

Hi Shong, I have managed to get this to work however i am certain this is not entirely correct.
WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="ImportSchema"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.talend.org/service/"
  targetNamespace="http://www.talend.org/service/">
  <wsdl:types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://www.talend.org/service/"
      xmlns:pref="http://www.dstv.com/services/DStvTypes/">
      <xsd:import schemaLocation="schemas/DStvTypes.xsd" namespace="http://www.dstv.com/services/DStvTypes/"></xsd:import>
      <xsd:element name="ImportSchemaOperationRequest">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="in" type="xsd:string"></xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="ImportSchemaOperationResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="out" type="xsd:string"></xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="ImportSchemaOperationRequest">
    <wsdl:part name="parameters" element="tns:ImportSchemaOperationRequest"></wsdl:part>
  </wsdl:message>
  <wsdl:message name="ImportSchemaOperationResponse">
    <wsdl:part name="parameters" element="tns:ImportSchemaOperationResponse"></wsdl:part>
  </wsdl:message>
  <wsdl:portType name="ImportSchemaPortType">
    <wsdl:operation name="ImportSchemaOperation">
      <wsdl:input message="tns:ImportSchemaOperationRequest"></wsdl:input>
      <wsdl:output message="tns:ImportSchemaOperationResponse"></wsdl:output>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ImportSchemaBinding" type="tns:ImportSchemaPortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="ImportSchemaOperation">
      <soap:operation soapAction="http://www.talend.org/service/ImportSchemaOperation" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

```
<wsdl:service name="ImportSchema">
  <wsdl:port name="ImportSchemaPort" binding="tns:ImportSchemaBinding">
    <soap:address location="http://localhost:8090/services/ImportSchema" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

When i export the service i get a file not found exception.

```
2013-08-16 10:05:34,198 | INFO | xtenderThread-14 | OsgiServiceProxyFactoryBean | al.aop.ServiceDynamicInterceptor 476 | 249 - org.springframework
2013-08-16 10:05:34,403 | WARN | xtenderThread-14 | ReflectionServiceFactoryBean | org.springframework.ReflectionServiceFactoryBean 2605 | 262 - org.apache.c
javax.wsdl.WSDLException: WSDLException (at /wsdl:definitions/wsdl:types/xsd:schema): faultCode=PARSER_ERROR: Problem parsing 'schemas/DStvTypes.x
```

```
at com.ibm.wsdl.xml.WSDLReaderImpl.getDocument(Unknown Source)
at com.ibm.wsdl.xml.WSDLReaderImpl.parseSchema(Unknown Source)
at com.ibm.wsdl.xml.WSDLReaderImpl.parseSchema(Unknown Source)
at com.ibm.wsdl.xml.WSDLReaderImpl.parseTypes(Unknown Source)
at com.ibm.wsdl.xml.WSDLReaderImpl.parseDefinitions(Unknown Source)
at com.ibm.wsdl.xml.WSDLReaderImpl.readWSDL(Unknown Source)
at com.ibm.wsdl.xml.WSDLReaderImpl.readWSDL(Unknown Source)
at org.apache.cxf.wsdl11.WSDLManagerImpl.loadDefinition(WSDLManagerImpl.java:260)
at org.apache.cxf.wsdl11.WSDLManagerImpl.getDefinition(WSDLManagerImpl.java:205)
at org.apache.cxf.service.factory.ReflectionServiceFactoryBean.isEmptywsdl(ReflectionServiceFactoryBean.java:2603)
at org.apache.cxf.service.factory.ReflectionServiceFactoryBean.isFromWsd(ReflectionServiceFactoryBean.java:541)
at org.apache.cxf.service.factory.ReflectionServiceFactoryBean.initializeServiceModel(ReflectionServiceFactoryBean.java:545)
at org.apache.cxf.service.factory.ReflectionServiceFactoryBean.create(ReflectionServiceFactoryBean.java:263)
at org.apache.cxf.jaxws.support.JaxWsServiceFactoryBean.create(JaxWsServiceFactoryBean.java:205)
at org.apache.cxf.frontend.AbstractWSDLBasedEndpointFactory.createEndpoint(AbstractWSDLBasedEndpointFactory.java:102)
at org.apache.cxf.frontend.ServerFactoryBean.create(ServerFactoryBean.java:159)
at org.apache.cxf.jaxws.JaxWsServerFactoryBean.create(JaxWsServerFactoryBean.java:211)
at org.apache.cxf.jaxws.EndpointImpl.getServer(EndpointImpl.java:454)
at org.apache.cxf.jaxws.EndpointImpl.doPublish(EndpointImpl.java:334)
at org.apache.cxf.jaxws.EndpointImpl.publish(EndpointImpl.java:251)
at org.apache.cxf.jaxws.EndpointImpl.publish(EndpointImpl.java:537)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:601)
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.invokeCustomInitMethod(AbstractAutowireCapableBeanFactory.java:
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.invokeInitMethods(AbstractAutowireCapableBeanFactory.java:
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:145)
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:294)
at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:225)
at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:291)
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:193)
at org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.java:607)
```

```

at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:925)
at org.springframework.osgi.context.support.AbstractDelegatedExecutionApplicationContext.access$1600(AbstractDelegatedExecutionApplicatio
at org.springframework.osgi.context.support.AbstractDelegatedExecutionApplicationContext$4.run(AbstractDelegatedExecutionApplicationConte
at org.springframework.osgi.util.internal.PrivilegedUtils.executeWithCustomTCCL(PrivilegedUtils.java:85)
at org.springframework.osgi.context.support.AbstractDelegatedExecutionApplicationContext.completeRefresh(AbstractDelegatedExecutionApplicat
at org.springframework.osgi.extender.internal.dependencies.startup.DependencyWaiterApplicationContextExecutor$CompleteRefreshTask.run(Depend
at java.lang.Thread.run(Thread.java:722)
Caused by: java.io.FileNotFoundException: /home/graham/Apps/Talend-5.3.1/runtime/schemas/DStvTypes.xsd (No such file or directory)
at java.io.FileInputStream.open(Native Method)
at java.io.FileInputStream.<init>(FileInputStream.java:138)
at java.io.FileInputStream.<init>(FileInputStream.java:97)
at sun.net.www.protocol.file.FileURLConnection.connect(FileURLConnection.java:90)
at sun.net.www.protocol.file.FileURLConnection.getInputStream(FileURLConnection.java:188)
at org.apache.xerces.impl.XMLEntityManager.setupCurrentEntity(Unknown Source)
at org.apache.xerces.impl.XMLVersionDetector.determineDocVersion(Unknown Source)
at org.apache.xerces.parsers.XML11Configuration.parse(Unknown Source)
at org.apache.xerces.parsers.XML11Configuration.parse(Unknown Source)
at org.apache.xerces.parsers.XMLParser.parse(Unknown Source)
at org.apache.xerces.parsers.DOMParser.parse(Unknown Source)
at org.apache.xerces.jaxp.DocumentBuilderImpl.parse(Unknown Source)
... 42 more
karaf@trun>

```

The above error can be overcome by doing the following

1. In the IDE under the Services node create a new schemas folder
2. import the xsd file into it.
3. In the WSDL add an import as follows.

```
<xsd:import schemaLocation="DStvTypes.xsd" namespace="http://www.dstv.com/services/DStvTypes/"></xsd:import>
```

In the Runtime create the following folder \$TALEND_BASE/runtime/schemas

Place the DStvTypes.xsd file in this directory.

The server translates the WSDL as follows

```

<wsdl:definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://www.talend.org/serv
<wsdl:types>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://www.talend.org/service/"
<xsd:import namespace="http://www.dstv.com/services/DStvTypes/" schemaLocation="http://localhost:8040/services/ImportSchema?xsd=schemas/DStvTypes.
<xsd:element name="ImportSchemaOperationRequest">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="in" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="ImportSchemaOperationResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="out" type="xsd:string"/>

```

```
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>
<wsdl:message name="ImportSchemaOperationResponse">
<wsdl:part element="tns:ImportSchemaOperationResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="ImportSchemaOperationRequest">
<wsdl:part element="tns:ImportSchemaOperationRequest" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:portType name="ImportSchemaPortType">
<wsdl:operation name="ImportSchemaOperation">
<wsdl:input message="tns:ImportSchemaOperationRequest"></wsdl:input>
<wsdl:output message="tns:ImportSchemaOperationResponse"></wsdl:output>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ImportSchemaBinding" type="tns:ImportSchemaPortType">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="ImportSchemaOperation">
<soap:operation soapAction="http://www.talend.org/service/ImportSchemaOperation"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="ImportSchema">
<wsdl:port binding="tns:ImportSchemaBinding" name="ImportSchemaPort">
<soap:address location="http://localhost:8040/services/ImportSchema"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

Export the service and it starts up in the container

Import the WSDL in soapUI and call the operation above

SUCCESS 😊

Well almost. I cannot import the WSDL schemas into the repository for use. I need to build them up manually in the tXMLMap

I do suspect this is not correct though can you please confirm this?

The schemaLocation in the import can be changed however the runtime and Studio do both require the schema in the same relative location to work.

Graham