cyberintegrator XML guide

for use with the java executor

Preface

Cyberintegrator's Java Executor is the environment by which Java tools in Cyberintegrator get executed. The Java Executor uses XML scripts to specify execution options and other settings needed to run the Java tool. This guide is a concise, simple approach to XML scripting for Cyberintegrator.

Introduction: Instantiating a class in Cyberintegrator

A Cyberintegrator XML script starts and ends with the following commands.

<classToInstantiate className = “INSERT CLASS NAME HERE”>

// **XML SCRIPT HERE**

</classToInstantiate>

The classToInstantiate command does exactly what it sounds like. It chooses the class that you want the Java executor to instantiate. The className modifier is where you specify the name of the class that you want to instantiate. Say you have a class called fooBar. The appropriate command then would be:

<classToInstantiate className = “fooBar”>

//**SCRIPT HERE**

</classToInstantiate>

Easy enough.

==========

Part 2: Invoking Methods

A class is nothing without its methods. To invoke a method, or method(s), you use the following syntax:

<methods>
    <method static = “EITHER TRUE OR FALSE” methodName = “METHOD NAME”>

    // **REST OF SCRIPT GOES HERE**

    </method>
</methods>

When you are invoking a method, you **MUST** specify whether it is a static method or not. To do this, either set static = true, or static = false. Then specify the method's name in methodName, and you are good to go.

Part 2a: Method Arguments

Cyberintegrator needs to know a few things about the method before it can execute it, namely, the arguments that you are going to pass to the method.

There are 3 components of an argument; the refid, which is provided to you by Cyberintegrator, the parameter switch, and the type of the argument. For example, if you had a method that took in a String parameter, you would do the following:

<arguments>

<argument refid = "INSERT REFID HERE" parameter = "true" className = "java.lang.String" />

// **REST OF SCRIPT GOES HERE**

</arguments>

An important, useful thing to remember is that the Java executor only likes to take in fully qualified class names. Here are a few examples.

String	java.lang.String
File	java.io.File
Integer	java.lang.Integer
Double	java.lang.Double
Long	java.lang.Long
Boolean	java.lang.Boolean

You can add as many arguments as you would like this way.

Part 2b: Specifying the return type.

The last component of the method is the return type and the refid. The refid is just a link to where the method will deposit its output. To specify the refid and type, you use the following syntax, placed after the **</arguments>** closing brace:

<return refid = "INSERT REFID HERE" className = "java.lang.String"/>

Once again, you must specify fully qualified class names here. Refer to the table above to find the appropriate class name for your type.

After you specify the return refid, you close off the method with the following command:
</method>

Part 3: Closing it up.

To finalize your scripts, after you have added all the methods you want and need, your script should look something like this. For purpose of example, we are going to use this Java class:

```
public class fooBar{

	public static void main(String[] args){

	}

	public static String sayHello(String name)
	{

		return “Hello, “ + name;
	}

}
```

The finalized XML script for this function would be this:

```
<classToInstantiate className = “fooBar”>
	<methods>
		<method static = “true” methodName = “sayHello”>
			<arguments>
				<argument refid = “refid” parameter = “true” className = “java.lang.String”/>
			</arguments>
			<return refid = “refid” className = “java.lang.String”/>
		</method>
	</methods>
</classToInstantiate>
```

You should now be able to effectively script in XML for Cyberintegrator's java executor.