.0.dll
//using net.sf.jni4net.jni
//O2File:E:\O2_V4\O2.Platform.Scripts\3rdParty\Jni4Net\API_Jni4Net_Active.cs
```



See next document**: Using Jni4Net - Controling ZAP remotely**

.