

Hi,

I have been successfully deploying Web services to Talend Service Factory as OSGi bundles for some time.

But I have been given a new WSDL to build a service for that I can not get to run, this WSDL uses MTOM for large binary attachments.

```
<xs:element name="data" ns1:expectedContentTypes="application/octet-stream" type="xs:base64Binary" minOccurs="0" xmlns:ns1="http://www.w3.org/2005/05/xmlmime"/>
```

(I also tried adding the xmlmime namespace in the WSDL header, but it made no difference to the issue).

When I run the wsdl2java tool it correctly (according to MTOM docs) changes the type of the binary data from byte[] to:

```
@XmlMimeType("application/octet-stream")
```

```
javax.activation.DataHandler;
```

I deploy the web service as two separate bundles, one containing all the stub classes and one containing the service implementation and a spring beans deployment descriptor (project package names edited):

```
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:sgi="http://www.springframework.org/schema/osgi"
xmlns:jaxws="http://cxf.apache.org/jaxws"
xmlns:soap="http://cxf.apache.org/bindings/soap"
xsi:schemaLocation="
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
http://www.springframework.org/schema/osgi
http://www.springframework.org/schema/osgi/spring-osgi.xsd
http://cxf.apache.org/jaxws
http://cxf.apache.org/schemas/jaxws.xsd">
```

```
<import resource="classpath:META-INF/cxf/cxf.xml" />
<import resource="classpath:META-INF/cxf/cxf-extension-soap.xml" />
<import resource="classpath:META-INF/cxf/osgi/cxf-extension-osgi.xml" />
```

```
<jaxws:endpoint xmlns:s="http://api.myservice.company.org/"
serviceName="s:MyAPIService"
endpointName="s:MyAPIPort"
id="myapi_soap"
implementor="#my-service-bean"
address="/my-soap">
<jaxws:properties>
<entry key="mtom-enabled" value="true"/>
</jaxws:properties>
</jaxws:endpoint>
```

```
<bean id="my-service-bean" class="org.company.myservice.impl.MyAPIImpl">
</bean>
</beans>
```

After deploying to TSF I get the following stack trace:

```
karaf@tsf> Exception in thread "SpringOsgiExtenderThread-4" org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'my_soap': Invocation of in
it method failed; nested exception is javax.xml.ws.WebServiceException: org.apache.cxf.service.factory.ServiceConstructionException
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:1420)
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:519)
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:456)
at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:291)
at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:222)
at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:288)
```

```
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:190)
at org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.java:580)
at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:895)
at org.springframework.osgi.context.support.AbstractDelegatedExecutionApplicationContext.access$1600(AbstractDelegatedExecutionApplicationContext.java:69)
at org.springframework.osgi.context.support.AbstractDelegatedExecutionApplicationContext$4.run(AbstractDelegatedExecutionApplicationContext.java:355)
at org.springframework.osgi.util.internal.PrivilegedUtils.executeWithCustomTCCL(PrivilegedUtils.java:85)
at org.springframework.osgi.context.support.AbstractDelegatedExecutionApplicationContext.completeRefresh(AbstractDelegatedExecutionApplicationContext.java:320)
at org.springframework.osgi.extender.internal.dependencies.startup.DependencyWaiterApplicationContextExecutor$CompleteRefreshTask.run(DependencyWaiterApplicationContextExec
utor.java:132)
at java.lang.Thread.run(Thread.java:662)
Caused by: javax.xml.ws.WebServiceException: org.apache.cxf.service.factory.ServiceConstructionException
at org.apache.cxf.jaxws.EndpointImpl.doPublish(EndpointImpl.java:350)
at org.apache.cxf.jaxws.EndpointImpl.publish(EndpointImpl.java:239)
at org.apache.cxf.jaxws.EndpointImpl.publish(EndpointImpl.java:509)
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.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.invokeCustomInitMethod(AbstractAutowireCapableBeanFactory.java:1544)
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.invokeInitMethods(AbstractAutowireCapableBeanFactory.java:1485)
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:1417)
... 14 more
Caused by: org.apache.cxf.service.factory.ServiceConstructionException
at org.apache.cxf.jaxb.JAXBDataBinding.initialize(JAXBDataBinding.java:292)
at org.apache.cxf.service.factory.AbstractServiceFactoryBean.initializeDataBindings(AbstractServiceFactoryBean.java:86)
at org.apache.cxf.service.factory.ReflectionServiceFactoryBean.buildServiceFromWSDL(ReflectionServiceFactoryBean.java:400)
at org.apache.cxf.service.factory.ReflectionServiceFactoryBean.initializeServiceModel(ReflectionServiceFactoryBean.java:499)
at org.apache.cxf.service.factory.ReflectionServiceFactoryBean.create(ReflectionServiceFactoryBean.java:241)
at org.apache.cxf.jaxws.support.JaxWsServiceFactoryBean.create(JaxWsServiceFactoryBean.java:202)
at org.apache.cxf.frontend.AbstractWSDLBasedEndpointFactory.createEndpoint(AbstractWSDLBasedEndpointFactory.java:101)
at org.apache.cxf.frontend.ServerFactoryBean.create(ServerFactoryBean.java:157)
at org.apache.cxf.jaxws.JaxWsServerFactoryBean.create(JaxWsServerFactoryBean.java:202)
at org.apache.cxf.jaxws.EndpointImpl.getServer(EndpointImpl.java:433)
at org.apache.cxf.jaxws.EndpointImpl.doPublish(EndpointImpl.java:322)
... 23 more
Caused by: com.sun.xml.bind.v2.runtime.IllegalAnnotationsException: 1 counts of IllegalAnnotationExceptions
javax.activation.DataHandler does not have a no-arg default constructor.
this problem is related to the following location:
at javax.activation.DataHandler
at protected javax.activation.DataHandler org.company.myService.api.SubmitJob.data
at org.company.myService.api.SubmitJob

at com.sun.xml.bind.v2.runtime.IllegalAnnotationsException$Builder.check(IllegalAnnotationsException.java:102)
at com.sun.xml.bind.v2.runtime.JAXBContextImpl.getTypeInfoSet(JAXBContextImpl.java:472)
at com.sun.xml.bind.v2.runtime.JAXBContextImpl.<init>(JAXBContextImpl.java:302)
at com.sun.xml.bind.v2.runtime.JAXBContextImpl$JAXBContextBuilder.build(JAXBContextImpl.java:1140)
at com.sun.xml.bind.v2.ContextFactory.createContext(ContextFactory.java:154)
at com.sun.xml.bind.v2.ContextFactory.createContext(ContextFactory.java:121)
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 javax.xml.bind.ContextFinder.newInstance(ContextFinder.java:213)
```

```
at javax.xml.bind.ContextFinder.find(ContextFinder.java:387)
at javax.xml.bind.JAXBContext.newInstance(JAXBContext.java:618)
at org.apache.cxf.jaxb.JAXBContextCache.createContext(JAXBContextCache.java:258)
at org.apache.cxf.jaxb.JAXBContextCache.getCachedContextAndSchemas(JAXBContextCache.java:167)
at org.apache.cxf.jaxb.JAXBDataBinding.createJAXBContextAndSchemas(JAXBDataBinding.java:418)
at org.apache.cxf.jaxb.JAXBDataBinding.initialize(JAXBDataBinding.java:290)
... 33 more
```

If I remove the MTOM entry in the WSDL and build stubs that take a byte[] array build and deploy my bundles the service is exposed by CXF correctly.

The MTOM example provided with TSF appears to use a standalone server and not OSGi bundles.

Reading various blogs I think the issue is with the ire 1.6 implementations of JAXB and JAX-WS but my understanding is that TSF and ESB are preconfigured to use the correct CXF bundles and block the JRE classes vi the ire.properties file?

I have tried various hacks people have used for similar class path issues but so far nothing as helped at all.

TIA

Andy