XMLBeans

Apache XMLBeans is another technology for mapping XML Schema to java objects. CXF added support for XMLBeans in 2.1. Note that this data-binding has been removed from Apache CXF 3.2.0 onwards, as Apache XMLBeans has been retired to the attic.

There are a two parts to the support for XMLBeans:

Code Generation

The wsdl2java tool now allows a "-db xmlbeans" flag to be added that will generate XMLBeans types for all the schema beans instead of the default JAXB beans. With 2.1 and 2.2, the types are generated, but you still need to configure the XMLBeans databinding to be used at runtime. With 2.3, the generated code contains an @Databinding annotation marking it as XMLBeans and the configuration is unnecessary.

Runtime

You need to configure the runtime to tell it to use XMLBeans for the databinding instead of JAXB.

Spring config

For the server side, your spring configuration would contain something like:

```
<jaxws:server serviceClass="demo.hw.server.HelloWorld" address="/hello_world">
    <jaxws:dataBinding>
       <bean class="org.apache.cxf.xmlbeans.XmlBeansDataBinding" />
    </jaxws:dataBinding>
</jaxws:server>
```

or

```
<jaxws:endpoint
    id="helloWorld"
    implementor="demo.spring.HelloWorldImpl"
    address="http://localhost/HelloWorld">
    <jaxws:dataBinding>
       <bean class="org.apache.cxf.xmlbeans.XmlBeansDataBinding" />
    </jaxws:dataBinding>
</jaxws:endpoint>
```

The client side is very similar:

```
<jaxws:client id="helloClient"
                  serviceClass="demo.spring.HelloWorld"
                  address="http://localhost:9002/HelloWorld">
    <jaxws:dataBinding>
       <bean class="org.apache.cxf.xmlbeans.XmlBeansDataBinding" />
    </jaxws:dataBinding>
<jaxws:client>
```

FactoryBeans

If using programmatic factory beans instead of spring configuration, the databinding can be set on the ClientProxyFactoryBean (and subclasses) and the ServerFactoryBean (and subclasses) via:

```
factory.getServiceFactory().setDataBinding(new org.apache.cxf.xmlbeans.XmlBeansDataBinding());
```