JAVA CORE SERVICE CLIENT & TOM.JAVA

Mihai Cădariu

yatb.mitza.net

About Mihai

Principal Consultant, SDL WCM, San Jose, USA

Joined Tridion in 2005, Amsterdam, NL

Previously Java developer

Yet Another Tridion Blog (yatb.mitza.net)

Agenda

- Java Core Service Client
  - Proxy Generation
  - JAXB Custom Bindings
  - Bindings and Endpoints
  - Known Limitations
  - Demo
- TOM.Java
  - JNI
  - jni4net
  - ProxyGen
  - Demo

Java Core Service Client

Proxy Generation

- Core Service is a Web Communication Foundation (WCF) web service
- Proxy (client) classes for each Core Service data model
- .NET client proxies OOTB or trivial to generate in Visual Studio
- What about Java?

Which Java Framework?

- Plethora of Java web service tools available
  - WSIT (Web Services Interoperability Technologies)
  - JAX-WS (Java API for XML Web Services)
  - JAX-RS (Java API for RESTful Web Services)
  - Apache Axis
  - Apache CXF
  - Metro
  - …
- Less is more
  - Simple, standard, out-of-the-box, free?

wsimport.exe

- Part of and shipped with the JDK
- Command line
- Simple to use
- Point to WSDL (Web Service Definition Language) location and generate proxies

```
wsimport.exe http://localhost/webservices/CoreService2012.svc?wsdl
```

Demo

- Generation of simplest Java Core Service client

Custom JAXB Bindings

- JAXB – Java Architecture for XML Binding
- Do not generate the JAXB element
- Provide proper conversion for Date objects

```
<jxb:globalBindings generateElementProperty="false">
    <jxb:javaType name="java.util.Date" xmlType="xs:dateTime"
        parseMethod="mitza.coreservice.util.Utils.parseStringToDate“
        printMethod="mitza.coreservice.util.Utils.parseDateToString" />
</jxb:globalBindings>
```

- Use custom bindings XML

```
wsimport.exe -b custombindings.xml
    http://localhost/webservices/CoreService2012.svc?wsdl
```

Demo

- Generate client with custom bindings

Full Java Core Service Client

- **CoreServiceFactory** class
- Provides authentication (BasicHttp)

```
CoreServiceFactory.setDefault(“user", "password");
```

- Possible to point to specific Core Service URL

```
CoreServiceFactory.setDefault("http://staging.cme.example.com");
```

- Provides easy access to different endpoints

```
CoreServiceFactory.getBasicHttpClient();
CoreServiceFactory.getBatchBasicHttpClient();
CoreServiceFactory.getStreamDownloadBasicHttpClientClient();
CoreServiceFactory.getStreamUploadBasicHttpClientClient();
```

Full Java Core Service Client

- Available online (link)
  - core-service-client-2011sp1hr1.jar
  - core-service-client-2013ga.jar

- http://yatb.mitza.net/2013/04/a-java-core-service-client-for-sdl.html
- http://yatb.mitza.net/2013/04/streamlining-core-service-java-client.html

Demo

- Full Java Core Service Client

Known Limitations

- WsHttp endpoint not supported
  - Metro WS has a potential implementation (incomplete)
  - NTLM authentication not implemented
- BasicHttp authentication not working on server (remote only)

Resources

- http://yatb.mitza.net/search?q=java+core+service
- https://code.google.com/p/yet-another-tridion-blog
- https://wsit.java.net
- https://metro.java.net

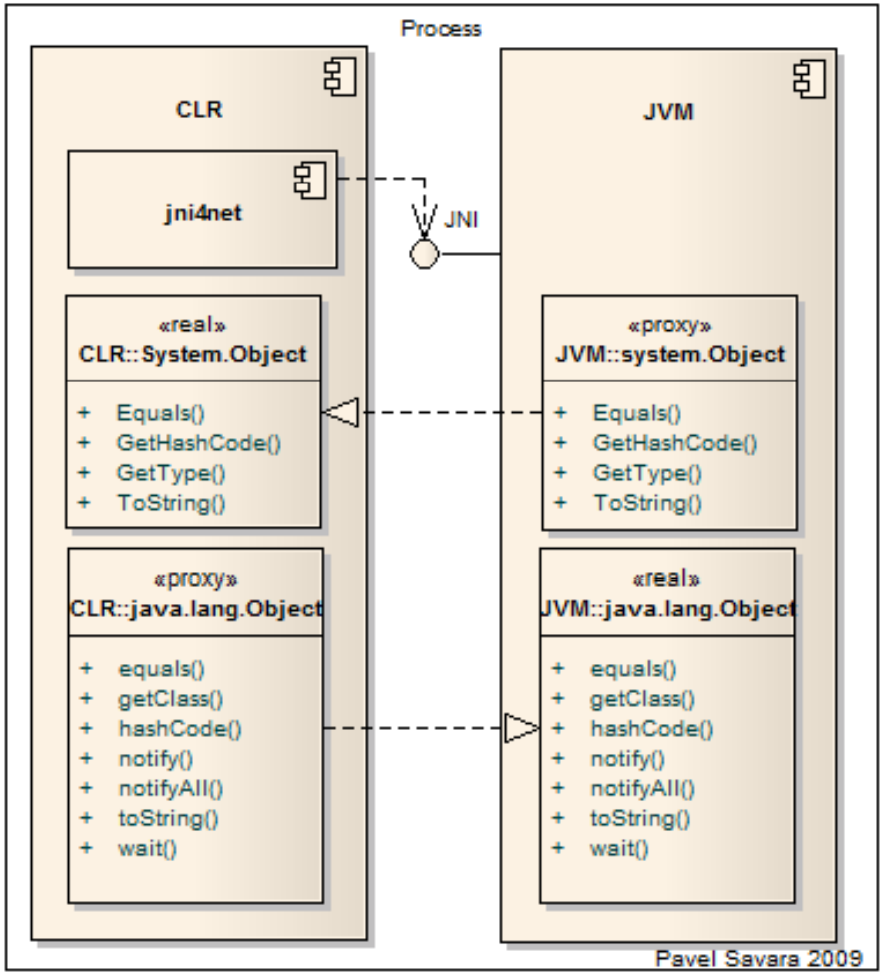
TOM.Java

Still TOM.NET… only inside Java

- Tridion Object Model (TOM) .NET API
- Execute .NET inside Java Virtual Machine (JVM)
- Java Native Interface (JNI)
- Intra-process execution (JVM inside CLR or CLR inside JVM)
- CLR – Common Language Runtime
- Proxy classes needed on the “other” side
- Real objects have “proxy” counterparts
- Need Java proxy classes from .NET classes

jni4net

- jni4net.sourceforge.net
- “bridge between Java and .NET”
- Thank you, Pavel Šavara!



ProxyGen

- Generates Java proxies from .NET DLL
- Types mapping (Java->.NET and .NET->Java)
- Primitives mapping – OOTB
- Collection mapping – possible through wrappers
- Generics – not implemented
- TOM.NET uses generic collections

```
public IList<ComponentPresentation> ComponentPresentations { get; }
```

Build Your Own ProxyGen

- Thank you, Open Source!
- Added (limited) generics support to ProxyGen
  - https://groups.google.com/forum/#!topic/jni4net/FZwTetghskA
  - https://jni4net.googlecode.com/svn/branches/0-8-generics/
- Able to generate full TOM.NET API to Java Proxies

Collection Mapping

Java	C#
java.util.List<E>	System.Collections.Generic.IList<T>
java.util.Map<K, V>	System.Collections.Generic.IDictionary<K, V>
java.util.Set<E>	System.Collections.Generic.ISet<T>
java.util.Collection<E>	System.Collections.Generic.ICollection<T>
java.lang.Iterable<E>	System.Collections.Generic.IEnumerable<T>

Resources

- http://yatb.mitza.net/2012/11/tomjava-now-with-generics-support.html
- https://code.google.com/p/yet-another-tridion-blog/source/browse/trunk/Tom.Java
- http://jni4net.sourceforge.net
- https://groups.google.com/forum/#!forum/jni4net

Demo

Questions & Answers

That's all, folks!