WSDL JMS Extension

1 Overview

WSIF defines extra WSDL extensions that are not part of WSDL4J itself. Amongst others, these WSDL extensions are needed for JMS. You should use WSIF to read in your WSDL, rather than using WSDL4J directly, because WSIF adds in its own extension registries that understand these extra WSDL extensions. For example you can use WSIFUtils.readWSDL to do this. This page describes WSIF's WSDL extensions for JMS. Currently, these WSDL extensions are valid for Soap over Jms, Axis over Jms and NativeJms. The jms namespace must be `xmlns:jms="http://schemas.xmlsoap.org/wsdl/jms/"`

2 Jms address

<jms:address describes a target port that is accessible via JMS.

```
<jms:address jmsVendorURI="xxx"
    jndiDestinationName="xxx"
    destinationStyle="xxx"
    jndiConnectionFactoryName="xxx"
    initialContextFactory="xxx"
    jndiProviderURL="xxx"
    jmsProviderDestinationName="xxx"
    jmsImplementationSpecificURI="xxx" />
```

- there must be exactly one jms:address in the port instead of the soap:address
- jmsVendorURI is optional and unused by WSIF. It allows to the client to check the JMS implementation.
- either destinationStyle or jmsImplementationSpecificURI must be specified, but not both.
- jmsImplementationSpecificURI specifies the queue manager and queues in a implementation specific format. This is currently unimplemented by WSIF.
- destinationStyle must either be queue or topic, but topics aren't yet implemented by WSIF.
- if destinationStyle is specified, then either jndiDestinationName or jmsProviderDestinationName must be specified but not both.
- jndiDestinationName is the JNDI name of the JMS queue that WSIF will send requests to.
- jmsProviderDestinationName is the JMS name of the JMS queue that WSIF will send requests to.

- if jndiDestinationName is specified then jndiConnectionFactoryName must also be specified.
- if jmsProviderDestinationName is specified then jndiConnectionFactoryName may also be specified. jndiConnectionFactoryName would be needed if the JNDI name of a replyTo queue is passed to WSIF.
- jndiConnectionFactoryName is the JNDI name of the connection factory that WSIF will use.
- if destinationStyle is specified then either both jndiProviderURL and initialContextFactory or neither must be specified.
- jndiProviderURL and initialContextFactory specify which JNDI database to use. If they are not present, WSIF uses the default JNDI.

WSIF uses the following order to lookup queues and queue managers in JNDI

- Lookup java:comp/env/<name> in the default (local) JNDI
- Lookup java:comp/env/<name> in the JNDI specified by the WSDL
- Lookup <name> in the default (local) JNDI
- Lookup <name> in the JNDI specified by the WSDL.

This allows a client administrator to override the JNDI definition specified in the WSDL.

3 Jms binding

The Jms binding is used to bind Native Jms.

```
<binding name="xxx" type="xxx">
  <jms:binding type="ObjectMessage"/>
  <format:typeMapping encoding="Java" style="Java">
     <format:typeMap typeName="xxx" formatType="xxx" />
  </format:typeMapping>
  <operation name="xxx">
    <input name="xxx" />
    <output name="xxx" />
  </operation>
</binding>
```

For information about the format:typeMapping and format:typeMap please see Format Type Mapping.

The jms:binding specifies that this binding is for Native Jms. The type is the type of the JMS message that will be sent. Currently only object messages are supported.

4 Jms property

```
<jms:property name="<name>" part="<part>" />
```

- this must go in the input, output or fault section of the binding operation. Output jms

properties are not yet implemented in WSIF.
- the <name> may be the name of a property defined by JMS, or the name of a property defined by the JMS implementation, or the name of a user property.
- the <part> must be the name of a part in the message.
- JMS user properties that are objects are not implemented by WSIF.
- When using stubs to invoke WSIF, this property appears as a parameter on the stub's signature, but will not appear on the web service's method signature.

5 Jms property value

```
<jms:propertyValue name="<name>" type="<type>"
value="<value>" />
```

- this must go in either the <jms:address or in the input section of the binding operation.
- the <name> may be the name of a property defined by JMS, or the name of a property defined by the JMS implementation, or the name of a user property.
- the <type> is the datatype of the <value> that hardcodes the value of this property in the WSDL.
- JMS user properties that are objects are not implemented by WSIF.

JMS properties can also be set on the message context, without being defined in the WSDL.

6 Jms fault, fault property and fault indicator

The JMS fault, fault property and fault indicator tags are WSDL extensions that describe the Native JMS binding for a fault message. These tags appear in the fault message of a JMS binding operation. These faults are problems that the backend web service wishes to report to the client. In Native JMS these are mapped to either body parts or JMS properties. There is a JMS property that is the fault indicator. If this fault indicator is set, this message describes a fault, not an output message. The fault indicator also shows which fault occurred.

```
<binding name="xxx" type="xxx">
  <jms:binding type="ObjectMessage"/>
  <format:typeMapping encoding="Java" style="Java" />
  <operation name="xxx">
    <input name="xxx" />
    <output name="xxx" />
    <fault name="xxx">
      <jms:faultIndicator type="property">
        <jms:faultProperty
          name="xxx"
          type="xxx"
          value="xxx"
          part="xxx"/>
      </jms:faultIndicator>
      <jms:fault parts="a b c"/>
      <jms:property
```

```
        name="xxx"
        type="xxx"
        part="xxx"/>
    </fault>
  </operation>
</binding>
```

- There can be multiple fault messages for a particular operation. Each fault message describes a different fault that this operation can return.
- The <jms:faultIndicator tag describes how to recognise this fault message. The type of the faultIndicator must be set to property which means that this faultIndicator is described by a JMS property. Other faultIndicator types may be supported in the future. WSIF currently supports each fault message having exactly one jms:faultIndicator.
- The <jms:faultProperty tag appears in the <jms:faultIndicator, if the faultIndicator type is property (which is the only value allowed currently). The jms:faultProperty describes the JMS property that indicates that this message is a fault. The faultProperty name is the name of the JMS property. The type is the type of the JMS property, which must also match the type of the message part that this faultProperty maps to. Also the value must be able to be converted into this type. The value is the value that the JMS property must be set to, if the message describes this fault. The part is the message part that maps to this faultProperty. The name, type and value are mandatory, whereas the part is optional. There cannot be two fault messages for the same operation that have the same faultProperty name and value.
- The <jms:fault tag describes which message parts will be set for this fault. The parts is a space separated list of message parts.
- For a description of the <jms:property tag, please see above.