



Mapping Message Parts To SOAP Header And Body Elements

Technical Note

© 2012 Active Endpoints Inc. ActiveVOS is a trademark of Active Endpoints, Inc. All other company and product names are the property of their respective owners.



2012

Content

Introduction	3
About the Project	3
Steps to deploy and execute the project:.....	6
Troubleshooting questions:	7
Reference.....	7
About Active Endpoints.....	8

Introduction

This technote guides users on how to map separate WSDL message parts to a SOAP header and its body.

About the Project

The ActiveVOS project accompanying this sample demonstrates the use of 'WSDL Binding Reference' partner policy to direct ActiveVOS to map header and body parts according to the mappings declared in the binding element in the WSDL for the service.

The sample takes two input parameters consumed by the 'InvokePartsInHeader' process and this in turn invoke another service – 'ReceivePartsInHeader' - by passing information in the SOAP header and its body. For this mechanism to work the WSDL must define a multipart message with both the header element and the body element defined as parts in the message.

WSDL Snippet:

```
...  
...  
<wsdl:message name="partInHeaderRequest">  
  <wsdl:part name="bodyPart" type="xs:string" />  
  <wsdl:part name="headerPart"  
    element="simple:ElementForHeaderRequest" /> </wsdl:message>  
...  
...
```

In the binding element of the WSDL, the header part is declared within <soap:header>. This is what informs the server to look in the header for that part, and for the outgoing message, map that part of the multipart message into the SOAP Header of the outgoing message.

WSDL Snippet:

```
...
...
<wsdl:binding name="RPCSoapBinding" type="tns:portTypeTwo">
  <soap:binding style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http"
  xml/>
  <wsdl:operation name="headerInvoke">
    <soap:operation soapAction="http://docs.active-
endpoints/activebpel/test/wsdl/wsi-headers/headerInvoke" style="rpc"
ns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
    <wsdl:input>
      <soap:body
        namespace="http://docs.active-
endpoints/activebpel/test/wsdl/wsi-headers/2006/03/wsi-headers.wsdl"
        use="literal"
        xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
      <soap:header message="tns:partInHeaderRequest"
        part="headerPart" use="literal" wsdl:required="true"
        xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" />

    </wsdl:input>

  </wsdl:operation>
</wsdl:binding>
...
...
```

To enable this behavior, you need to instruct ActiveVOS server to honor WSDL binding declarations by specifying "Policy Driven" as the binding type on the partner link, and then selecting the WSDLBindingReference policy from the available policies in the Policy tab below.

PDD Screen Print:

My Role

Binding:

Service:

Allowed Roles:

Policy:

<wsp:Policy xmlns:abp="http://schemas.active-endpoints.com/ws/2005/12/policy" xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
 <abp:WSDLBindingReference xmlns:ns="http://docs.active-endpoints/activebpel/test/wsdl/wsi-headers/2006/03/wsi-headers.wsdl" name="ns:DC" />
 </wsp:Policy>

Inside the ActiveVOS process (ReceivePartsInHeader as demonstrated in the sample), the incoming message is received into a variable of type Message rather than Element. This allows multiple parts to be received into the process. In the process, the header element is seen as just another part in the message variable – it's the ActiveVOS runtime that figures out, using the WSDL binding definition, that one or more of those parts needs to be mapped to the SOAP header.

To also exemplify "RPC" type of Binding, the top level process i.e. the 'InvokePartsInHeader' is invoked with 'RPC Encoded' type of Binding.

With RPC-style, all the parts are contained within the body element itself.

Example:

```
<soapenv:Envelope ....>
  <soapenv:Header/>
  <soapenv:Body>
    <wsi:runTest
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <one xsi:type="xsd:string">One</one>
      <two xsi:type="xsd:string">Two</two>
    </wsi:runTest>
  </soapenv:Body>
</soapenv:Envelope>
```

Notice that the root is the name of the element, rather than a message.

The subsequent invokes within the top level process is invoked with 'Policy Driven' binding, where the header is passed within the wsdl header element and body within the header body element.

Steps to deploy and execute the project

1. Import the attached project.
2. Deploy MultiParts.bpr
3. Instantiate the top level process – InvokePartsInHeader, available at the endpoint: <http://localhost:8080/active-bpel/services/InvokePartsInHeaderService> with the following request message:

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsdl="http://docs.active-endpoints/activebpel/test/wsdl/wsi-headers/2006/03/wsi-headers.wsdl">
  <soapenv:Header/>
  <soapenv:Body>
    <wsdl:runTest
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <one xsi:type="xsd:string">RPC</one>
      <two xsi:type="xsd:string">DOC</two>
    </wsdl:runTest>
  </soapenv:Body>
</soapenv:Envelope>
```

4. Note that it invokes the ReceivePartsInHeader.bpel by passing on the header into the soap:header and body into soap:body and receives a response from them.

Raw Text as seen from Charles Soap monitoring tool:

```
POST /active-bpel/services/PartsInHeaderDocService HTTP/1.1
Content-Type: text/xml; charset="utf-8"
SOAPAction: "http://docs.active-endpoints/activebpel/test/wsdl/wsi-headers/headerInvoke"
Accept: text/xml, text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
User-Agent: Jakarta Commons-HttpClient/3.1
Content-Length: 544
Host: localhost:8080
```

```
<?xml version='1.0' encoding='UTF-8'?><SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"><SOAP-
ENV:Header><aetgt:ElementForHeaderRequest
xmlns:aetgt="http://schemas.active-endpoints.com/test/wsi-headers/2006/03/wsi-headers.wsdl" xmlns:ns="http://schemas.active-endpoints.com/test/wsi-headers/2006/03/wsi-headers.wsdl">
  <childOne>headerOne</childOne>
  <childTwo>headerTwo</childTwo>
</aetgt:ElementForHeaderRequest>
</SOAP-ENV:Header>
<SOAP-ENV:Body><bodyPart>bodyPart</bodyPart>
</SOAP-ENV:Body></SOAP-ENV:Envelope>
```

Troubleshooting

1. The message gets passed into the body instead of header.

Possible reasons:

- a. Verify if you get any warning message like the following when you invoke

```
org.activebpel.rt.AeException logWarning  
WARNING: Elements for the header must be from the same message  
as the operation being invoked. Message from  
operation=Response Message in binding={2}
```

- b. If so, this might indicate that the message is not mapped correctly in the wsdl correctly and hence the binding can't be resolved. The header parts all have to actually belong to the input message definition or it won't work right.

Check if the soap header "message" attribute contain the value prefixed with namespace i.e they must be:

```
soap:header message="tns:Request" part="SomeHeader" use="literal"/>
```

and not

```
<soap:header message="Request" part="SomeHeader" use="literal"/>
```

- c. Make sure all of the namespace declarations in the binding WSDL match those that are in the WSDL that defines the partner link (i.e. the .public.wsdl that is located within wsdl > bpel directory). Importantly make sure that the file has the target namespace declaration and its prefix.

Reference

[1] About MyRole Binding:

<http://infocenter.activevos.com/infocenter/ActiveVOS/v91/index.jsp?to pic=/com.activee.bpel.doc/html/UG20-5-6.html>

[2] About WSDL Binding reference Policy:

<http://infocenter.activevos.com/infocenter/ActiveVOS/v91/index.jsp?to pic=/com.activee.bpel.doc/html/UG20-5-26.html>



About Active Endpoints

Active Endpoints, Inc. delivers a unique suite of process automation products and related services to develop, integrate and deploy custom applications quickly and easily. The company's products — [Socrates](#) and [ActiveVOS](#) empower business users and IT project teams to collaborate more effectively.

The company has hundreds of customers worldwide from large enterprises to SMBs, spanning multiple industries such as telecommunications, government, financial services, and media and entertainment.

Active Endpoints is headquartered in Waltham, MA with development facilities in Shelton, CT.

To find out how Active Endpoints can help your business, visit <http://www.activevos.com>, call +1 781 547 2900 and press 1 for Sales, or email us at info@activevos.com.