

Web Services with XFire & XMLBeans

A Java Web Services Stack

David Lucek
dave@lucek.com
August 3, 2006

XMLBeans

- Originally from BEA
- Latest version is 2.2, June 2006
- xmlbeans.apache.org
- Why use it? One of the few open source tools that fully supports the XML Schema standard.
- Reads full XML instance and uses it as a backing buffer.
- XML is not parsed until accessed, until a `getXXX()` method is called.

Common Java SOAP Stacks

- Axis 1.x
 - Generated Model is built into the product
 - Can not marshal and un-marshal independently.
(i.e. Read flat files)
 - Its Slow
- XFire
 - Fixes most of the issues with Axis 1
 - Has Spring and JSR 181 support
- Axis 2
 - Looks good, fixes most issues with Axis 1
 - Version 1.0 came out in May
 - No Spring or JSR 181 support yet

Hints on Web Services Development

- Forget the WSDL
 - It was good in the RPC days, not needed now.
 - They are always a pain to work with.
 - Let a tool generate it.
- Do Schema Development First
 - Easiest
 - Cleanest
 - Easiest to version and manage
- Do not give a WSDL to your customers
 - If you have to, use imported schema's inside the WSDL

XFire Project

- Java Web Services Stack
- xfire.codehaus.org
- Only for Document Literal Web Services
- Supports SOAP, WSDL generation, etc..
- High Performance
- Uses Stax
- Supports Many Data Bindings
 - POJO
 - XMLBeans
 - JAXB 1.1/2.0
-

XFire Project

- Supports many transports, HTTP, JMS, etc
- Supports many containers, Spring, Pico
- Why use it? It already does most of what Axis 2 will do.
- Currently has better Spring and JSR 181 support than Axis 2
- Currently used by ServiceMix and Mule projects
- Current Version is 1.1.2
- They have some good testing support fixtures.

XFire Usage Scenarios

- Use XFireServlet and the services.xml file
 - Simplest way to get up and running.
- Use XFireSpringServlet with Spring configuration files.
- Using JSR 181 annotations (Web Service Annotations) with Spring
 - This is probably the best scenario

XFire/XMLBeans Project Setup

- Get XML beans
- Generate XML Beans Model
- Get Spring
- Get XFire
- Update web.xml
 - Spring config
 - XFire config
- Update the XFire Spring config file
 - This is where you add your POJOs

XFire/XMLBeans Demo & Conclusions

- See “xfiredemo” project
- I liked XFire over Axis 2 because of Spring and JSR 181 support.
 - CoC (Convention over Configuration)
- Axis 2 does support attachments and will gain a large audience because of Axis 1
- The goal should be:
 - Inter-changeable SOAP stacks
 - Independent data binding technology that can be used outside of the stack.