

**git\_comments:****git\_commits:**

1. **summary:** Merged revisions 1433011 via git cherry-pick from  
**message:** Merged revisions 1433011 via git cherry-pick from <https://svn.apache.org/repos/asf/cxf/branches/2.7.x-fixes> .....  
r1433011 | dkulp | 2013-01-14 12:58:39 -0500 (Mon, 14 Jan 2013) | 10 lines Merged revisions 1433007 via git cherry-pick from  
<https://svn.apache.org/repos/asf/cxf/trunk> ..... r1433007 | dkulp | 2013-01-14 12:44:36 -0500 (Mon, 14 Jan 2013) | 2 lines [CXF-4750] Allow XMLBeans to work with enums ..... git-svn-id: <https://svn.apache.org/repos/asf/cxf/branches/2.6.x-fixes>@1433068 13f79535-47bb-0310-9956-ffa450edef68

**github\_issues:****github\_issues\_comments:****github\_pulls:****github\_pulls\_comments:****github\_pulls\_reviews:****jira\_issues:**

1. **summary:** Wrong return type for enumeration in generated service interface using xmlbeans  
**description:** I have a WSDL which has an enumeration as the response type for a SOAP operation. From that, I am generating Java classes using the cxf-codegen-plugin with xmlbeans. The generated service interface has a String as response type. If my implementation returns a String containing one of the specified enum values, I get an Exception (using CXF 2.3.11):  
org.apache.cxf.interceptor.Fault at  
org.apache.cxf.databinding.AbstractWrapperHelper.createWrapperObject(AbstractWrapperHelper.java:107) at  
org.apache.cxf.jaxws.interceptors.WrapperClassOutInterceptor.handleMessage(WrapperClassOutInterceptor.java:105) at  
org.apache.cxf.phase.PhaseInterceptorChain.doIntercept(PhaseInterceptorChain.java:255) at  
org.apache.cxf.interceptor.OutgoingChainInterceptor.handleMessage(OutgoingChainInterceptor.java:77) at  
org.apache.cxf.phase.PhaseInterceptorChain.doIntercept(PhaseInterceptorChain.java:255) at  
org.apache.cxf.transport.ChainInitiationObserver.onMessage(ChainInitiationObserver.java:113) at  
org.apache.cxf.transport.servlet.ServletDestination.invoke(ServletDestination.java:102) at  
org.apache.cxf.transport.servlet.ServletController.invokeDestination(ServletController.java:464) at  
org.apache.cxf.transport.servlet.ServletController.invoke(ServletController.java:188) at  
org.apache.cxf.transport.servlet.AbstractCXFServlet.invoke(AbstractCXFServlet.java:148) at  
org.apache.cxf.transport.servlet.AbstractHTTPServlet.handleRequest(AbstractHTTPServlet.java:179) at  
org.apache.cxf.transport.servlet.AbstractHTTPServlet.doPost(AbstractHTTPServlet.java:103) at  
javax.servlet.http.HttpServlet.service(HttpServlet.java:751) at  
org.apache.cxf.transport.servlet.AbstractHTTPServlet.service(AbstractHTTPServlet.java:159) If I manually change the return type of the generated interface and my implementation from String to "org.apache.test.ETestResponseType.Enum", everything works well. I could only test the wsdl2java generation with a newer CXF version, but 2.7.2 also generates a String as the response type. I will attach a sample project.

**jira\_issues\_comments:**

1. Sample project
2. **body:** Thanks for the quick fix! Sadly it also changed the behavior when Enums are used as input parameters, where I get exceptions now. These had a String type previously, which was fine, and also need an Enum now. Here's a stacktrace of 2.5.9-SNAPSHOT: org.apache.cxf.interceptor.Fault: argument type mismatch while invoking public abstract org.apache.test.MyResponseType org.apache.test.TestServicePortType.testFunction2(org.apache.test.ETestResponseType\$Enum) with params [<xml-fragment>ONE</xml-fragment>]. at  
org.apache.cxf.service.invoker.AbstractInvoker.createFault(AbstractInvoker.java:159) at  
org.apache.cxf.jaxws.AbstractJAXWSMethodInvoker.createFault(AbstractJAXWSMethodInvoker.java:86) at  
org.apache.cxf.service.invoker.AbstractInvoker.invoke(AbstractInvoker.java:133) at  
org.apache.cxf.jaxws.JAXWSMethodInvoker.invoke(JAXWSMethodInvoker.java:66) at  
org.apache.cxf.service.invoker.AbstractInvoker.invoke(AbstractInvoker.java:75) at  
org.apache.cxf.interceptor.ServiceInvokerInterceptor\$1.run(ServiceInvokerInterceptor.java:58) at  
java.util.concurrent.Executors\$RunnableAdapter.call(Executors.java:439) at  
java.util.concurrent.FutureTask\$Sync.innerRun(FutureTask.java:303) at  
java.util.concurrent.FutureTask.run(FutureTask.java:138) at  
org.apache.cxf.workqueue.SynchronousExecutor.execute(SynchronousExecutor.java:37) at  
org.apache.cxf.interceptor.ServiceInvokerInterceptor.handleMessage(ServiceInvokerInterceptor.java:107) at  
org.apache.cxf.phase.PhaseInterceptorChain.doIntercept(PhaseInterceptorChain.java:263) at  
org.apache.cxf.transport.ChainInitiationObserver.onMessage(ChainInitiationObserver.java:122) at  
org.apache.cxf.transport.http.AbstractHTTPDestination.invoke(AbstractHTTPDestination.java:233) at  
org.apache.cxf.transport.servlet.ServletController.invokeDestination(ServletController.java:209) at  
org.apache.cxf.transport.servlet.ServletController.invoke(ServletController.java:189) at  
org.apache.cxf.transport.servlet.CXFNonSpringServlet.invoke(CXFNonSpringServlet.java:129) at  
org.apache.cxf.transport.servlet.AbstractHTTPServlet.handleRequest(AbstractHTTPServlet.java:223) at

org.apache.cxf.transport.servlet.AbstractHTTPServlet.doPost(AbstractHTTPServlet.java:143) at  
javax.servlet.http.HttpServlet.service(HttpServlet.java:751) at  
org.apache.cxf.transport.servlet.AbstractHTTPServlet.service(AbstractHTTPServlet.java:199) ... Caused by:  
java.lang.IllegalArgumentException: argument type mismatch at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at  
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at  
java.lang.reflect.Method.invoke(Method.java:597) at  
org.apache.cxf.service.invoker.AbstractInvoker.performInvocation(AbstractInvoker.java:173) at  
org.apache.cxf.service.invoker.AbstractInvoker.invoke(AbstractInvoker.java:89) ... 44 more

**label:** code-design

3. We'll likely need to see a full testcase. The test I added does a complete echo of the enum:  
<https://fisheye6.atlassian.com/browse/cxf/trunk/systests/databinding/src/test/java/org/apache/cxf/systest/xmlbeans/GreeterImpl.java?r2=1433007&r1=1127624> which works. Thus, it looks like the incoming enum should work OK based on the testcase.
4. Hi, I created two test projects, one using CXF 2.7.2, the other one using CXF 2.7.3-SNAPSHOT, which I will attach. You can run them via Maven, first "mvn clean install" and then "mvn jetty:run". The projects only differ in the CXF version in the pom.xml and in the implementation of the generated service interface. I use soapUI for generating test requests from the wsdl and for requesting the server. The problematic request is called "sampleMethod" and looks like the following: 

```
{code:xml} <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:test="http://test.apache.org/"> <soapenv:Header/> <soapenv:Body> <test:sampleMethod> <sampleValue>LEFT</sampleValue> </test:sampleMethod> </soapenv:Body> </soapenv:Envelope> {code}
```

 It works on 2.7.2, but does not on 2.7.3-SNAPSHOT where I get the exception from above. For completeness, I also added the previous testFunction, which works on 2.7.3-SNAPSHOT, but does not on 2.7.2.
5. OK. This now works. If you have a type that extends an enumeration, XmlSchemas doesn't map those to an enumeration. I've updated the code generator to use the same type mapping in this case. For example, in the SampleMethodDocument.java, the getSampleValue() call returns an ESample and not an Enum. Thus, we need to make sure we use that ESample type instead.