

Base Language - Feature #2232

finish testing the wrapped mode using the web services exposed by the testing server

01/23/2014 04:45 PM - Constantin Asofiei

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Marius Gligor	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	Cleanup and Stablization for Server Features	vendor_id:	GCD
billable:	No		
Description			
Related issues:			
Related to Base Language - Feature #2208: create a java application running w...		Closed	
Related to Base Language - Feature #1645: implement web services support		Closed	02/20/2013 08/09/2013

History

#1 - 01/23/2014 04:51 PM - Constantin Asofiei

Using the server created in [#2208](#), test the web services using wrapped mode and fix any found problems.

Also, fix this from [#1645](#):

Evgeny Kiselev wrote:

Code Review 0116a:

1) com.goldencode.p2j.xml.XNodeRefImpl#getOwnerDocument()

Does RESTRICTED_DOCUMENT can affair to other methods in XNode ? if yes, we need merge this code with isInitialized() method.

Access is restricted only to OWNER-DOCUMENT, for X-Nodes attached to a SOAP-Header or SOAP-Fault-Detail resource; the X-Nodes are valid, is just that in 4GL access to the SOAP Envelope via DOM is not allowed. But I think I've missed a case: when cloning the DOM Node in SOAPHeaderEntryImpl.setNode, I think RESTRICTED_DOCUMENT info should have been set at the clone's document. I will double check how it behaves and fix this.

#2 - 01/23/2014 05:30 PM - Greg Shah

- Target version set to Milestone 11

#3 - 06/02/2014 10:18 AM - Greg Shah

- Assignee set to Marius Gligor

#4 - 06/06/2014 01:29 PM - Constantin Asofiei

In wrapped mode, the request and the response are both treated as a string with the serialized XML contents of the SOAP Envelope body. When

invoking the web service, there will be at most 2 parameters of type char, one INPUT (for the REQUEST) and one OUTPUT (for the RESPONSE). Some testcase examples are in testcases/uast/web_services/, which use some web services freely available on the internet. The key statement is CREATE SERVER to create the server and CONNECT method to connect it.

The tests need to cover at least (in order of importance):

1. does 4GL prepare the RESPONSE on its own, when serializing the SOAP body to string? I.e. does the RESPONSE on 4GL side differ from the RESPONSE actually sent by the remote web service?
2. do we treat the namespaces right? As I recall, 4GL duplicates the namespace definitions in the serialized RESPONSE, because 4GL doesn't provide access to the full SOAP envelope - we need to make sure we are duplicating this properly.
3. how does the full SOAP ENVELOPE (for REQUEST) sent by 4GL looks like on the web service side? Do we match it?
4. some more tests of the SOAP Faults/Headers.
5. if you can think of anything else, please let me know.

#5 - 06/20/2014 02:59 AM - Marius Gligor

- Status changed from New to WIP

#6 - 06/23/2014 10:39 AM - Marius Gligor

1. I used the following existing test case (wsdl-test1.p) to check the RESPONSE on 4GL when call a SOAP WS in wrapped mode:

```
def var h as handle.
def var l as log.
def var hp as handle.
def var chi as char.
def var cho as char.

create server h.

l = h:connect("-WSDL http://www.webserviceX.net/CovertPressure.asmx?WSDL -Service PressureUnit").

if l <> yes then do:
    message "could not connect".
    pause.
    quit.
end.

run PressureUnitSoap set hp on server h.

if not valid-handle(hp) then do:
    message "could not run port".
    pause.
    quit.
end.

chi = "<tns:ChangePressureUnit xmlns:tns='http://www.webserviceX.NET/'><tns:PressureValue>100.0</tns:Pressure
Value>
    <tns:fromPressureUnit>bar</tns:fromPressureUnit><tns:toPressureUnit>atmosphere</tns:toPressureUnit></tns:Ch
angePressureUnit>".

run ChangePressureUnit in hp(input chi, output cho).

if cho <> '<ChangePressureUnitResponse xmlns="http://www.webserviceX.NET/" xmlns:xsi="http://www.w3.org/2001/X
MLSchema-instance" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XM
LSchema"><ChangePressureUnitResult>98.692326671601279</ChangePressureUnitResult></ChangePressureUnitResponse>'
then message "response is incorrect".

message cho.
```

Executing this test case in P4G environment the RESPONSE looks like:

```
<ChangePressureUnitResponse xmlns="http://www.webserviceX.NET/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ChangePressureUnitResult>98.692326671601279</ChangePressureUnitResult>
</ChangePressureUnitResponse>
```

2. Then I used a free web SOAP client (<http://www.soapclient.com/>) to invoke the same service from a web browser and to see the REQUEST and RESPONSE objects.

Invoking the same service from <http://www.soapclient.com/soaptest.html> the RESPONSE looks like:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ChangePressureUnitResponse xmlns="http://www.webserviceX.NET/">
      <ChangePressureUnitResult>98.692326671601279</ChangePressureUnitResult>
    </ChangePressureUnitResponse>
  </soap:Body>
</soap:Envelope>
```

Comparing the RESPONSE from 4GL with this one we can see that in 4GL the RESPONSE is combined from ChangePressureUnitResponse and soap:Envelope elements.

The namespaces attributes from soap:Envelope element are duplicated into ChangePressureUnitResponse element.

3. Regarding the REQUEST in 4GL this is basically the same as the SOAP-ENV:Body content.

The request looks like:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://www.webserviceX.NET/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:w3c="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <SOAP-ENV:Body>
    <tns:ChangePressureUnit xmlns:tns="http://www.webserviceX.NET/">
      <tns:PressureValue>100.0</tns:PressureValue>
      <tns:fromPressureUnit>bar</tns:fromPressureUnit>
      <tns:toPressureUnit>atmosphere</tns:toPressureUnit></tns:ChangePressureUnit>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4. I've converted the test case in P2J. When I executed the converted test in P2J I obtained the following RESPONSE:

```
<ChangePressureUnitResponse xmlns="http://www.webserviceX.NET/">
  <ChangePressureUnitResult>98.692326671601279</ChangePressureUnitResult>
</ChangePressureUnitResponse>
```

The RESPONSE in this case is basically the content of the soap:Body element which is correct but is not the same like on 4GL and the message "response is incorrect" is displayed.

In order to have the same behaviour in P2J like on 4GL we have to duplicate the namespaces from soap:Envelope element to ChangePressureUnitResponse element.

#7 - 06/23/2014 10:42 AM - Greg Shah

As part of this task, please do prepare code changes that resolve any found issues.

#8 - 06/23/2014 10:44 AM - Greg Shah

Have you reviewed [#2208](#) already? If so, can we close it or does that code need some changes?

#9 - 06/23/2014 11:09 AM - Marius Gligor

No I didn't do any code review for [#2208](#) so far.

#10 - 06/24/2014 11:34 AM - Marius Gligor

Today I started to do a code review on issue [#2208](#)

1. I did a checkout of testcases project and I have the testcases/web_services folder which contains a lot of test cases and the axis2 artefacts used to create and deploy SOAP web services

The folder testcases/web_services/server contains axis2 1.6.2 artefacts and some example for test SOAP web services: SimpleService SimpleOperations and MediCareSupplier

Also a StartServer.java class is provided if we need to run the axis2 server within a Java application.

The server can be started also from a command line prompt using the axis2server.sh (axis2server.bat) batch command located in AXIS2_HOME/bin directory.

2. Each SOAP web service example has a build.xml script file which allow to create and deploy services. Web services are deployed into AXIS2_HOME/repository/services folder.

3. Using a standalone client I tested SimpleService and SimpleOperations services and I found to work properly.

4. On the next step I created a P4GL script like:

```
def var h as handle.
def var l as log.
def var hp as handle.
def var chi as char.
def var cho as char.

create server h.

l = h:connect("-WSDL http://localhost:8080/axis2/services/SimpleOperations?wsdl -Service SimpleOperations").

if l <> yes then do: message "could not connect". pause. quit. end.

run SimpleOperationsPortType set hp on server h.

if not valid-handle(hp) then do: message "could not run port". pause. quit. end.

chi = "<ns1:getData xmlns:ns1='http://goldencode.com/xsd'><ns1:str>test</ns1:str></ns1:getData>".

run getData in hp(input chi, output cho).

message cho.
```

I converted the script into P2J Java code and I tried to execute the test but I've got the following error.

```
Web service operation getData generated a SOAP Fault. SOAP faultstring is: No port found for the given name :SimpleOperationsPortType (11506)
```

The same error is displayed for SimpleService service call. Basically no service calls are made. The P2J code cannot determine and bind the service port when parsing WSDL.

5. Then I changed the axis2 server port from 8080 to 80. On my network router I created a tunnel which allow to access the axis2 server from outside (Internet)

I created the same script in the customer 4GL environment but I changed the connection line as follow:

```
l = h:connect("-WSDL http://5.15.92.196/axis2/services/SimpleOperations?wsdl -Service SimpleOperations").
```

The 5.15.92.196 is my current public WAN IP address. The port was changed to standard HTTP (80) because the 8080 port seems to be blocked by the customer's firewall.

Running the script I found that's work properly on P4GL.

6. In conclusion the test web services are working on 4GL but not on P2J perhaps due to a bug in the code that must be fixed.

7. In the mean time I downloaded and deployed the axis2-1.6.2-war.zip inside an Apache Tomcat 7 web server. Also I did some changes in configuration

in order to enable the SOAPMonitor an axis2 tool which allow to monitor and see the SOAP request and response messages.

Next I will try to switch from port 8080 to port 80 in order to monitor SOAP messages from remote P4GL service calls.

8. From my point of view the code review on issue is done and you may close the issue [#2208](#).

#11 - 06/25/2014 11:16 AM - Marius Gligor

- File `mag_upd20140625a.zip` added

Debugging the SimpleOperations converted web service test case in P2J an AxisFault is throw in WebServiceHelper line 897

```
sc = new ServiceClient(null, wsdlDef, new QName(ns, srvName), portTypeName);
```

Looking into the source code the ServiceClient constructor looks like:

```
/**
 * This is WSDL4J based constructor to configure the Service Client/ TODO: make this policy
 * aware
 *
 * @param configContext active ConfigurationContext
 * @param wsdl4jDefinition the WSDL we're going to be using to configure ourselves
 * @param wsdlServiceName QName of the WSDL service we'd like to access
 * @param portName name of the WSDL port we'd like to access
 * @throws AxisFault in case of error
 */
public ServiceClient(ConfigurationContext configContext, Definition wsdl4jDefinition,
                    QName wsdlServiceName, String portName) throws AxisFault {
    configureServiceClient(configContext, AxisService.createClientSideAxisService(
        wsdl4jDefinition, wsdlServiceName, portName, options));
}
```

which means that the portName parameter is the name of the port from WSDL that we have to use (bind).

Further debugging inside WSDL11ToAxisServiceBuilder on findBinding() method the portName parameter is used as a key in a Map of available ports.

Here is a snippet of code from WSDL11ToAxisServiceBuilder class containing the code for findBinding() method.

```
public class WSDL11ToAxisServiceBuilder extends WSDLToAxisServiceBuilder {
    /**
     * Look for the relevant binding!
     * if user has spcified a port get it
     * otherwise find first soap port or pick random one if there is no soap port
     *
     * @param dif
     * @param service service can not be null
     * @throws AxisFault
     */
    private Binding findBinding(Definition dif, Service service) throws AxisFault {
```

```

Binding binding = null;
Port port = null;
copyExtensibleElements(service.getExtensibilityElements(), dif, axisService, SERVICE);
if (portName != null) {
    // i.e if user has specified a service
    port = service.getPort(portName);
    if (port == null) {
        throw new AxisFault("No port found for the given name : " + portName);
    }
} else {
    Map ports = service.getPorts();
    if (ports != null && ports.size() > 0) {
        // pick the port with the SOAP address as the default port
        port = findPort(ports);
        if (port == null) {
            // a SOAP port was not found - log a warning
            // and use the first port in the list
            log.info("A SOAP port was not found - "
                + "picking a random port!");
            port = (Port) ports.values().toArray()[0];
        }

        if (port != null) {
            // i.e we have find a correct port
            // this is only use full in codegen time.
            if (this.isCodegen && !this.isAllPorts) {
                // if user has not set all option
                // we have to generate code only for that option.
                this.portName = port.getName();
            }
        }
    }
}
}
}
}

```

```
axisService.setName(service.getQName().getLocalPart());
```

```

if (port != null) {
    copyExtensibleElements(port.getExtensibilityElements(), dif,
        axisService, PORT);
    Definition parentDefinition = getParentDefinition(dif,
        port.getBinding().getQName(), COMPONENT_BINDING, new HashSet());
    binding = parentDefinition.getBinding(port.getBinding().getQName());
    if (binding == null) {
        binding = port.getBinding();
    }
}
}

```

```
return binding;
```

The available ports from WSDL are stored on a Map in Servicelmpl class which is an implementation of Service interface.

```

protected Map ports = new HashMap();

/**
 * Get the specified port.
 *
 * @param name the name of the desired port.
 * @return the corresponding port, or null if there wasn't
 * any matching port
 */
public Port getPort(String name)
{
    return (Port)ports.get(name);
}

```

If the portName is not a key in the Map set of keys an AxisFault exception is throw.

If no portName is specified a port with the SOAP address is selected automatically from the existing ports in the Map.

```
if (portName != null) {
    // i.e if user has specified a service
    port = service.getPort(portName);
    if (port == null) {
        throw new AxisFault("No port found for the given name : " + portName);
    }
} else {
    Map ports = service.getPorts();
    if (ports != null && ports.size() > 0) {
        // pick the port with the SOAP address as the default port
        port = findPort(ports);
    }
}
```

In conclusion if we specify a portName this port should be a key in the in the Map set of keys.

In WebServiceHelper when the ServiceClient constructor is invoked the portTypeName is the name of the port type and not the name of a port.

```
sc = new ServiceClient(null, wsdlDef, new QName(ns, srvName), portTypeName);
```

In order to avoid an AxisFault throw the portTypeName must be equals with one of the available port names. Unfortunately this is not always true.

1 For <http://www.webserviceX.net/CoverPressure.asmx?wsdl> ChangePressureUnit service the portType name and binding name are equals and no AxisFault exception is throw.

```
<wsdl:portType name="PressureUnitSoap">
  <wsdl:operation name="ChangePressureUnit">
    <wsdl:input message="tns:ChangePressureUnitSoapIn"/>
    <wsdl:output message="tns:ChangePressureUnitSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
.....
<wsdl:binding name="PressureUnitSoap" type="tns:PressureUnitSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="ChangePressureUnit">
    <soap:operation soapAction="http://www.webserviceX.NET/ChangePressureUnit" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
```

2. For <http://localhost:8080/axis2/services/SimpleOperations?wsdl> SimpleOperations service the port type name is SimpleOperationsPortType and the name of the ports are SimpleOperationsSoap11Binding, SimpleOperationsSoap12Binding and SimpleOperationsHttpBinding so an AxisFault exception is throw with the message

```
No port found for the given name : SimpleOperationsPortType
```

```
<wsdl:portType name="SimpleOperationsPortType">
  <wsdl:operation name="getData">
    <wsdl:input message="tns:getDataRequest" wsaw:Action="urn:getData"/>
    <wsdl:output message="tns:getDataResponse" wsaw:Action="urn:getDataResponse"/>
  </wsdl:operation>
.....
<wsdl:binding name="SimpleOperationsSoap11Binding" type="tns:SimpleOperationsPortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
  <wsdl:operation name="getData">
    <soap:operation soapAction="urn:getData" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
```

```
</wsdl:input>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
```

3. I've fixed this bug by providing a `getPortName` method which search the WSDL binding list for a port name equals with the parameter port type name.

If a port having the same name like the port type name is found the port type name is returned otherwise a null value is returned and the port will be automatically selected from the list of available ports. The `ServiceClient` constructor call became:

```
sc = new ServiceClient(null, wsdlDef, new QName(ns, srvName), getPortName(portTypeName))
```

Having these fixes all tested web services are working in P2J.

#12 - 06/25/2014 03:44 PM - Constantin Asofiei

Marius Gligor wrote:

Having these fixes all tested web services are working in P2J.

The update looks OK. For this change, no regression testing is required.

#13 - 06/26/2014 01:57 PM - Marius Gligor

- File `mag_upd20140626a.zip` added

- % Done changed from 0 to 80

Bellow are the request and response SOAP messages on P4GL and P2J.

1. Industry standards supported by OpenEdge according to web services manual:

- WSDL 1.1 is the de facto standard according to the W3C.
- SOAP 1.1 HTTP Binding is the de facto standard according to the W3C.
- XML Schema — W3C 2001 XML Schema Recommendation.

By default axis2 use SOAP 1.2 for binding. For compatibility when port name is automatically choose from the list of available ports by axis2 client we force also the use of SOAP 1.1 binding as default.

Looking to request SOAP messages on both P4GL and P2J SOAP 1.1 is used for binding.

The content of the Body element is the same on both cases and is provided when we call de service as input parameter.


```
<ns0:getData xmlns:ns0="http://goldencode.com/xsd"><ns0:str>test</ns0:str></ns0:getData>
```

Basically the SOAP request messages are equivalent regarding the prefix of namespaces.

2. The response is obviously identical in both P4GL and P2J because is generated by the same server.

3. The messages displayed on test case are also the same because I implemented the code to merge SOAP Envelope namespaces with namespaces in response. I tested also the wsdl_test1.p test case on <http://www.webservicex.net/CovertPressure.asmx> service and now the generated message match the expected message.

4. I tested also some fault messages and I didn't find differences. However in P2J in case of a WS fault message the error message is displayed twice so I provided a quick fix.

P4GL Request:

```
=====  
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
  <SOAP-ENV:Body xmlns:s0="http://goldencode.com/xsd" xmlns:s1="http://www.w3.org/2001/XMLSchema">  
    <ns0:getData xmlns:ns0="http://goldencode.com/xsd">  
      <ns0:str>test</ns0:str>  
    </ns0:getData>  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

P2J Request:

```
=====  
<?xml version='1.0' encoding='UTF-8'?>  
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
  <soapenv:Body>  
    <ns0:getData xmlns:ns0="http://goldencode.com/xsd">  
      <ns0:str>test</ns0:str>  
    </ns0:getData>  
  </soapenv:Body>  
</soapenv:Envelope>
```

Response:

```
=====  
<?xml version='1.0' encoding='UTF-8'?>  
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
  <soapenv:Body>  
    <ns:getDataResponse xmlns:ns="http://goldencode.com/xsd">  
      <ns:return>1.7976931348623157E308</ns:return>  
    </ns:getDataResponse>  
  </soapenv:Body>  
</soapenv:Envelope>
```

P4GL client display:

```
=====  
<ns:getDataResponse xmlns:ns="http://goldencode.com/xsd" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
  <ns:return>1.7976931348623157E308</ns:return>  
</ns:getDataResponse>
```

P2J client display:

```
=====  
<ns:getDataResponse xmlns:ns="http://goldencode.com/xsd" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
  <ns:return>1.7976931348623157E308</ns:return>  
</ns:getDataResponse>
```

#14 - 06/27/2014 06:56 AM - Constantin Asofiei

Review for 0626a.zip:

- when adding multi-line comments, use single-line commenting, not block commenting
- `WebServiceHelper.mergeNameSpaces` - you are looking for the `>` char to determine where the first node is opened. Are you sure `>` can appear only at the end of the first opened node? More, if the response has no `>` char in it, then the response will become the empty string.
- about the change in `ControlFlowOps` related to this:

4. I tested also some fault messages and I didn't find differences. However in P2J in case of a WS fault message the error message is displayed twice so I provided a quick fix.

I don't think it is correct what you are doing. When you call `ErrorHandler.setSilent(true)`, you are enabling silent error mode, as if `NO-ERROR` clause is in effect. What test are you using that results in two soap fault messages? I wonder if it is not related to the problems in `ErrorHandler` Vadim G. is currently working on.

#15 - 06/27/2014 08:07 AM - Marius Gligor

- File `mag_upd20140627a.zip` added

- File `fault1.png` added

- File `fault2.png` added

1. I fixed the issues from `WebServiceHelper`.

2. For `FAULT` test I used for example the `wsdl_test1.p` and I did a change on `bar` parameter to `xbar` in the input string like:

```
chi = "<tns:ChangePressureUnit xmlns:tns='http://www.webserviceX.NET/'><tns:PressureValue>100.0</tns:PressureValue><tns:fromPressureUnit>xbar</tns:fromPressureUnit><tns:toPressureUnit>atmosphere</tns:toPressureUnit></tns:ChangePressureUnit>".
```

The fault message generated by the server is displayed twice.

Indeed seems to be a more general bug because I tried the same test case but doing a change like:

```
run ChangePressureUnitX in hp(input chi, output cho).
```

The error message is displayed twice and the message is generated by P2J. (see the attached pictures).

First the `fault1` message is displayed then after pressing a key `fault2` message is displayed which basically is the same like the first message.

So I reverted my changes in `ControlFlowOps`.

#16 - 06/27/2014 11:23 AM - Marius Gligor

- Status changed from WIP to Review
- File mag_upd20140627b.zip added
- File soap_hdr_1.p added
- % Done changed from 80 to 90

I created a simple JAX-WS using Netbeans IDE deployed on GlassFish AS which allow easily to insert a SOAP header.

```
@WebService(serviceName = "MyService")
public class MyService
{
    @WebMethod(operationName = "hello")
    public String hello(@WebParam(name = "name") String txt,
        @WebParam(header = true, mode = Mode.OUT) Holder<String> headerParam)
    {
        headerParam.value = "Test header";
        return "Hello " + txt + " !";
    }
}
```

The SOAP response message containing the SOAP Header looks like:

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <ns2:arg1 xmlns:ns2="http://soap.goldencode.com/">Test header</ns2:arg1>
  </S:Header>
  <S:Body>
    <ns2:helloResponse xmlns:ns2="http://soap.goldencode.com/">
      <return>Hello Marius !</return>
    </ns2:helloResponse>
  </S:Body>
</S:Envelope>
```

Then I created a test case script to call this service and I defined 2 callback procedures to handle the request and response SOAP headers. Running the converted test in P2J I'm able to get and display the message "Test header" from response SOAP Header. On request the following SOAP message containing a SOAP Header is send. The generated SOAP Header is OK.

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <header xmlns="http://schemas.xmlsoap.org/soap/envelope/">
      <AuthHeader xmlns="http://ServiceHost/SOAPHeader">
        <UserName>usr</UserName>
        <Password>psw</Password>
      </AuthHeader>
    </header>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:hello xmlns:ns1="http://soap.goldencode.com/">
      <name>test</name>
    </ns1:hello>
  </soapenv:Body>
</soapenv:Envelope>
```

In conclusion the implementation of SOAP Header is OK. However I found a conversion issue when a statement like

```
hXdoc:IMPORT-NODE(hXnoderef2, hXnoderef1, v-log).
```

is converted.

The v-log is a parameter of type logical. When we use YES or TRUE instead of a variable the conversion is made to Java boolean and we get a compile error.

I fixed this issue by adding a new method having a Java boolean value as parameter.

#17 - 06/27/2014 11:34 AM - Greg Shah

I will let Constantin do the code review.

My only question: it seems like our output has case differences from the 4GL output (e.g. soapenv vs SOAPENV). Is that right? If so, it seems like we need to match the case of the output so that any code that is somehow dependent upon case won't be broken. For example, if someone wrote their own code to parse the output, they might have unintended case-sensitive matching that would break in our approach.

#18 - 06/27/2014 11:55 AM - Marius Gligor

- File deleted (soap_hdr_1.p)

#19 - 06/27/2014 12:03 PM - Marius Gligor

- File soap_hdr_1.p added

1. I've just tested the script on P4GL and I changed the test script. In 4GL the order of procedure parameters is mandatory else an error is thrown. The generated request looks like:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header>
    <ns0:AuthHeader xmlns:ns0="http://ServiceHost/SOAPHeader">
      <ns0:UserName>usr</ns0:UserName>
      <ns1:Password xmlns:ns1="http://ServiceHost/SOAPHeader">psw</ns1:Password>
    </ns0:AuthHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body xmlns:s0="http://soap.goldencode.com/" xmlns:s1="http://www.w3.org/2001/XMLSchema">
    <ns1:hello xmlns:ns1="http://soap.goldencode.com/">
      <name>test</name>
    </ns1:hello>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2. The call back for response header does not work on P4GL. I have to fix it inside the test script.

3. The case of prefix must be the same as described in namespace. For example xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" define prefix as SOAP-ENV in upper case and is used exactly as is described SOAP-ENV:Body. It is not mandatory to have the same prefix like on P4GL

#20 - 06/27/2014 02:26 PM - Constantin Asofiei

Marius Gligor wrote:

3. The case of prefix must be the same as described in namespace. For example
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
define prefix as SOAP-ENV in upper case and is used exactly as is described SOAP-ENV:Body. It is not mandatory to have the same prefix like
on P4GL

You are referring to the request packaged by 4GL and sent to the server, right? I don't think we need to mimic the exact case in this case; but what we need to mimic exactly is the response: the response produced by P2J must use the same case as 4GL does. The same applies for response headers, faults, etc, which are received/interpreted by 4GL.

The 0627b.zip update looks OK.

About your ControlFlowOps problem: I think this IF clause if (e.getCause() instanceof NumberedException) in this code in ControlFlowOps.invokeImpl:3711:

```
catch (ConditionException e)
{
    if (e.getCause() instanceof NumberedException)
    {
        ErrorManager.recordOrThrowError((NumberedException) e.getCause());
    }
    else
    {
        throw e;
    }
}
```

must not be executed for web service invocation; it is related to treating a condition received from a remote appserver call. Thus, change the IF to this if (!isWebService && e.getCause() instanceof NumberedException), so the condition can be throw for the web service case.

#21 - 06/30/2014 08:58 AM - Marius Gligor

- File deleted (soap_hdr_1.p)

#22 - 06/30/2014 08:59 AM - Marius Gligor

- File soap_hdr_1.p added

- File Axis2-OutSoapHeader.zip added

- File mag_upd20140630a.zip added

The previous JAX-WS service used to test the SOAP header in response does not send the header OOB (Out Of Band) P4G when parse the WSDL for this service interpret the header as a service method parameter and an error is throw indicating that the actual number of parameters does not match the number of expected parameters.

Then I created an axis2 module which inject the SOAP Header in the response in an OOB manner. Testing the script with SimpleOperations service and the injected SOAP Header it works on both P4GL and P2J and produces the same results. Below is the description of the axis2 module project. In conclusion P2J SOAP Header implementation is OK.

This project is an axis2 module designed to inject a simple SOAP Header on response.

1. Build the project using the ant script. On success a module archive headers.mar is created.
2. Copy the module headers.mar into AXIS2_HOME/repository/modules directory.
3. Change the AXIS2_HOME/conf/axis2.xml as follow:
 - a). Register module by add the following line:

```
<module ref="sampleModule"/>
```

to global modules. After add it should looks like:

```
<!-- ===== -->
<!-- Global Modules -->
<!-- ===== -->
<!-- Comment this to disable Addressing -->
<module ref="addressing"/>
<module ref="sampleModule"/>
```

- b). Add the following line

```
<phase name="SamplePhase"/>
```

to Phases section to looks like:

```
<phaseOrder type="OutFlow">
  <!-- user can add his own phases to this area -->
  <phase name="soapmonitorPhase"/>
  <phase name="OperationOutPhase"/>
  <phase name="SamplePhase"/>
  <!--system predefined phase-->
  <!--these phase will run irrespective of the service-->
```

4. Start the axis2 server an check a web service like SimpleOperations.getData. The response should looks like:

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <ns1:SampleOutHeader xmlns:ns1="http://sample.module">Test Header</ns1:SampleOutHeader>
  </soapenv:Header>
  <soapenv:Body>
    <ns:getDataResponse xmlns:ns="http://goldencode.com/xsd">
      <ns:return>1.7976931348623157E308</ns:return>
    </ns:getDataResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

The response contains the injected header:

```
<ns1:SampleOutHeader xmlns:ns1="http://sample.module">Test Header</ns1:SampleOutHeader>
```

#23 - 06/30/2014 09:54 AM - Constantin Asofiei

Marius, the update looks good. Please commit the tests and the Axis2-OutSoapHeader.zip to bsr.

#24 - 06/30/2014 10:13 AM - Marius Gligor

Tests and the Axis2-OutSoapHeader project has been committed in testcases revision 1161

#25 - 07/02/2014 04:34 AM - Constantin Asofiei

You can release the 0630a.zip update, the changes don't affect MAJIC runtime.

#26 - 07/02/2014 04:50 AM - Marius Gligor

- % Done changed from 90 to 100

No regression tests are mandatory for this changes.
Committed revision 10557.

#27 - 07/02/2014 05:49 AM - Greg Shah

- Status changed from Review to Closed

#28 - 11/16/2016 12:07 PM - Greg Shah

- Target version changed from Milestone 11 to Cleanup and Stabilization for Server Features

Files

mag_upd20140625a.zip	15.7 KB	06/25/2014	Marius Gligor
mag_upd20140626a.zip	41.6 KB	06/26/2014	Marius Gligor
fault1.png	14.3 KB	06/27/2014	Marius Gligor
fault2.png	14.3 KB	06/27/2014	Marius Gligor
mag_upd20140627a.zip	16.6 KB	06/27/2014	Marius Gligor
mag_upd20140627b.zip	25.5 KB	06/27/2014	Marius Gligor
mag_upd20140630a.zip	50.5 KB	06/30/2014	Marius Gligor
soap_hdr_1.p	4.06 KB	06/30/2014	Marius Gligor
Axis2-OutSoapHeader.zip	5.62 KB	06/30/2014	Marius Gligor