

Base Language - Feature #3310

add support for the equivalent of the OpenEdge Web Services Adapter (and ProxyGen) and PASOE SOAP transport

07/05/2017 08:11 AM - Greg Shah

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Constantin Asofiei	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		vendor_id:	GCD
billable:	No		
Description			
Related issues:			
Related to Base Language - Feature #3855: implement equivalent support for RE...			New
Related to Database - Bug #5801: write-xml is not using XML-NODE-NAME for the...			Feedback
Related to Database - Bug #5799: write-xml doesn't serialize the prods:id att...			Feedback

History

#1 - 07/05/2017 08:30 AM - Greg Shah

In addition to providing a web-services client, the 4GL can also expose 4GL code as web-services. They only support SOAP over HTTP or HTTPS.

The way it works is that the 4GL code is exposed as a set of APIs using the normal appserver (unified broker/agent architecture). Then ProxyGen is used to create a "web services mapping" (.wsm) file which is used to generate the WSDL definition (used as an API definition by clients to program the SOAP RPC) and also used to deploy the code as accessible through the Web Services Adapter (WSA). The WSA is a runtime component that listens for the SOAP requests of HTTP/HTTPS and translates those into appserver calls to the matching 4GL code. The responses returned by the 4GL code are packaged up and returned as the SOAP response.

https://documentation.progress.com/output/ua/OpenEdge_latest/index.html#page/gsaiss%2Fweb-services-3a-architecture-and-tools.html%23

This task is meant to create a functional replacement for the WSA. We also probably need to implement the ProxyGen features related to this. While we are implementing ProxyGen, we should go ahead and implement the remote object interfaces related to the Java Open Client (the traditional OE Appserver access from Java code). Our current appserver support has a hand-coded client side that uses a generic appserver access API instead of using a remote object interface that represents each application-specific API.

#2 - 07/05/2017 08:32 AM - Greg Shah

Some thoughts from Constantin:

At this time, it looks like we need to build a web service server, which can be run either embedded in FWD or as a standalone server. There are some details about how to publish a web service using proxygen, here:

<http://ultirags.com/ultiimages/How-to-publish-an-OpenEdge-Web-Service%E2%80%93an-overview-v1.0.pdf>

From the looks of it, we will need to configure/implement:

1. the web service server
2. the details about the appserver connection - this is related to the settings configured in proxygen when generating the WSA files
3. details about the exported services and how they map to the 4GL code - this includes registering the WSDL file(s) in use (which I assume will

not be generated by us, but emitted by proxygen), the proxygen settings for it and how to map the to 4GL code. My assumption is that proxygen uses same 4GL legacy names (but they might pre-process the names for special chars), so we need a way to map them, as we will have better control, and not just assume the service name is the procedure name.

4. security at the web service level - is this needed? In OpenEdge/WSA, I think they rely on the security configurations in web.xml deployment descriptor; we can do something similar (if we rely on Tomcat or other container to deploy this web app) or just rely on the FWD directory.

The tricky part from all of this I think is the security stuff.

#3 - 07/05/2017 08:40 AM - Greg Shah

which can be run either embedded in FWD or as a standalone server.

To simplify deployment I prefer it embedded in FWD. This is also consistent with the fact that our appserver "broker" is embedded in the FWD server.

this includes registering the WSDL file(s) in use (which I assume will not be generated by us, but emitted by proxygen)

I think we do need to implement our own version of ProxyGen, which would include handling the WSDL part. I think we can implement all of this as part of the conversion process. Ultimately, ProxyGen is just outputting the API for a set of 4GL code in various forms. It seems straightforward to generate the WSDL, the API definitions suitable for web services "exporting" and even the RemoteObject implementation, during conversion. All we need is a list of all the exported APIs and any other configuration hints that one might normally pass the the 4GL ProxyGen utility. In other words, our ProxyGen is just an additional set of artifacts from the conversion process.

I think they rely on the security configurations in web.xml deployment descriptor; we can do something similar (if we rely on Tomcat or other container to deploy this web app) or just rely on the FWD directory.

I don't want to use Tomcat for this. It adds overhead, extra configuration requirements and complexity to the deployment.

I agree, we would put such configuration in the FWD directory.

#4 - 07/05/2017 08:48 AM - Constantin Asofiei

Greg Shah wrote:

which can be run either embedded in FWD or as a standalone server.

To simplify deployment I prefer it embedded in FWD. This is also consistent with the fact that our appserver "broker" is embedded in the FWD server.

A note about standalone mode: I mean to decouple the FWD application server and the web service server. One might want to expose the web services to the world and this server to be ran on a different machine than the FWD application server. In embedded mode, if the web services target appserver calls exposed internally, we will route this as an internal call. In standalone mode, we will establish a remote connection to the FWD server and invoke the APIs remotely.

#5 - 07/05/2017 09:04 AM - Greg Shah

A note about standalone mode

I understand. But that involves making the solution more complex. Initially, I just want to target the embedded in FWD approach. We can always add standalone mode later if needed by customers.

#6 - 01/10/2019 09:51 AM - Greg Shah

- Subject changed from add support for the equivalent of the OpenEdge Web Services Adapter (and ProxyGen) to add support for the equivalent of the OpenEdge Web Services Adapter (and ProxyGen) and PASOE SOAP transport

We need to include the equivalent of SOAP transport for PASOE mode. I believe this is no different from the WSA in classic appserver but if there are any differences, we need to handle them.

#7 - 04/05/2019 06:17 PM - Constantin Asofiei

For Web Services, we need these questions answered, for both PASOE and WSA modes:

- what artifacts are required to run the web services in 4GL (WSDL?). What is the role of each artifact, so we can determine what we need at conversion and/or runtime from it.
- how is the mapping from the web service to the actual 4GL external program being made? I assume the WSDL will have this knowledge? Can

internal procedures/functions be mapped?

- how are parameters passed and mapped to 4GL parameters?
- what is the SOAP structure for this call? How can the header and body look?
- examples for SOAP request and response XML messages
- error handling - how is the SOAP FAULT error generated by the server and what does it contain?

The [#3855-2](#) questions for REST apply here, too. The client executing the SOAP requests needs to be in 4GL, too.

#8 - 04/08/2019 12:39 PM - Constantin Asofiei

Regarding PASOE: we need to determine in which mode the SOAP requests are ran (I assume session-free), when handled by the PASOE Agent. Also, are there SESSION attributes or methods (like REMOTE) which are dependant on executing a SOAP request?

#9 - 04/15/2019 07:36 AM - Greg Shah

From Marian:

I don't know much about your FWD server side (assumed it's some Java but looks like Tomcat is ruled out so what is it then?)... at first sight it's not clear what we need to do for WSA/REST. Do you have an appserver (classic/pasoe) where we need to setup those WSA (SOAP) and REST endpoints?

FWD has its own built-in application server as well as an embedded J2EE container (Jetty) that supports the full range of features.

However, the tests should be designed to run under OpenEdge, using both classic appserver and pasoe. Please include all necessary configuration/scripts for the setup and execution of appserver/pasoe for these tests.

GCD will implement the necessary features in FWD so that these same WSA/REST configuration/artifacts will work in FWD to call the converted versions of the same procedures that would be executed in OpenEdge.

#10 - 04/15/2019 07:36 AM - Greg Shah

- Related to Feature #3855: implement equivalent support for REST (classic appserver, PASOE REST and WEB transports) added

#11 - 04/15/2019 07:40 AM - Marian Edu

Greg Shah wrote:

FWD has its own built-in application server as well as an embedded J2EE container (Jetty) that supports the full range of features.

However, the tests should be designed to run under OpenEdge, using both classic appserver and pasoe. Please include all necessary configuration/scripts for the setup and execution of appserver/pasoe for these tests.

GCD will implement the necessary features in FWD so that these same WSA/REST configuration/artifacts will work in FWD to call the converted versions of the same procedures that would be executed in OpenEdge.

OK, we're currently setting up OE application servers and then build artifacts as well as test scripts in 4GL and postman/soapui. I guess we will have to test all possible cases as usual :)

#12 - 04/15/2019 07:48 AM - Constantin Asofiei

There are the SESSION:CURRENT-REQUEST-INFO and SESSION:CURRENT-RESPONSE-INFO, which return a Progress.Lang.OERequestInfo instance. These are used only in PASOE I think, and we need some tests to know how each property is formatted, and possible values.

There are also the GetClientPrincipal() and SetClientPrincipal() methods, they look like simple getter/setter, and the only way to set a value is via the setter (default is unknown). These are assumptions from the docs, if they are correct, then nothing special is needed for testing.

#13 - 04/15/2019 08:08 AM - Greg Shah

I guess we will have to test all possible cases as usual :)

Yes, please. :)

#14 - 09/16/2019 10:18 AM - Constantin Asofiei

Marian, on xfer you have a appserver/soap folder - please run the tests in SoapUI and upload the entire request/response log.

#15 - 09/23/2019 07:08 PM - Constantin Asofiei

Marian, the .xpxg can have a SubAppObject node (as a child of AppObject) - please create some tests for this. I'm not sure how this can be invoked.

#16 - 09/24/2019 07:14 AM - Constantin Asofiei

Created task branch 3310a from trunk rev 11329.

#17 - 09/24/2019 08:31 AM - Constantin Asofiei

Marian, one more question: what happens if there are multiple wsdl generated via proxygen (i.e. from fwd.xpxg and fwdgc.xpxg), when ?wsdl is queried? This query parameter AFAIK should return the WSDL content, and I don't know what happens if there are multiple WSDL files, all served by the same URL.

Are the services merged into a single one?

#18 - 09/25/2019 07:52 AM - Marian Edu

Constantin Asofiei wrote:

Marian, one more question: what happens if there are multiple wsdl generated via proxygen (i.e. from fwd.xpxg and fwdgc.xpxg), when ?wsdl is queried? This query parameter AFAIK should return the WSDL content, and I don't know what happens if there are multiple WSDL files, all served by the same URL.

Are the services merged into a single one?

For 'classical' application server you need to use a 'web service adapter' and there can be only one wsm deployed so the wsdl will list all services

defined in that one.

For 'pasoe' you can deploy multiple .wsm files but those will be deployed as separate web app services (of SOAP type) with different endpoints so there is no merge, each endpoint ?wsdl will list the content of the deployed wsm file.

#19 - 09/26/2019 04:35 AM - Marian Edu

Constantin Asofiei wrote:

Marian, the .xpxg can have a SubAppObject node (as a child of AppObject) - please create some tests for this. I'm not sure how this can be invoked.

Updated the proxygen definition to add one sub app object, added tests for it in SoapUI suite. Basically a separate namespace is added for the sub app object in WSDL: subFwd. So, when calling endpoints in that sub app object the correct namespace must be used:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:tempuri-org:subFwd">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:pipe_char>
      <urn:inputValue>234</urn:inputValue>
    </urn:pipe_char>
  </soapenv:Body>
</soapenv:Envelope>
```

#20 - 09/26/2019 04:37 AM - Marian Edu

Constantin Asofiei wrote:

Marian, on xfer you have a appserver/soap folder - please run the tests in SoapUI and upload the entire request/response log.

Log file of test suite ran in SoapUI uploaded in appserver/soap/test/web folder.

Sorry for taking so long, have had issues setting up the web service adapter on my environment but now everything is up and running again.

Constantin Asofiei wrote:

For Web Services, we need these questions answered, for both PASOE and WSA modes:

- what artifacts are required to run the web services in 4GL (WSDL?). What is the role of each artifact, so we can determine what we need at conversion and/or runtime from it.

What is deployed on both WSA and PASOE is the .wsm file, that one is generated from the .xpxg by the ProxyGen and basically contains the mapping between each soap service and the 4GL procedure, including data type for parameters. When deployed the WSDL is generated and saved on the WSA/PASOE, although the WSDL is also generated by the ProxyGen - I believe the URL is updated there to match the actual server name where it was deployed (in proxygen you can set a static URL).

- how is the mapping from the web service to the actual 4GL external program being made? I assume the WSDL will have this knowledge?

No, the WSDL is a regular web service definition and doesn't have any mapping information. I presume they just use the .wsm file for that purpose.

Can internal procedures/functions be mapped?

Not directly, one can add the external procedure that contains those internal procedures/functions as persistent and then you can instantiate one of those persistent procedures and later call an internal entry in that one... this is possible from 4GL using set-callback-procedure to catch the persistent procedure instance UUID (sample in appsrv/soap/test/4glws. I'll write some test case in SoapUI for that as well.

- how are parameters passed and mapped to 4GL parameters?

The parameters must be sent according the the WSDL definition, the WSA/PASOE SOAP will handle the data conversion as per mapping defined in .wsm file, if there are any data conversion errors SOAP errors will be thrown - we have tests for most if not all possible combinations in the primitives test suite.

- what is the SOAP structure for this call? How can the header and body look?
- examples for SOAP request and response XML messages
- error handling - how is the SOAP FAULT error generated by the server and what does it contain?

Log file of test suite ran from SoapUI was uploaded in bazaar and you will probably find all the details there.

#23 - 05/13/2020 01:11 PM - Constantin Asofiei

Marian, can you please run the full 'primitives' suite in SoapUI and commit the log? The current log has only a small subset of the tests...

#24 - 05/13/2020 01:13 PM - Marian Edu

Constantin Asofiei wrote:

Marian, can you please run the full 'primitives' suite in SoapUI and commit the log? The current log has only a small subset of the tests...

I will ask Mihai to do that tomorrow morning, I don't have the appsrv setup on my environment :(

#25 - 05/13/2020 02:48 PM - Constantin Asofiei

OK, do you know if you can use table with extent fields (or extent arguments) for SOAP requests?

#26 - 05/14/2020 06:24 PM - Constantin Asofiei

Marian, I need also tests about these:

- raised errors (QUIT, STOP, ERROR, etc)
- some tests for the before-table - is it possible at all?
- return values from procedure - do you have this? If not, please add.
- extent (fields or vars), if is possible.

#27 - 05/15/2020 02:17 AM - Marian Edu

Constantin Asofiei wrote:

OK, do you know if you can use table with extent fields (or extent arguments) for SOAP requests?

It's definitively possible to have extent parameters and extent fields in temp-tables. For parameters those can be both fixed or indeterminate size, temp-tables only support fixed size extents so that's the only option anyway.

#28 - 05/15/2020 02:48 AM - Marian Edu

Constantin Asofiei wrote:

Marian, I need also tests about these:

- raised errors (QUIT, STOP, ERROR, etc)

There are procedures for all those cases, in SoapUI project I can see those but not in any of the log files.

- some tests for the before-table - is it possible at all?

It is, however as before table can only work for tables that are part of a dataset (enable tracking changes) the test is for a dataset, again I don't see anything in the log files although the test is in SoapUI.

- return values from procedure - do you have this? If not, please add.

The test is there, it is possible to use return value.

- extent (fields or vars), if is possible.

Already answered that, possible and tests are being added.

Now, the issue we have is because we have a subscription plan our licenses for 11.7 expired and we now have OE12 installed instead so Mihai is trying to setup the application servers again in order to run the tests, right now there is no environment where we can run those so it might take some time :(

#29 - 05/15/2020 03:02 AM - Constantin Asofiei

Marian Edu wrote:

There are procedures for all those cases, in SoapUI project I can see those but not in any of the log files.

I don't see any of these encoded in appsrv/soap/test/web/fwd-soapui-project.xml.

- return values from procedure - do you have this? If not, please add.

The test is there, it is possible to use return value.

Can you point me to which one uses this, in the fwd-soapui-project.xml?

#30 - 05/15/2020 03:11 AM - Constantin Asofiei

Marian Edu wrote:

Now, the issue we have is because we have a subscription plan our licenses for 11.7 expired and we now have OE12 installed instead so Mihai is trying to setup the application servers again in order to run the tests, right now there is no environment where we can run those so it might take some time :(

After you can run it, please don't forget to upload the modified .xpxg and .wsdl files, too.

#31 - 05/15/2020 03:14 AM - Marian Edu

Constantin Asofiei wrote:

Marian Edu wrote:

There are procedures for all those cases, in SoapUI project I can see those but not in any of the log files.

I don't see any of these encoded in appsrv/soap/test/web/fwd-soapui-project.xml.

- return values from procedure - do you have this? If not, please add.

The test is there, it is possible to use return value.

Can you point me to which one uses this, in the fwd-soapui-project.xml?

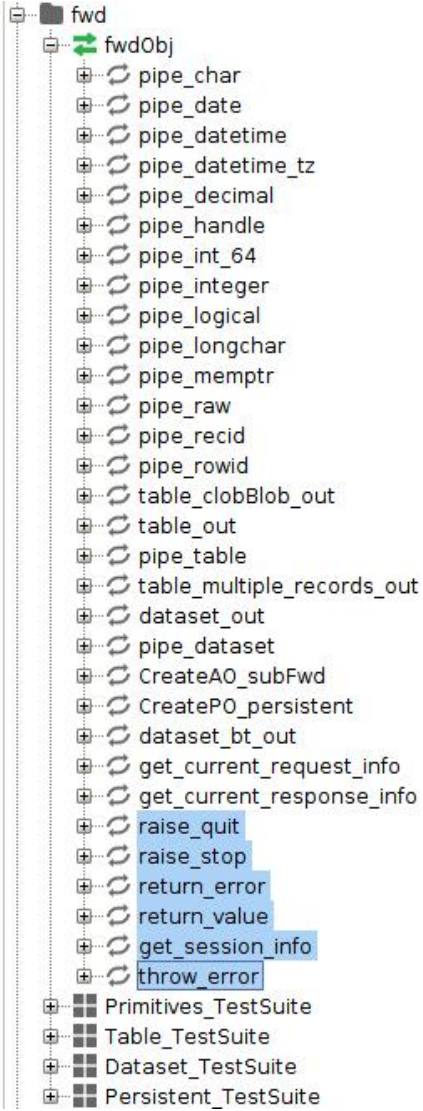
Look for 'raise_error', 'raise_quit', 'raise_stop', 'return_value', 'return_error'...

- File soap-ui-wsa-tests.png added

Marian Edu wrote:

Look for 'raise_error', 'raise_quit', 'raise_stop', 'return_value', 'return_error'...

I see them only like this:



They don't appear as 'runnable' in a _TestSuite.

#33 - 05/21/2020 06:23 AM - Constantin Asofiei

- Assignee set to Constantin Asofiei
- Status changed from New to WIP
- % Done changed from 0 to 90

The conversion and runtime support is in 4231b rev 11551

What remains are some edge-cases (related to fault, legacy error conditions), response payload double-check, and what was mentioned in [#3310-26](#)

#34 - 05/22/2020 08:04 AM - Marian Edu

OK, finally got everything setup again and updated the tests/add new ones as well as the SoapUI project/test cases. You can find the logs for all testcases in appsrv/soap/test/web/. Proxygen definition was updated to use before-image for dataset and temp-table tests that had before image defined, the settings in the API annotation are used by default unless overwritten in the proxygen definition (can be done for each procedure, custom setup). Now, there seems to be a bug in the WSA when you have a temp-table as output and that one has a before table defined (it needs to be part of a dataset, this is not sent as output, just the temp-table). The execution of the procedure on appsrv completes with no errors, before image data is returned but for whatever reason it looks like the WSA is not expecting that and it throws an SOAP error while leaving the result set unconsumed, hence all further requests will fail until the WSA webapp is restarted (or tomcat). The test is the last one in the 'Table TestSuite' so if you ran that against a OE WSA you better run that one last or make sure you restart the WSA afterwards.

Extents are supported both for regular parameters as well as temp-table fields, before image is sent if output is a dataset, I don't know of anyone using that for SOAP though. Added logs for request/response info and exception conditions: error(return, throw), quit, stop.

#35 - 05/22/2020 08:13 AM - Constantin Asofiei

Thanks, Marian.

I'm missing a pipe_char_ext.p file, and in the revision I don't see any changes in the pipe_ tests.

#36 - 05/22/2020 08:23 AM - Marian Edu

Constantin Asofiei wrote:

Thanks, Marian.

I'm missing a pipe_char_ext.p file

I think the file name is pipe_ext.p, did added the before apis (dataset, table) as it turns out bazaar didn't pick files from subfolders on commit :(and in the revision I don't see any changes in the pipe_ tests.

What do you mean, in SoapUI project? I did export it now, looks like save was only saving the workspace or something, added in rev 612.

#37 - 05/22/2020 08:35 AM - Constantin Asofiei

Marian Edu wrote:

Constantin Asofiei wrote:

Thanks, Marian.

I'm missing a pipe_char_ext.p file

I think the file name is pipe_ext.p, did added the before apis (dataset, table) as it turns out bazaar didn't pick files from subfolders on commit :(

There is this entry in fwd.xpxg:

```
<Procedure isPersistent="false" useFullName="false">
  <Name>pipe_char_ext</Name>
  <ProcPath />
  <ProPath>C:\Projects\goldencode\workspace-fwd\testcases\appsrv\api\</ProPath>
</Procedure>
```

but is missing in fwd.wsm and fwd.wsdl. Maybe OE has some checks to not generate the SOAP operation for missing programs?

What do you mean, in SoapUI project? I did export it now, looks like save was only saving the workspace or something, added in rev 612.

OK, so if you didn't change anything in these files, then is OK. I just wanted to double check with you.

#38 - 05/22/2020 08:51 AM - Constantin Asofiei

Marian, please help with these, too:

- what happens if the WSDL is not found, when trying to access it via the URL?

- what happens if the SOAP request is not a valid XML?
- what happens if there is more than one operation in the SOAP body, like this:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:tempuri-org:fwd"
">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:pipe_char>
      <urn:inputValue>234</urn:inputValue>
    </urn:pipe_char>
    <urn:pipe_char>
      <urn:inputValue>456</urn:inputValue>
    </urn:pipe_char>
  </soapenv:Body>
</soapenv:Envelope>
```

- what is the CreateAO operation for a sub-service (look in the WSDL)? Can this ever be used in a request?

#39 - 05/22/2020 08:55 AM - Marian Edu

Constantin Asofiei wrote:

There is this entry in fwd.xpxg:

[...]

but is missing in fwd.wsm and fwd.wsdl. Maybe OE has some checks to not generate the SOAP operation for missing programs?

Ah, yes... was renamed but then left in xpxg. Proxygen skips all entries for which it can't find the r-code (not interested in source code) when it generated the client (soap, java/.net). There might be some warnings in the log file but the entries are missing from WSDL and WSM.

What do you mean, in SoapUI project? I did export it now, looks like save was only saving the workspace or something, added in rev 612.

OK, so if you didn't change anything in these files, then is OK. I just wanted to double check with you.

The project was updated, just had to export it back and then commit it to bazaar. There are a bunch of xpath validations in there and you can probably ran that against FWD by just changing the WsaURL custom property at the project level to point to your server... I've changed some test steps to use that endpoint and updated the SoapUI project.

#40 - 05/22/2020 11:52 AM - Marian Edu

Constantin Asofiei wrote:

- what happens if the WSDL is not found, when trying to access it via the URL?

what do you mean here? you don't need the WSDL to call the web services. renaming/deleting the wsdl file doesn't render the services unusable, just one can't get the definition (as in update definition in soapui will fail).

- what happens if the SOAP request is not a valid XML?

added more exceptions tests, updated the log for exceptions test suite and soapui project.

- what happens if there is more than one operation in the SOAP body, like this:
[...]

it just pick the first entry and ignore everything else

- what is the CreateAO operation for a sub-service (look in the WSDL)? Can this ever be used in a request?

no idea, if it's there then it might but no sample in PSC docs... we can explore if needed although createPO is the way to go with persistent procedures although not really for SOAP.

#41 - 05/22/2020 11:57 AM - Constantin Asofiei

Marian Edu wrote:

Constantin Asofiei wrote:

- what happens if the WSDL is not found, when trying to access it via the URL?

what do you mean here? you don't need the WSDL to call the web services. renaming/deleting the wsdl file doesn't render the services unusable, just one can't get the definition (as in update definition in soapui will fail).

In FWD, I want to know what to do if I access the <https://localhost:7443/ws/notaservice/wsdl?targetURI=urn:tempuri-org> URL. Just open an URL and replace the wsafwd token with something bogus.

- what happens if the SOAP request is not a valid XML?

added more exceptions tests, updated the log for exceptions test suite and soapui project.

- what happens if there is more than one operation in the SOAP body, like this:
[...]

it just pick the first entry and ignore everything else

Thanks!

- what is the CreateAO operation for a sub-service (look in the WSDL)? Can this ever be used in a request?

no idea, if it's there then it might but no sample in PSC docs... we can explore if needed although createPO is the way to go with persistent procedures although not really for SOAP.

Yes, it doesn't make sense for me either, as I don't see it mapped to a 4GL program. I need some tests to know what to return for this.

#42 - 05/22/2020 01:24 PM - Marian Edu

Constantin Asofiei wrote:

In FWD, I want to know what to do if I access the `https://localhost:7443/ws/notaservice/wsdl?targetUri=urn:tempuri-org` URL. Just open an URL and replace the wsafwd token with something bogus.

This is running in tomcat so 404 if not found is the expected result :)

If the wsdl file is not found a plain 404 response is sent back on `/wsdl?targetUri=urn:tempuri-org`, if the targetUri is not right a 'styled' 404 is sent back... I've added a log file for WSDL requests along the others we had.

Yes, it doesn't make sense for me either, as I don't see it mapped to a 4GL program. I need some tests to know what to return for this.

I'll do more tests on Monday, so far this is the only thing I've could found about this... [[

https://documentation.progress.com/output/ua/OpenEdge_latest/index.html#page/dywsv%2Fsubappobject-methods.html%23wwID0E6XTO]]

#43 - 05/22/2020 03:25 PM - Constantin Asofiei

Something I forgot to note. As part of this task, I've added LegacySignature annotations to external programs (and their internal entries), too. This was required to be able to link an external program/internal entry exposed as a service (be it SOAP, REST, etc) without having to rely on resolving the target. Now we have a hard-link between the InternalEntry and the invoked method.

This will pave the road to reduce the name_map.xml size, and let this remain only as a registry of external programs or legacy classes (plus the forward/in-handle/in-super cases).

The downside here is that the bytecode for the entire converted application source code will be loaded at startup. And Method instances will be cached with each InternalEntry.

This requires a heap increase for some of the large projects, but this would be needed either way, for long-running FWD servers: application bytecode gets loaded on a lower pace, but will still end up in memory.

#44 - 05/25/2020 09:06 AM - Constantin Asofiei

Marian, do you have tests to use as input extent vars or tables with extent fields? pipe_ext.p and table_extent.p have only output.

#45 - 05/25/2020 10:16 AM - Constantin Asofiei

BTW, I think we also missed the 'pipe extent' for vars or table fields for REST... please add these, too (input and output versions).

#46 - 05/26/2020 05:46 AM - Marian Edu

Constantin Asofiei wrote:

Marian, do you have tests to use as input extent vars or tables with extent fields? pipe_ext.p and table_extent.p have only output.

Added tests for input, input-output. Parameters can be indeterminate, fields only fixed. For fixed extents it throws errors if not sending one entry for each extent, if sending more it's also an error - unless the whole field is not sent. For indeterminate array zero or more entries can be sent. The order is important, the xml field name is sometimes not relevant as in if you have two extent parameters you can switch their position in request, in that case the order is used not the xml node name. However, if an xml node name that does not match any of defined parameters is used it will throw an error.

For temp-tables looks like the number of fields is first one that is checked (extra fields will cause error), then the fields order must match - if you switch the entries in response it will throw error saying field is not found. Otherwise basic data conversion is supported as for any input parameter.

#47 - 05/26/2020 08:14 PM - Constantin Asofiei

Marian, your previous note is bugging me... assuming that:

- you have 4 input arguments with the same type (int)
- they are defined like a1, a2, a3, a4
- in the SOAP body they appear as XML elements in some random order (like a2=2, a3=3, a1=1, a4=4)

Then they will still be read as a1=2, a2=3, a3=1, a4=4. Is this right? I mean, OE is **not** using the XML element names to identify the arguments, but instead it gets the operation's input message, and just reads stuff from the body, expecting at index idx to have a certain type for that argument... If this is right, this is complete opposite than how I made the validation for the request arguments (I'm using the XML name to identify them).

For table fields, am I correct to assume that they must match both as in number and in order, like:

- all fields must be specified.
- all fields must appear in the SOAP body in the same order as defined in the schema.

But this doesn't seem to apply for datasets... you have an example (pipe_dataset_fields_mismatch) where a table is specified only two fields; and the missing fields are set to their 'default value'.

#48 - 05/26/2020 08:16 PM - Constantin Asofiei

Also, please upload the log for SubAppObject_TestSuite.

#49 - 05/26/2020 08:22 PM - Constantin Asofiei

And one more note: the rowid in FWD is an integer; in OE is a 'byte array' (same as memptr/raw); we will not be able to transport this via SOAP the same way (as SOAP requires it to be base64 for the transport).

#51 - 05/27/2020 09:06 AM - Marian Edu

Constantin Asotiei wrote:

Marian, your previous note is bugging me... assuming that:

- you have 4 input arguments with the same type (int)
- they are defined like a1, a2, a3, a4
- in the SOAP body they appear as XML elements in some random order (like a2=2, a3=3, a1=1, a4=4)

Then they will still be read as a1=2, a2=3, a3=1, a4=4. Is this right? I mean, OE is **not** using the XML element names to identify the arguments, but instead it gets the operation's input message, and just reads stuff from the body, expecting at index idx to have a certain type for that argument... If this is right, this is complete opposite than how I made the validation for the request arguments (I'm using the XML name to identify them).

As far as we've observed from the tests the only exception to strict validation was in the case of those two extent fields, first one is variable (inputValueExt) and second is fixed (inputValueFix). As you can see in the "pipe_ext - switch params" test Progress is happy to switch those and then only consider the order instead of the field names: last two entries are considered as inputValueFix although the xml node name is inputValueExt, and the one before are considered as inputValueExt. Still it does somehow look at the xml node name because sending any entry with a node name that does not match a input/in-out parameter will cause a SOAP error.

For table fields, am I correct to assume that they must be match both as in number and in order, like:

- all fields must be specified.
- all fields must appear in the SOAP body in the same order as defined in the schema.

But this doesn't seem to apply for datasets... you have an example (pipe_dataset_fields_mismatch) where a table is specified only two fields; and the missing fields are set to their 'default value'.

Yes, for dataset it looks like that validation does not apply - you can have missing fields and also new fields and Progress seems only to use whatever it was sent, the order is not important either. For each xml node it will set the value if it can find a field with the same name in the temp-table, if no field matches the node name it simply ignore that and move on. The xml name of the temp-table is important, it's not just using the order of tables in the dataset.

#52 - 05/27/2020 09:13 AM - Marian Edu

Constantin Asofiei wrote:

And one more note: the rowid in FWD is an integer; in OE is a 'byte array' (same as memptr/raw); we will not be able to transport this via SOAP the same way (as SOAP requires it to be base64 for the transport).

I have no idea how do they handle that, it's not just using the input as string and pass it to to-rowid... some values works but I don't understand why. I don't think 234, 'toto' are valid rowid values but still Progress does not throw any error and is happy to assign those values back to the output parameter.

On the other hand I don't expect anyone from outside to be able to send a hand-made rowid, best(worst) case will send a rowid that they've received from the same system before else it doesn't make any sense.

#53 - 05/27/2020 10:19 AM - Constantin Asofiei

Marian Edu wrote:

As far as we've observed from the tests the only exception to strict validation was in the case of those two extent fields, first one is variable (inputValueExt) and second is fixed (inputValueFix). As you can see in the "pipe_ext - switch params" test Progress is happy to switch those and then only consider the order instead of the field names: last two entries are considered as inputValueFix although the xml node name is inputValueExt, and the one before are considered as inputValueExt. Still it does somehow look at the xml node name because sending any entry with a node name that does not match a input/in-out parameter will cause a SOAP error.

I don't mean to pass a node name which doesn't exist in the message's schema type (i.e. not a valid parameter name). I mean to interchange existing parameter names, with the same type. If my assumption is correct, OE reads the SOAP body in order (based on the schema for the message input type), and it looks at the node's XSD type; if these match (e.g. int vs. int), then it uses the argument.

#54 - 05/27/2020 02:35 PM - Constantin Asofiei

Marian, to check the above, please use a pipe_two_ints.p program like this:

```
def input-output p1 as int.  
def input-output p2 as int.
```

and test it with a payload like this:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:tempuri-org:fwd">  
  <soapenv:Header/>
```

```

<soapenv:Body>
  <urn:pipe_two_ints>
    <urn:p1>1</urn:p1>
    <urn:p1>2</urn:p1>
  </urn:pipe_two_ints>
</soapenv:Body>
</soapenv:Envelope>

```

If the response is like this:

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <pipe_two_intsResponse xmlns="urn:tempuri-org:fwd">
      <result xsi:nil="true"/>
      <p1>1</p1>
      <p2>2</p2>
    </pipe_two_intsResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

then my assumption is correct.

#55 - 05/28/2020 03:11 PM - Constantin Asofiei

Second pass for SOAP support is in 4231b rev 11569

- The SOAP input are the .wsm files, not .xpxg (as these contain info configured when running ProxyGen).
- SOAP fixes for extent parameters, BEFORE-TABLE and error management.

What remains are the rules how to exactly interpret the SOAP body, when parsing the arguments.

#56 - 06/08/2020 07:06 AM - Constantin Asofiei

To determine how OE interprets the SOAP envelope, when resolving the arguments, we need at least:

- all arguments as input-output (so we can check their response value against the one at the request).
- arguments of the same type (int) and also int vs char (to check if OE verifies the argument's XSD type). For example, if OE uses the argument order and its schema doesn't match the input message definition.
- argument order:

- use the same argument name in the request, but the number of arguments matches the input message definition - does OE read the arguments 'in order'?
- use the argument names in 'reverse order' - how does OE read this? By name or by order?
- if OE reads the arguments in order, and not by their name, what happens if at a certain position is an argument name for a different type (char) and it expects an int (from the input message definition)?
- for extent values:
 - can you mix them, like:

```
<p1>10</p1>
<px1>1</px1>
<p2>10</p1>
<px1>2</px1>
<p3>10</p1>
<px1>3</px1>
<px1>4</px1>
```

where px1 is an extent 4 value.

- if OE doesn't rely on the argument name, what happens if it expects to read an extent and the next one to read is non-extent, or it wants to read a non-extent, and the next one is an extent?
- what happens if for a fixed extent, there are not enough values to read?

By 'input message definition', if all arguments are input-output, then this matches the real 4GL signature. Otherwise, 'input message definition' in the WSDL will contain all INPUT and INPUT-OUTPUT arguments, in their definition order.

Same for standalone table arguments or part of a dataset (extent fields, mixing fields so that it doesn't match the schema, missing fields, using the same field more than once, etc).

Marian: please create a separate test suite in Soap UI to cover these. The idea above is that I need variations of mixing the arguments (and anything else you can think of that it may be relevant).

#57 - 06/11/2020 08:25 AM - Marian Edu

Constantin Asofiei wrote:

To determine how OE interprets the SOAP envelope, when resolving the arguments, we need at least:

- all arguments as input-output (so we can check their response value against the one at the request).

There are now tests for this case, as expected input-output are nothing more than input in request and output in response. All input, input-output parameters are added in order to the request and the same for input-output and/or output parameters in response.

- arguments of the same type (int) and also int vs char (to check if OE verifies the argument's XSD type). For example, if OE uses the argument order and its schema doesn't match the input message definition.

I have no idea if the validation is done using XSD schema definition or something else, there are plenty of tests of this in "pipe" requests for all data types. The only data conversions that are accepted as far as I've saw are for integer/int64/decimal when passing an integer and date/datetime/datetime-tz (you can send time/tz info when not expected and will be ignored). If the time or time zone part is missing for a datetime-tz field the operation will fail, if time/time zone is sent for a data field then it works and only use the date part from the input value.

- argument order:
 - use the same argument name in the request, but the number of arguments matches the input message definition - does OE read the arguments 'in order'?
 - use the argument names in 'reverse order' - how does OE read this? By name or by order?

For input/input-output parameters the order is as important as the parameter name, the only exception is when the parameters have the same data type - in that case you can switch them and the values will be assigned in order rather than using the xml node name. Aka, if you have in1 and in2 as input having the same data type you can send then as `<int2>val2</in2><in1>val1</in1>` and on 4GL side in1 will be 'val2' and in2 'val1', so the order prevails and that apply also for extent fields. If the data types does not match it will throw error if you switch them.

- if OE reads the arguments in order, and not by their name, what happens if at a certain position is an argument name for a different type (char) and it expects an int (from the input message definition)?

That was already in the test suite, it will fail saying the value is incorrect.

- for extent values:
 - can you mix them, like:
[...]
where px1 is an extent 4 value.

It fails with SOAPError, it's a server side fault but then not much detail other than 'an error occurred' :(

- if OE doesn't rely on the argument name, what happens if it expects to read an extent and the next one to read is non-extent, or it wants to read a non-extent, and the next one is an extent?

If multiples values are sent for a non-extent it throws error. If single (non-extent) value is sent for an extent field it will only work for an indeterminate extent.

- what happens if for a fixed extent, there are not enough values to read?

For fixed extents the number of entries sent must match, it throws error is less or more are sent.

By 'input message definition', if all arguments are input-output, then this matches the real 4GL signature. Otherwise, 'input message definition' in the WSDL will contain all INPUT and INPUT-OUTPUT arguments, in their definition order.

Same for standalone table arguments or part of a dataset (extent fields, mixing fields so that it doesn't match the schema, missing fields, using the same field more than once, etc).

Mihai added tests for that, as opposed to when the input is a temp-table where validations is damn strict when it's part of a dataset you can pretty much anything there. If there are entries for tables that do not exist in dataset will ignore those, if table rows are empty in xml it will create a record in temp-table with all fields set to default (will probably fail if there is a unique index but that is standard 4gl). If you send fields that are not in the temp-table it will ignore those and only set those that are being filled.

Now, one interesting thing here is that you can send the value for the same field multiple times - multiple xml nodes with that field name in a table row, it can be one after another or mixed anywhere in the table record. This works for extent and non-extents fields, for non-extents the last entry will win, for extent the assignment of values in the array start on each sequence and will set the values started from first index overriding those set in previous sequence. It will only fail if any of those sequences has more entries than the field extent. If last sequence has less entries than the previous one then

only the values for those indexes will be overwritten, the ones with higher index from previous sequences will survive (it does not reset all values from the array).

Marian: please create a separate test suite in Soap UI to cover these. The idea above is that I need variations of mixing the arguments (and anything else you can think of that it may be relevant).

i believe the test suite is PassingArguments, I've also updated some in primitive/temp-table and dataset with some more cases :(

#58 - 06/20/2020 11:54 AM - Greg Shah

Task branch 4231b has been merged to trunk as revision 11347.

#59 - 11/25/2020 03:57 PM - Greg Shah

For conversion, do I understand correctly that this is the process:

1. place the WSA artifacts (.wsm and .xpxg files) in the project (e.g. ./abl/wsa/)
2. add a soap-cfg entry in cfg/p2j.cfg.xml to point to the artifacts and activate the WSA conversion processing (<parameter name="soap-cfg" value="./abl/wsa/">)

After the conversion is done, all the SOAP mapping and API exports will be defined as annotations in the converted Java source code.

Is there any runtime configuration needed to deploy this?

#60 - 11/25/2020 04:55 PM - Constantin Asofiei

For SOAP, only the .wsm files are required for conversion.

For runtime, you need an entry like this under e.g. server/default:

```
<node class="container" name="soap">
  <node class="boolean" name="enabled">
    <node-attribute name="value" value="TRUE"/>
  </node>
  <node class="string" name="wsdl">
    <node-attribute name="value" value="srv1.wsdl,srv2.wsdl,srvx.wsdl"/>
  </node>
  <node class="string" name="appserver">
    <node-attribute name="value" value="app_server"/>
  </node>
  <node class="string" name="alias">
    <node-attribute name="value" value="ehotel"/>
  </node>
</node>
```

where:

- soap/wsdl, a comma-separated list of WSDL files, relative to the root Java package.
- soap/basepath, to specify the basepath. Defaults to /ws
- soap/alias, a process certificate alias associated with the workers which will process the SOAP requests; these will delegate the call to the appserver. This context will automatically be set as HEADLESS.
- soap/keyStore, optional, the key store with the certificate to authenticate this process. If is missing, the store will be built from the directory.

- soap/storePassword, the store password, only if soap/keyStore is set.
- soap/poolSize - the pool size of workers. Defaults to 10.
- soap/appserver - the name of the appserver handling the requests.

#62 - 11/27/2020 09:41 AM - Greg Shah

soap/wSDL, a comma-separated list of WSDL files, relative to the root Java package.

The WSDL files are generated by OE, right?

If they are not used in the OE environment, then we don't need this?

If we do have these files, do we need to modify the project's build.xml to include these into the jar?

soap/basepath, to specify the basepath. Defaults to /ws

Can I pull this from the artifacts or do I have to find this in the OE appserver configuration?

soap/appserver - the name of the appserver handling the requests.

I assume this must be one of the entries in /server/standard/appservers/? In the case where we have more than one configured, can I determine this from the OE appserver configuration?

soap/alias, a process certificate alias associated with the workers which will process the SOAP requests; these will delegate the call to the appserver.

I'm not sure where to get this value from.

Greg Shah wrote:

soap/wsdl, a comma-separated list of WSDL files, relative to the root Java package.

The WSDL files are generated by OE, right?

No, they are generated during conversion.

If they are not used in the OE environment, then we don't need this?

They are reported by OE, too (I don't know how/where are kept in OE, but they are available via the /wsdl path). FWD provides this, too. More, the .wsdl files are the registry of all exposed SOAP operations. Maybe we can re-generate them at runtime, from the annotations, but I think some info exists only in the .wsm files.

If we do have these files, do we need to modify the project's build.xml to include these into the jar?

Yes, they need to be in the jar. If build.xml doesn't include them, it needs to be fixed.

soap/basepath, to specify the basepath. Defaults to /ws

Can I pull this from the artifacts or do I have to find this in the OE appserver configuration?

This is from the OE configuration. You need to find an entry like wsaRoot=http://host:port/soap and get the path from it (soap).

soap/appserver - the name of the appserver handling the requests.

I assume this must be one of the entries in /server/standard/appservers/? In the case where we have more than one configured, can I determine this from the OE appserver configuration?

Yes, look in the OE configuration for something like appservername.ROOT.SOAP.

soap/alias, a process certificate alias associated with the workers which will process the SOAP requests; these will delegate the call to the appserver.

I'm not sure where to get this value from.

This is something like ehotel - just the alias for a process account. In FWD is needed to authenticate the worker thread with the FWD server, and obtain a security context.

#64 - 11/27/2020 10:08 AM - Greg Shah

soap/alias, a process certificate alias associated with the workers which will process the SOAP requests; these will delegate the call to the appserver.

I'm not sure where to get this value from.

This is something like ehotel - just the alias for a process account. In FWD is needed to authenticate the worker thread with the FWD server, and obtain a security context.

This is the alias of the process account of the **caller** process (as opposed to the appserver process)?

#65 - 11/27/2020 10:09 AM - Constantin Asofiei

Greg Shah wrote:

This is the alias of the process account of the **caller** process (as opposed to the appserver process)?

Correct.

#66 - 10/12/2021 04:02 AM - Constantin Asofiei

Marian, I need some help - a client is using a DATASET-HANDLE and TABLE-HANDLE as parameters for SOAP.

I need some tests with INPUT/OUTPUT/INPUT-OUTPUT for these, and also the associated wsdl, the .wsm file, and the raw request/response log from SoapUI.

#67 - 10/12/2021 09:00 AM - Marian Edu

Constantin Asofiei wrote:

Marian, I need some help - a client is using a DATASET-HANDLE and TABLE-HANDLE as parameters for SOAP.

I need some tests with INPUT/OUTPUT/INPUT-OUTPUT for these, and also the associated wsdl, the .wsm file, and the raw request/response log from SoapUI.

Constantin, working on this but I have troubles setting-up the environment as I've been through an OS reinstall since last time working on this so need to setup all the appsrv's from scratch. And also missed the SOAP part and started with the REST mapping for PASOE and that damn instance doesn't want to start anymore as soon as I publish the webapp into it :(

I'll wave when everything is pushed in testcases.

#68 - 10/12/2021 09:55 AM - Constantin Asofiei

Marian, please add some tests which have unique/primary indexes at a table (part of a dataset or not) - I see that in the WSDL indexes are exported only for dataset tables, and I need to see a little what impact the runtime has (at the request/response payloads).

Include unique/primary indexes for dataset/table-handle, too.

#69 - 10/12/2021 10:37 AM - Constantin Asofiei

And the desc clause for an index...

#70 - 10/12/2021 10:56 AM - Constantin Asofiei

And relations in a dataset... a client uses these.

#71 - 10/13/2021 08:09 AM - Marian Edu

Constantin Asofiei wrote:

And relations in a dataset... a client uses these.

So, relations and unique indexes are visible in wsdl as "unique" and "keyref" in dataset's complex definition. As expected, non unique indexes or sort information doesn't appear there. For dataset handle there is no schema definition since that is a dynamic object, this is the defined as a two-part schema object - first is the definition then the actual data:

```
<complexType name="DataSetHandleParam"><annotation><documentation>This is the schema definition for an OpenEdge dynamic ProDataSet parameter. The first element in this sequence must be a w3c XML Schema document describing the definition of the ProDataSet. The second element contains the serialized data.</documentation></annotation><sequence><any maxOccurs="2" minOccurs="2"/></sequence></complexType>
```

WSDL and everything committed in testcases rev #1149, for actual output I will try to get the PASOE working as this is now the only option we have available (did remember it was not an OS reinstall but an OpenEdge upgrade, classical AppSrv is not part of our OESDK bundle).

#72 - 10/14/2021 09:03 AM - Constantin Asofiei

Marian, another request, please: proxygen uses on the XML-NODE-NAME attribute (at the dataset, table or field). So please create a copy of the tests which have the dataset relation and add XML-NODE-NAME (something obvious and not similar to the actual 4GL name for the dataset/table/field). The same for a pipe_table test, a copy with XML-NODE-NAME added.

Also, I don't see the table-handle tests, please add them too, when you have time.

#73 - 10/14/2021 09:48 AM - Marian Edu

Constantin Asofiei wrote:

Marian, another request, please: proxygen uses on the XML-NODE-NAME attribute (at the dataset, table or field). So please create a copy of the tests which have the dataset relation and add XML-NODE-NAME (something obvious and not similar to the actual 4GL name for the dataset/table/field). The same for a pipe_table test, a copy with XML-NODE-NAME added.

Also, I don't see the table-handle tests, please add them too, when you have time.

Added serialization options (serialize-name is the default for xml/json) - only works for datasets as far as I can see not plain temp-table input-output. Also the table-handle as input/output, same as for dataset that is a two-part complex type - first schema then data.

Pushed rev #1150.

#74 - 10/19/2021 10:44 AM - Constantin Asofiei

Marian, did you manage to get the SOAP tests for these programs?

#75 - 10/19/2021 10:47 AM - Marian Edu

Constantin Asofiei wrote:

Marian, did you manage to get the SOAP tests for these programs?

Sort of managed to deploy the WSM to PASOE - looks like the issue was with the sub-object that is not supported, at least that lead to a plain N.P.E and deployment failed mid-way.

I will try to get the requests and responses for those new endpoints added tomorrow, will let you know when available.

#76 - 10/25/2021 07:18 AM - Constantin Asofiei

Marian, I think these programs are missing from xfer:

```
pipe_serialize_table  
pipe_table_handle
```

#77 - 10/25/2021 08:26 AM - Marian Edu

Constantin Asofiei wrote:

Marian, I think these programs are missing from xfer:
[...]

Right, added those. Mihai will add them to proxygen and make requests for those in SoapUI.

#78 - 10/25/2021 10:00 AM - Constantin Asofiei

Marian Edu wrote:

Right, added those. Mihai will add them to proxygen and make requests for those in SoapUI.

Thanks, please make sure that the appsrv/soap/test/web/fwd-soapui-project.xml project is the latest one you have and all tests pass in OE (any changes are saved); also, please re-generate all the logs, as I think some of them are out-of-sync (see pipe_char_null for which the request XML does not match what is in the fwd-soapui-primitives.log, and it fails in FWD).

#79 - 10/25/2021 10:40 AM - Constantin Asofiei

I'm logging it here, because I found a flaw in ControlFlowOps when executing the SOAP requests - if argument validation fails, then the pending resources must be cleared. Previously this was done only if the default constructor (when creating a new instance for the external program, for example) failed.

See 3821c/13107.

#80 - 10/25/2021 05:35 PM - Greg Shah

Code Review Task Branch 3821c Revision 13107

The change is good.

#81 - 10/26/2021 03:40 AM - Marian Edu

Constantin Asofiei wrote:

Thanks, please make sure that the appsrv/soap/test/web/fwd-soapui-project.xml project is the latest one you have and all tests pass in OE.

Constantin, the assertions there were made for 'classical appsrv' responses and we do not have this available as it was retired from the latest OE versions. What we did now is to deploy the SOAP services inside a PASOE instance but there are some differences that we see now:

1. The PASOE SOAP interface does not accept sub-objects when deploying the .wsm file generated with proxygen so we had to take those out from proxygen definition, will have to remove them from SoapUI tests as well.
2. The SOAP errors seems to be slightly different.
3. Temp-table/dataset structure match seems to be a bit more relaxed - the order or nodes in xml had to match the fields order before, now as long as the field names match the order in XML node is not important.

There might be other but this is what we see at first look.

Now, we can change the assertions in SoapUI to match the PASOE response. Leave the SoapUI project as is for 'classical appsrv' and make a new one for PASOE. However, if you want to have the output regenerated for classical appsrv this can't be done on our side as we do not have that product available :(

#82 - 10/26/2021 03:42 AM - Constantin Asofiei

Marian Edu wrote:

Now, we can change the assertions in SoapUI to match the PASOE response. Leave the SoapUI project as is for 'classical appsrv' and make a new one for PASOE. However, if you want to have the output regenerated for classical appsrv this can't be done on our side as we do not have that product available :(

Thanks, I understand. Just make sure that the logs and SoapUI project are in sync for testing with PASOE.

#83 - 10/26/2021 06:23 AM - Greg Shah

We need to support both classic appserver and PASOE. Both our tests and the FWD implementation must be able to be "switched" between the two types.

#84 - 10/26/2021 06:57 AM - Marian Edu

Greg Shah wrote:

We need to support both classic appserver and PASOE. Both our tests and the FWD implementation must be able to be "switched" between the two types.

Understand, then we make a new soapui suite for PASOE - what we can't help with though is re-run the suite on classic appserver and afraid this one is missing responses for the newly added endpoints using serialize-name and table/dataset handles.

#85 - 10/26/2021 08:24 AM - Greg Shah

We will need to run this for a customer that is using classic appserver and we can run it on their dev environment.

#86 - 10/27/2021 09:03 AM - Marian Edu

Constantin Asofiei wrote:

Marian Edu wrote:

Now, we can change the assertions in SoapUI to match the PASOE response. Leave the SoapUI project as is for 'classical appsrv' and make a new one for PASOE. However, if you want to have the output regenerated for classical appsrv this can't be done on our side as we do not have that product available :(

Thanks, I understand. Just make sure that the logs and SoapUI project are in sync for testing with PASOE.

Mihai added the tests suite and logs for PASOE, should be available in testcases #rev1164 (the names have a 'pasoe' prepended).

#87 - 11/02/2021 11:14 AM - Constantin Asofiei

A bug which affects dataset_bt_out is logged as [#5799](#).

#88 - 11/02/2021 11:24 AM - Constantin Asofiei

- Related to Bug #5801: write-xml is not using XML-NODE-NAME for the temp-table added

#89 - 11/02/2021 11:24 AM - Constantin Asofiei

- Related to Bug #5799: write-xml doesn't serialize the prods:id attributes and the deleted fields for before-table added

#90 - 11/02/2021 11:31 AM - Constantin Asofiei

Marian: from SOAP's PASOE logs, the response for pipe_dataset_handle is:


```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/200
1/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header />
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>An error was detected while executing the Web Service request. (10893)</faultstring>
      <detail>
        <ns1:FaultDetail xmlns:ns1="urn:soap-fault:details">
          <errorMessage>Exception running ABL procedure</errorMessage>
          <requestID>42105b3e-9787-4019-8537-cf40f10f09d9</requestID>
        </ns1:FaultDetail>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

For classic appsrv, the response has the actual dataset (which is weird, as this is **not** configured in the SoapUI project XML file). Is this something wrong in the PASOE log? There are others in the log (like dataset_order) which are the same, the log shows a SOAP Fault message instead of the DATASET.

Please double-check if everything is committed to bzz on xfer.

#91 - 11/03/2021 04:57 AM - Mihai Popescu-Tiganea

Constantin, logs were checked, corrected and committed in bzz with revision nr 1166. The error was thrown because few tests had invalid endpoint in SoapUi.

#92 - 11/12/2021 03:07 PM - Constantin Asotiei

3821c/13156 adds fixes for SOAP and REST support:

- fixed differences between PASOE and classic appserver, when error conditions are raised.
- fixed cases when the XML node name differences from the parameter name (XML-NODE-NAME and SERIALIZE-NAME at the dataset, table and field), for SOAP calls.
- SOAP transfer for DATASET, TABLE, DATASET-HANDLE and TABLE-HANDLE is made via XML, so that relations, indexes, schema, etc are all processed by the server-side.
- other misc fixes for READ/WRITE-XML methods for DATASET and TABLE

I've also made a small program to generate harness tests from the SoapUI logs. I've added the generated tests to xfer testcases rev 1168, including the logs.

#93 - 04/27/2022 01:53 PM - Constantin Asofiei

Greg, in OE the proxy Java Open Client allows a 'SubAppObject' which groups access to the programs in its own Java class. I don't see anything special about this, except a way of allowing grouping together i.e. CRUD operations on the same table, while the 4GL programs for these CRUD operations being separate .p files.

proxygen still emits Java classes for these standalone 4GL programs (grouped in the SubAppObject), but it prefixes them with the SubAppObject name, so you will have 'book_create', 'book_delete', etc for a 'book' SubAppObject. In FWD, I don't see a reason to emit this 'book' class which just allows to call 'book.create' instead of working directly with the 'book_create' Java class.

Instead, I can either place them in sub-packages or just prefix them with the SubAppObject name, to have 'book_create' Java classes.

#94 - 04/27/2022 03:19 PM - Greg Shah

In FWD, I don't see a reason to emit this 'book' class which just allows to call 'book.create' instead of working directly with the 'book_create' Java class.

I think the primary reason is just for compatibility. Customers that have code that depends upon this naming/structure will have less work to use FWD if we match the same approach.

#95 - 04/27/2022 03:21 PM - Constantin Asofiei

Greg Shah wrote:

In FWD, I don't see a reason to emit this 'book' class which just allows to call 'book.create' instead of working directly with the 'book_create' Java class.

I think the primary reason is just for compatibility. Customers that have code that depends upon this naming/structure will have less work to use FWD if we match the same approach.

Thanks, I'll look at how to generate these, too.

#96 - 04/28/2022 06:59 AM - Constantin Asofiei

Constantin Asofiei wrote:

Greg Shah wrote:

In FWD, I don't see a reason to emit this 'book' class which just allows to call 'book.create' instead of working directly with the 'book_create' Java class.

I think the primary reason is just for compatibility. Customers that have code that depends upon this naming/structure will have less work to use FWD if we match the same approach.

Thanks, I'll look at how to generate these, too.

And it makes sense as the same program can appear in more than one .xpxg files, so it will be emitted (and grouped) with that AppObject generated code. So you can have a commit.p program which will exist as book.commit, author.commit, if you include it in book and author .xpxg files.

Need to test if the same happens with SubAppObject, but I assume it does.

#97 - 10/09/2023 02:38 PM - Greg Shah

Is the "qualified prefixing" feature from [#3310-93](#) the only remaining feature? I'm trying to understand what is left to do here.

#98 - 10/09/2023 02:43 PM - Constantin Asofiei

Greg Shah wrote:

Is the "qualified prefixing" feature from [#3310-93](#) the only remaining feature? I'm trying to understand what is left to do here.

SubAppObject is already supported in FWD.

#99 - 10/09/2023 03:51 PM - Greg Shah

Is there anything left to do on this task?

#100 - 10/10/2023 02:12 AM - Constantin Asofiei

Greg Shah wrote:

Is there anything left to do on this task?

I don't think so.

#101 - 10/10/2023 08:24 AM - Greg Shah

- Status changed from WIP to Closed

- % Done changed from 90 to 100

Files

soap-ui-wsa-tests.png	16.9 KB	05/15/2020	Constantin Asofiei
-----------------------	---------	------------	--------------------