

XBean-Spring

Dain Sundstrom
Chief Architect, IBM Gluecode

Agenda

- * The Problem
- * Our Solution
- * Extra Tricks
- * Implementation

The Problem

```
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">
<beans>
  <bean id="restaurant" class="org.apache.xbean.spring.example.RestaurantService">
    <property name="favourite">
      <bean class="org.apache.xbean.spring.example.PizzaService">
        <property name="topping" value="Salami" />
        <property name="cheese" value="Edam" />
        <property name="size" value="17" />
      </bean>
    </property>
    <property name="dinnerMenu">
      <list>
        <bean class="org.apache.xbean.spring.example.PizzaService">
          <property name="topping" value="Ham" />
          <property name="cheese" value="Mozzarella" />
          <property name="size" value="15" />
        </bean>
        <bean class="org.apache.xbean.spring.example.PizzaService">
          <property name="topping" value="Eggs" />
          <property name="cheese" value="Mozzarella" />
          <property name="size" value="16" />
        </bean>
      </list>
    </property>
    <property name="lunchMenu">
      <list>
        <bean class="org.apache.xbean.spring.example.PizzaService">
          <property name="topping" value="Chicken" />
          <property name="cheese" value="Brie" />
          <property name="size" value="17" />
        </bean>
      </list>
    </property>
  </bean>
</beans>
```

Configuration Complexity

- * Heaps of XML
 - * Generic XML - Bean, Prop & ref
 - * No schema
 - * Need source code
-
- * Basically, you need to be a programmer

The Solution

```
<beans >
  <restaurant id="restaurant"
    xmlns="http://xbean.apache.org/schemas/pizza-simple">
    <dinnerMenu>
      <pizza myTopping="Ham" cheese="Mozzarella" size="15"/>
      <pizza myTopping="Eggs" cheese="Mozzarella" size="16"/>
    </dinnerMenu>

    <lunchMenu>
      <pizza myTopping="Chicken" cheese="Brie" size="17"/>
    </lunchMenu>

    <favourite>
      <pizza myTopping="Salami" cheese="Edam" size="17"/>
    </favourite>
  </restaurant>
</beans>
```

Generic XML

```
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"  
    "http://www.springframework.org/dtd/spring-beans.dtd">  
<beans>  
  <bean id="pizzaService" class="org.apache.xbean.spring.example.PizzaService">  
    <property name="topping" value="Salami"/>  
    <property name="cheese" value="Edam"/>  
    <property name="size" value="17"/>  
  </bean>  
</beans>
```

- * No schema
- * Can't use standard XML tools
- * Must read the code

Generic XML

```
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"  
    "http://www.springframework.org/dtd/spring-beans.dtd">  
<beans>  
  <bean id="pizzaService" class="org.apache.xbean.spring.example.PizzaService">  
    <property name="topping" value="Salami"/>  
    <property name="cheese" value="Edam"/>  
    <property name="size" value="17"/>  
  </bean>  
</beans>
```

```
<beans xmlns:p="http://xbean.apache.org/schemas/pizza">  
  <p:pizza id="pizzaService" myTopping="Salami" cheese="Edam" size="17"/>  
</beans>
```

Constructors

```
<beans>  
  <bean id="saladService" class="org.apache.xbean.spring.example.SaladService">  
    <constructor-arg index="0" value="Caesar"/>  
    <constructor-arg index="1" value="Small"/>  
    <constructor-arg index="2" value="true"/>  
  </bean>  
</beans>
```

- * Indexed arguments
- * Difficult to keep in sync
- * Again, read the code

Constructors

```
<beans>  
  <bean id="saladService" class="org.apache.xbean.spring.example.SaladService">  
    <constructor-arg index="0" value="Caesar"/>  
    <constructor-arg index="1" value="Small"/>  
    <constructor-arg index="2" value="true"/>  
  </bean>  
</beans>
```

```
<beans xmlns:s="http://xbean.apache.org/schemas/salad">  
  <s:salad id="saladService" dressing="Caesar" size="Small" addCroutons="true"/>  
</beans>
```

Lists

```
<beans>
  <bean id="restaurant"
    class="org.apache.xbean.spring.example.RestaurantService">
    <property name="dinnerMenu">
      <list>
        <bean class="org.apache.xbean.spring.example.PizzaService">
          <property name="topping" value="Ham" />
          <property name="cheese" value="Mozzarella" />
          <property name="size" value="15" />
        </bean>
        <bean class="org.apache.xbean.spring.example.PizzaService">
          <property name="topping" value="Eggs" />
          <property name="cheese" value="Mozzarella" />
          <property name="size" value="16" />
        </bean>
      </list>
    </property>
  </bean>
</beans>
```

Lists

```
<beans>
  <restaurant id="restaurant"
    xmlns="http://xbean.apache.org/schemas/pizza-simple">
    <dinnerMenu>
      <pizza myTopping="Ham" cheese="Mozzarella" size="15"/>
      <pizza myTopping="Eggs" cheese="Mozzarella" size="16"/>
    </dinnerMenu>
  </restaurant>
</beans>
```

Nested Bean

```
<beans>
  <bean id="restaurant"
    class="org.apache.xbean.spring.example.RestaurantService">
    <property name="favourite">
      <bean class="org.apache.xbean.spring.example.PizzaService">
        <property name="topping" value="Salami" />
        <property name="cheese" value="Edam" />
        <property name="size" value="17" />
      </bean>
    </property>
  </bean>
</beans>
```

Nested Bean

```
<beans>
  <restaurant id="restaurant"
    xmlns="http://xbean.apache.org/schemas/pizza-simple">
    <favourite>
      <pizza myTopping="Salami" cheese="Edam" size="17"/>
    </favourite>
  </restaurant>
</beans>
```

Extra Tricks

Root Element

```
<restaurant id="restaurant"
  xmlns="http://xbean.apache.org/schemas/restaurant">
  <dinnerMenu>
    <pizza myTopping="Ham" cheese="Mozzarella" size="15"/>
    <pizza myTopping="Eggs" cheese="Mozzarella" size="16"/>
  </dinnerMenu>

  <lunchMenu>
    <pizza myTopping="Chicken" cheese="Brie" size="17"/>
  </lunchMenu>

  <favourite>
    <pizza myTopping="Salami" cheese="Edam" size="17"/>
  </favourite>
</restaurant>
```

#REF

```
<beans xmlns:p="http://xbean.apache.org/schemas/pizza">  
  <p:pizza id="pizzaService" myTopping="#dailySpecial" cheese="Edam" size="17"/>  
</beans>
```

#null

```
<beans xmlns="http://xbean.apache.org/schemas/salad">  
  <salad id="saladService" dressing="#null" size="Small" addCroutons="true"/>  
</beans>
```


Default Property

```
<beans xmlns:b="http://xbean.apache.org/schemas/pizza">  
  <b:gin id="ginService">Bombay Sapphire</b:gin>  
</beans>
```

Custom Property Editor

```
<beans xmlns="http://xbean.apache.org/schemas/pizza">  
  <keg id="ml1000" remaining="1000 ml"/>  
  <keg id="pints5" remaining="5 pints"/>  
  <keg id="liter20" remaining="20 liter"/>  
  <keg id="empty" remaining="0"/>  
</beans>
```

Nested .properties

```
<beans xmlns:p="java://org.apache.xbean.spring.example">  
  <p:PizzaService id="pizzaService">  
    cheese    Edam  
    topping   Salami  
    size      17  
  </p:PizzaService>  
</beans>
```

Implementation

Our Implementation

- * Is backwards compatible
- * Works with Spring v1 and v2
- * Generates the schema
- * Generates html documentations
- * Generates confluence documentation

How do you implement it?

- * JavaDoc annotations
- * Source preprocessor
- * Custom ApplicationContext (Spring v1)

Element

```
@org.apache.xbean.XBean element="name" rootElement="true"  
  namespace="http://some.com/schemas/application"  
  description="For the docs"
```

Life-cycle Methods

```
@org.apache.xbean.FactoryMethod  
@org.apache.xbean.InitMethod  
@org.apache.xbean.DestroyMethod
```

Properties

```
@org.apache.xbean.Property alias="name"  
  propertyEditor="custom.PropertyEditor"
```

Lists

```
@org.apache.xbean.FlatCollection childElement="nestedElementName"  
@org.apache.xbean.Property nestedType="nested.Type"
```

Nested Property

```
@org.apache.xbean.Flat
```

Ant Task

```
<taskdef classname="org.apache.xbean.spring.generator.MappingGeneratorTask"
  name="mapping">
  <classpath>
    <pathelement path="{basedir}/target/classes"/>
    <pathelement path="{basedir}/target/test-classes"/>
    <path refid="maven.runtime.classpath"/>
  </classpath>
</taskdef>

<mapping namespace="http://xbean.apache.org/schemas/pizza"
  srcdir="src/test/java"
  destFile="target/restaurant.xsd"
  metaInfDir="{basedir}/target/test-generated"/>

<copy todir="{basedir}/target/test-classes">
  <fileset dir="{basedir}/target/test-generated"/>
</copy>
```


Maven Plugin

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.xbean</groupId>
      <artifactId>maven-xbean-plugin</artifactId>
      <version>2.7-SNAPSHOT</version>
      <executions>
        <execution>
          <goals>
            <goal>mapping</goal>
          </goals>
        </execution>
      </executions>
      <configuration>
        <namespace>http://xbean.apache.org/schemas/server</namespace>
      </configuration>
    </plugin>
  </plugins>
</build>
```

Custom ApplicationContext

```
org.apache.xbean.spring.context.ClassPathXmlApplicationContext  
org.apache.xbean.spring.context.FileSystemXmlApplicationContext  
org.apache.xbean.spring.context.ResourceXmlApplicationContext  
org.apache.xbean.spring.context.XmlWebApplicationContext
```

* Spring 1

- * Simply change import

* Spring 2

- * Automatically added

About XBean

About XBean

- * Subproject of Apache Geronimo
- * Light-weight server environment
- * Used by ActiveMQ, ServiceMix, XFire, OpenEJB, Jencks and Jetty

Questions?

* geronimo.apache.org/xbean