

Base Language - Feature #6417

cleanup some known items in appserver support

05/26/2022 03:15 PM - Greg Shah

Status:	WIP	Start date:	
Priority:	Normal	Due date:	
Assignee:	Galya B	% Done:	70%
Category:		Estimated time:	0.00 hour
Target version:		vendor_id:	GCD
billable:	No		
Description			
Related issues:			
Related to Base Language - Feature #3854: update appserver support with PASOE...			Closed

History

#1 - 05/26/2022 03:15 PM - Greg Shah

- Related to Feature #3854: update appserver support with PASOE features added

#2 - 05/26/2022 03:17 PM - Greg Shah

- Subject changed from finish PASOE features/cleanup to cleanup some known items in appserver support

Per [#3854-11](#), the following appserver support cleanup needs to be finished:

- versioninfo fields are read-only, but writing them is a runtime error and not compile error in FWD.
- oerequestinfo: adapterType requires enum, which is not supported by FWD yet
- agentId, sessionId, threadId: these are 'plugged in' but the Agent doesn't set the to anything (remain unknown).
- VersionInfo:clientType supports only 4GLCLIENT and APPSERVER at this time
- SESSION:CURRENT-REQUEST/RESPONSE-INFO don't know yet how this can be unknown or not
- some errors throw by SERVER:CONNECT are not the same
- argument translation to remote definition type I think is done the same as the CALL statement does it...
- OO instance serialization and transmission to remote side (GES: are we already handling this in [#4658](#)?)
- a QUIT on the remote side quits the client
- misc errors related to the ERROR, STOP, RETURN ERROR, raised by the remote side

These are items that can be found using the appserver test suite. The suite should run correctly/completely when this task is done.

#3 - 12/07/2023 07:08 AM - Galya B

- Status changed from New to WIP

- Assignee set to Galya B

#4 - 12/08/2023 06:40 AM - Galya B

- argument translation to remote definition type I think is done the same as the CALL statement does it...

What does that mean?

#5 - 12/11/2023 03:15 AM - Galya B

- versioninfo fields are read-only, but writing them is a runtime error and not compile error in FWD.

This one seems to have been changed.

Compiling session/local_version_info.p in testcases fails with:

```
[javac] /home/gbb/Code/testcases-working/src/com/goldencode/testcases/session/LocalVersionInfo.java:97: error: method getOeclientType in class OeversionInfo cannot be applied to given types;
[javac]         silent(() -> oeVersionInfo.ref().getOeclientType(new character("javaclient")));
[javac]         ^
[javac]     required: no arguments
[javac]     found: character
[javac]     reason: actual and formal argument lists differ in length
[javac] /home/gbb/Code/testcases-working/src/com/goldencode/testcases/session/LocalVersionInfo.java:100: error: method getOemaintVersion in class OeversionInfo cannot be applied to given types;
[javac]         silent(() -> oeVersionInfo.ref().getOemaintVersion(new character("test")));
[javac]         ^
[javac]     required: no arguments
[javac]     found: character
[javac]     reason: actual and formal argument lists differ in length
[javac] /home/gbb/Code/testcases-working/src/com/goldencode/testcases/session/LocalVersionInfo.java:103: error: method getOemajorVersion in class OeversionInfo cannot be applied to given types;
[javac]         silent(() -> oeVersionInfo.ref().getOemajorVersion(new character("test")));
[javac]         ^
[javac]     required: no arguments
[javac]     found: character
[javac]     reason: actual and formal argument lists differ in length
[javac] /home/gbb/Code/testcases-working/src/com/goldencode/testcases/session/LocalVersionInfo.java:106: error: method getOeminorVersion in class OeversionInfo cannot be applied to given types;
[javac]         silent(() -> oeVersionInfo.ref().getOeminorVersion(new character("test")));
[javac]         ^
[javac]     required: no arguments
[javac]     found: character
[javac]     reason: actual and formal argument lists differ in length
[javac] 4 errors
```

For some reason the assignment of new values (oeVersionInfo:OEClientType = 'javaclient' no-error.) is converted into calling the getter methods, but the important part is that the compilation fails for the four Progress.Lang.OEVersionInfo attributes.

If this is how it should work, then the test needs to be changed to not expect a runtime error.

- VersionInfo:clientType supports only 4GLCLIENT and APPSERVER at this time

Access to clientType via:

- SESSION:CURRENT-REQUEST-INFO (Progress.Lang.OERequestInfo) : VersionInfo property (Progress.Lang.OEVersionInfo) : OEClientType property
- SESSION:LOCAL-VERSION-INFO (Progress.Lang.OEVersionInfo) : OEClientType property

Value	OE Client Type	FWD
"4GLCLIENT"	ABL Client	AppServerManager
"JAVACLIENT"	Java Open Client	LegacyJavaAppserver,
".NETCLIENT"	NET Open Client	Not supported, #3326
"AIA"	AIA (AppServer Internet Adapter)	LegacyWebServiceHandler
"WSA"	WSA (Web Services Adapter)	WebServiceHelper / SOAP
"ESB"	ESB (Adapter for Sonic ESB)	Not supported, #3775
"APPSERVER"	AppServer agent	AppServerManager / the startup proc???
"WEBSPEED"	WebSpeed agent	converted to WebHandler endpoints / the startup proc???
"MULTI-SESSION-AGENT"	Progress Application Server agent	PASOE uses HTTP in FWD, so all HTTP requests are handled by processing concurrently multiple requests.
Unknown value (?)	Other special-purpose clients	?

To review [Appserver_Support](#).

The client type can be accessed from both the session and the request.

Is it possible to associate the session in FWD with only one type of web requests / client? What about concurrent execution?

Where does RemoteWebRequest fit?

#7 - 12/15/2023 07:04 AM - Galya B

WebSpeed used to be OE app in the browser (in some script) running on the WebSpeed agent. So the client knew it is WebSpeed. In FWD instead of a live app we have dead endpoints. What is WebSpeed session then? I think this clientType is not applicable. Am I missing something?

#8 - 01/03/2024 06:47 AM - Constantin Asofiei

Galya, these are the steps to run the web tests from xfer testcases; some info is at [Testcases](#), but some is obsolete; please follow these steps, if anything is confusing, let me know; I'll update the wiki once you have a running system.

- checkout xfer testcases project
- edit build.properties and set db.names=tstcasesdb
- copy file-cvt-list.txt.soap.rest.webhandler as file-cvt-list.txt
- run ant deploy.all
- for SOAP, make sure 'soap:address' is set to '/wsa/wsafwd' in fwd.wsdl. After that, run ant jar
- run ./install_spawner.sh
- in deploy/server, do cp directory.xml.template directory.xml
- edit directory.xml and:
 - replace "[os-user]" with your OS username
 - replace "[log-console]" with "true"
 - replace "[p2j-sport]" with "3333"
 - replace "[spawner-path]" with "/opt/spawner/spawn"
 - replace "[client-start-dir]" with your FWD client dir
 - fix "jdbc:h2:[client-start-dir]/deploy/db/tstcasesdb;DB_CLOSE_DELAY=-1;MV_STORE=FALSE;IFEXISTS=TRUE"
 - set REST/enabled to true
 - set SOAP/enabled to true
 - set webHandler/enabled to true
 - set scheduler/start_appserver/enabled to true
 - set processes/appserver_agent/enabled to true
 - set processes/appserver_process/enabled to true
 - change admin port from 7443 to 9443
- start the FWD server
 - check <https://localhost:9443/rest/goldencodeService/pipe/char/mihai> if it works
- install SoapUI
- the project files are:

```
./appsrv/soap/test/web/fwd-soapui-project.xml
./appsrv/soap/test/web/pasoe-fwd-soapui-project.xml
./appsrv/rest/test/fwd-soapui-project.xml
```

Import these in SoapUI and see if these work. There is also a file-cvt-list.txt.rest_and_appserver, to run the plain appserver client tests, but I haven't run these yet. Maybe you can give it a go and post the details here?

#9 - 01/03/2024 09:11 AM - Galya B

- some errors throw by SERVER:CONNECT are not the same

Tests in testcases/appsrv/test/connect_errors/ are now successful with 6417a r14907 & r14908, except for one condition in session_model_mismatch.p. I don't know how to set return-value to the error message. The actual return value of ServerImpl.connect is a boolean.

#10 - 01/03/2024 09:30 AM - Galya B

Constantin Asofiei wrote:

Import these in SoapUI and see if these work. There is also a file-cvt-list.txt.rest_and_appserver, to run the plain appserver client tests, but I haven't run these yet. Maybe you can give it a go and post the details here?

I'll plan to run this batch of tests later on, because I guess it will present more issues for solving, instead of answers at this stage. It won't help with [#6417-5](#) that is to make clear the requirement (do we want a runtime or compile error), or [#6417-6](#) and [#6417-7](#) because OEClientType tests are written in a way they can't fail (obviously not testing all types of clients). I will get back to those points when the rest is done.

#11 - 01/03/2024 10:55 AM - Constantin Asofiei

Galya B wrote:

It won't help with [#6417-5](#) that is to make clear the requirement (do we want a runtime or compile error),

We need to add the setter and make it just throw an ERROR condition - there is no error message displayed/logged. I can't explain why they would not leave this a hard compile error if the property is supposed to be read-only.

or [#6417-6](#)

There are tests IIRC for SESSION:CURRENT-REQUEST-INFO and SESSION:LOCAL-VERSION-INFO, for REST/SOAP/WEB.

#12 - 01/05/2024 04:14 AM - Galya B

Greg Shah wrote:

- a QUIT on the remote side quits the client

Solved in r14910. appsrv/test/test_raise_quit.p now is successful.

#13 - 01/05/2024 08:03 AM - Galya B

- versioninfo fields are read-only, but writing them is a runtime error and not compile error in FWD.

Solved in r14911. session/local_version_info.p now successful.

#14 - 01/05/2024 08:56 AM - Galya B

Constantin Asofiei wrote:

Import these in SoapUI and see if these work. There is also a file-cvt-list.txt.rest_and_appserver, to run the plain appserver client tests, but I haven't run these yet. Maybe you can give it a go and post the details here?

Actually, the set to be run is in appsrv/runall.p, because the lists for conversion contain procedures that can't be called independently, for example appsrv/api/dataset/get_current_request_info.p, which needs an argument. That's why it should be called from appsrv/test/dataset/test_get_current_request_info.p.

This is just a note.

#15 - 01/08/2024 02:43 AM - Galya B

- oerequestinfo: adapterType requires enum, which is not supported by FWD yet

I reworked com.goldencode.p2j.oo.applicationserver.AdapterTypes in r14912.

The only related tests I've found are `appsrv/api/dataset/get_current_request_info.p` and `appsrv/api/dataset/get_current_response_info.p` and they are not failing.

The skeleton might need to be changed on merge to include all the numeric associations.

oo4gl/Progress/ApplicationServer

```
enum Progress.ApplicationServer.AdapterTypes:
  define enum
    Unexpected = 0

  APSV

  SOAP

  REST

  WEB.

end class.
```

#16 - 01/08/2024 03:26 AM - Galya B

- `agentId`, `sessionId`, `threadId`: these are 'plugged in' but the Agent doesn't set the to anything (remain unknown).

I can find only one instance of Agent created, that is on appserver start. So the AgentPool is actually one agent per appserver. Am I correct? This means all tasks on the appserver are executed on the same agent, that's why the id is always 0. Depends on directory configs.

#17 - 01/08/2024 05:16 AM - Galya B

Constantin Asofiei wrote:

- for SOAP, make sure 'soap:address' is set to '/wsa/wsafwd' in `fwd.wsdl`. After that, run `ant jar`

There are two fwd.wsdl files, where soap:address is full url. Where do I need to replace soap:address with the path?

appsrv/soap/proxygen/fwd.wsdl:

```
<wsdl:service name="fwdService">
  <wsdl:port name="persistentObj" binding="tns:persistentObj">
<wsdl:documentation></wsdl:documentation>
  <soap:address location="http://localhost:9100/soap"/>
  </wsdl:port>
  <wsdl:port name="fwdObj" binding="tns:fwdObj">
<wsdl:documentation></wsdl:documentation>
  <soap:address location="http://localhost:9100/soap"/>
  </wsdl:port>
</wsdl:service>
```

appsrv/config/fwd-pasoe/webapps/goldencode/WEB-INF/adapters/soap/goldencode/fwd.wsdl:

```
<wsdl:service name="fwdService">
  <wsdl:port name="persistentObj" binding="tns:persistentObj">
<wsdl:documentation></wsdl:documentation>
  <soap:address location="http://DESKTOP-EIOD5PQ:8080/goldencode/soap"/>
  </wsdl:port>
  <wsdl:port name="fwdObj" binding="tns:fwdObj">
<wsdl:documentation></wsdl:documentation>
  <soap:address location="http://DESKTOP-EIOD5PQ:8080/goldencode/soap"/>
  </wsdl:port>
</wsdl:service>
```

#18 - 01/08/2024 07:09 AM - Constantin Asofiei

Galya B wrote:

Constantin Asofiei wrote:

- for SOAP, make sure 'soap:address' is set to '/wsa/wsafwd' in fwd.wsdl. After that, run ant jar

There are two fwd.wSDL files, where soap:address is full url. Where do I need to replace soap:address with the path?

You need to edit the fwd.wSDL file from the src/ folder, not abl/.

#19 - 01/08/2024 08:02 AM - Galya B

file-cvt-list.txt.soap.rest.webhandler (138 procedures) is a subset of file-cvt-list.txt.rest_and_appserver (988 procs). To run runall.p I need the latter and this is what I've used.

Several tests fail because appsrv/api/table/io_table_handle.p can't find table/help/get_session_dynamic_table.p although it's in the conversion list:

```
appsrv/test/table/test_input_output_table_handle.p| ABL TempTable type : - input/output-table-handle - start
appsrv/test/table/test_input_output_table_handle.p| UNEXPECTED_ERROR: calling procedure isValidTable should not
throw error (** "table/help/get_session_dynamic_table.p" was not found. (293)) #22
appsrv/test/table/test_input_output_table_handle.p| UNEXPECTED_ERROR: on server after calling procedure remain
a valid handle #27
appsrv/test/table/test_input_output_table_handle.p| UNEXPECTED_ERROR: calling procedure isValidTable should not
throw error (** "table/help/get_session_dynamic_table.p" was not found. (293)) #34
appsrv/test/table/test_input_output_table_handle.p| UNEXPECTED_ERROR: calling procedure isValidTable should not
throw error (** "table/help/get_session_dynamic_table.p" was not found. (293)) #46
appsrv/test/table/test_input_output_table_handle.p| UNEXPECTED_ERROR: on server after calling procedure remain
a valid handle #51
appsrv/test/table/test_input_output_table_handle.p| UNEXPECTED_ERROR: calling procedure isValidTable should not
throw error (** "table/help/get_session_dynamic_table.p" was not found. (293)) #58
appsrv/test/table/test_input_output_table_handle.p| ABL TempTable type : - input/output-table-handle - end
appsrv/test/table/test_input_table_handle.p| ABL TempTable type : - input-table-handle - start
appsrv/test/table/test_input_table_handle.p| UNEXPECTED_ERROR: calling procedure isValidTable should not throw
error (** "table/help/get_session_dynamic_table.p" was not found. (293)) #21
appsrv/test/table/test_input_table_handle.p| UNEXPECTED_ERROR: on server after calling procedure remain a valid
handle #26
appsrv/test/table/test_input_table_handle.p| UNEXPECTED_ERROR: calling procedure isValidTable should not throw
error (** "table/help/get_session_dynamic_table.p" was not found. (293)) #33
appsrv/test/table/test_input_table_handle.p| UNEXPECTED_ERROR: calling procedure isValidTable should not throw
error (** "table/help/get_session_dynamic_table.p" was not found. (293)) #45
appsrv/test/table/test_input_table_handle.p| UNEXPECTED_ERROR: on server after calling procedure remain a valid
handle #50
appsrv/test/table/test_input_table_handle.p| UNEXPECTED_ERROR: calling procedure isValidTable should not throw
error (** "table/help/get_session_dynamic_table.p" was not found. (293)) #57
appsrv/test/table/test_input_table_handle.p| ABL TempTable type : - input-table-handle - end
appsrv/test/table/test_output_table_handle.p| ABL TempTable type : - output-table-handle - start
appsrv/test/table/test_output_table_handle.p| UNEXPECTED_ERROR: calling procedure isValidTable should not throw
error (** "table/help/get_session_dynamic_table.p" was not found. (293)) #18
appsrv/test/table/test_output_table_handle.p| UNEXPECTED_ERROR: calling procedure isValidTable should not throw
error (** "table/help/get_session_dynamic_table.p" was not found. (293)) #30
appsrv/test/table/test_output_table_handle.p| UNEXPECTED_ERROR: on server after calling procedure should not r
emain a valid handle #35
appsrv/test/table/test_output_table_handle.p| UNEXPECTED_ERROR: calling procedure isValidTable should not throw
error (** "table/help/get_session_dynamic_table.p" was not found. (293)) #42
appsrv/test/table/test_output_table_handle.p| UNEXPECTED_ERROR: calling procedure isValidTable should not throw
error (** "table/help/get_session_dynamic_table.p" was not found. (293)) #54
appsrv/test/table/test_output_table_handle.p| UNEXPECTED_ERROR: on server after calling procedure should not r
emain a valid handle #59
appsrv/test/table/test_output_table_handle.p| UNEXPECTED_ERROR: calling procedure isValidTable should not throw
error (** "table/help/get_session_dynamic_table.p" was not found. (293)) #66
appsrv/test/table/test_output_table_handle.p| ABL TempTable type : - output-table-handle - end
```

#20 - 01/08/2024 08:20 AM - Galya B

Greg Shah wrote:

- OO instance serialization and transmission to remote side (GES: are we already handling this in [#4658](#)?)

I guess it has to be handled in [#4658](#).

#21 - 01/08/2024 08:28 AM - Greg Shah

- OO instance serialization and transmission to remote side (GES: are we already handling this in [#4658](#)?)

I guess it has to be handled in [#4658](#).

Correct, unless something is absolutely needed for this task.

#22 - 01/08/2024 08:35 AM - Galya B

Marian, can you help with [#6417-19](#). Something seems wrong with the paths to procedures for appsrv/runall.p.

#23 - 01/08/2024 09:55 AM - Galya B

agentId, sessionId, threadId: these are 'plugged in' but the Agent doesn't set the to anything (remain unknown).

Values added to sessionId and threadId in PASOE requests. r14913.

#24 - 01/09/2024 10:49 AM - Galya B

r14914 Fixes legacy appserver throw / return error.

Related tests appsrv/test/test_throw_error.p & appsrv/test/test_return_error.p.

appsrv/test/test_throw_error.p has an issue that needs to be fixed on merge: substitute({&msg_attr_test_value}, 'return-value', 'throwError') to be replaced by substitute({&msg_attr_test_value}, 'return-value', 'returnValue') as defined in appsrv/api/throw_error.p.

r14915 Handles legacy appserver stop error. Related test appsrv/test/test_raise_stop.p.

#25 - 01/09/2024 10:50 AM - Galya B

- % Done changed from 0 to 70

#26 - 01/09/2024 10:53 AM - Galya B

Constantin Asofiei wrote:

There is also a file-cvt-list.txt.rest_and_appserver, to run the plain appserver client tests, but I haven't run these yet. Maybe you can give it a go and post the details here?

Does noone run the tests regularly?

#27 - 01/09/2024 11:31 AM - Greg Shah

Until we are done with our work with [#6853](#), we only run specific sets of tests manually to check functionality. Once [#6853](#) is done, this will be done nightly at a minimum.

#28 - 01/11/2024 09:50 AM - Galya B

What is this port 8980? Something is off, but I can't tell what.

```
[24/01/11@16:48:55.347+0200] P-149379 T-000001 1 4GL -- (Procedure: '../..//appsrv/soap/test/4glws/test_raise_s
top.p') Error loading WSDL document http://localhost:8980/wsa/wsafwd/wsdl/?targetURI=urn:tempuri-org : WSDLE
xception: faultCode=OTHER_ERROR: Unable to resolve imported document at 'http://localhost:8980/wsa/wsafwd/wsdl
/?targetURI=urn:tempuri-org': java.net.ConnectException: Connection refused (Connection refused). (11748)
```

Otherwise <https://localhost:9443/rest/goldencodeService/pipe/char/mihai> works.

P.S. all configs are according to [#6417-8](#).

#29 - 01/12/2024 07:05 AM - Constantin Asofiei

Galya B wrote:

What is this port 8980? Something is off, but I can't tell what.

These files need to be changed manually to match the URL for the WSDL file (9443 instead of 8980):

```
./appsrv/soap/test/4glws/connect.i:9:hWebService:connect ('-WSDL http://localhost:8980/wsa/wsafwd/wsdl/?targetURI=urn:tempuri-org -Port {&portName}Obj').
./appsrv/soap/test/4glws/connect.i:12:hWebService:connect ('-WSDL http://localhost:8980/wsa/wsafwd/wsdl/?targetURI=urn:tempuri-org -Port fwdObj').
```

I missed this, is this when running something from SoapUI?

#30 - 01/12/2024 07:07 AM - Galya B

appsrv/soap/test/4glws.connect.i has the port 8980 hardcoded and soap tests ran from testcases are looking for the resources there.

#31 - 01/12/2024 07:09 AM - Constantin Asofiei

Galya B wrote:

appsrv/soap/test/4glws.connect.i has the port 8980 hardcoded and soap tests ran from testcases are looking for the resources there.

Are you running the SOAP tests from FWD Client (and using the FWD web service component), via SoapUI, or both?

#32 - 01/12/2024 07:54 AM - Galya B

All tests fail on all clients with different errors.

The FWD client shows:

```
(Procedure: '../..appsrv/soap/test/4glws/test_return_error.p') Error loading WSDL document http://localhost:8980/wsa/wsafwd/wsdl/?targetURI=urn:tempuri-org : WSDLException: faultCode=OTHER_ERROR: Unable to resolve imported document at 'http://localhost:8980/wsa/wsafwd/wsdl/?targetURI=urn:tempuri-org': java.net.ConnectException: Connection refused (Connection refused). (11748)
```

I changed the port in appsrv/soap/test/4glws.connect.i to 9443 and now it fails with :

```
[24/01/12@14:41:56.648+0200] P-100032 T-000001 1 4GL -- (Procedure: '../..appsrv/soap/test/4glws/test_return_error.p') Error loading WSDL document http://localhost:9443/wsa/wsafwd/wsdl/?targetURI=urn:tempuri-org : WSDLException: faultCode=OTHER_ERROR: Unable to resolve imported document at 'http://localhost:9443/wsa/wsafwd/wsdl/?targetURI=urn:tempuri-org': java.net.SocketException: Unexpected end of file from server. (11748)
```

corresponding to server error:

```
24/01/12 14:45:11.632+0200 | SEVERE | com.goldencode.p2j.util.ErrorManager | ThreadName:Conversation [00000041:bogus], Session:00000065, Thread:00000074, User:bogus | Invalid or inappropriate server handle specified for RUN fwdObj ... ON SERVER statement. (5453)
```

SoapUI shows NPE for many tests:

```
-> [Character Match] XPathContains assertion failed for path [declare namespace ns="urn:tempuri-org:fwd" //ns:table_outResponse//ns:ttPrimitive//ns:ttPrimitiveRow//ns:dataChar] : NullPointerException:Cannot invoke "String.length()" because "s" is null
-> [Date Match] XPathContains assertion failed for path [declare namespace ns="urn:tempuri-org:fwd" //ns:table_outResponse//ns:ttPrimitive//ns:ttPrimitiveRow//ns:dataDate] : NullPointerException:Cannot invoke "String.length()" because "s" is null
```

corresponding to server error:

```
24/01/12 14:38:06.841+0200 | SEVERE | com.goldencode.p2j.rest.RestHandler | ThreadName:REST worker #0, Session:00000004, Thread:00000006, User:appserver_agent | Could not execute //dataset/get_current_response_info
java.lang.NullPointerException
    at com.goldencode.p2j.util.AppServerManager.invoke(AppServerManager.java:1292)
    at sun.reflect.GeneratedMethodAccessor7.invoke(Unknown Source)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at com.goldencode.p2j.net.LocalStaticRedirector.invokeCore(LocalStaticRedirector.java:167)
    at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:144)
    at com.sun.proxy.$Proxy28.invoke(Unknown Source)
    at com.goldencode.p2j.util.AppServerHelper.invoke(AppServerHelper.java:2499)
    at com.goldencode.p2j.rest.ProgRestService.invoke(ProgRestService.java:196)
    at com.goldencode.p2j.rest.RestHandler.invoke(RestHandler.java:404)
    at com.goldencode.p2j.rest.RestHandler.lambda$handle$3(RestHandler.java:580)
    at com.goldencode.p2j.main.AppServerConnectionPool.lambda$dispatch$1(AppServerConnectionPool.java:248)
    at com.goldencode.p2j.main.LegacyServiceWorker.run(LegacyServiceWorker.java:305)
    at com.goldencode.p2j.main.AppServerConnectionPool.lambda$doInitialize$2(AppServerConnectionPool.java:479)
    at java.lang.Thread.run(Thread.java:750)
24/01/12 14:38:06.842+0200 | WARNING | org.eclipse.jetty.server.HttpChannel | ThreadName:qtp1394842015-53 | handleException /rest/goldencodeService//dataset/get_current_response_info java.io.IOException: java.lang.NullPointerException
```

#33 - 01/12/2024 08:22 AM - Constantin Asofiei

Galya, <http://localhost:9443/wsa/wsafwd/wsdl/?targetURI=urn:tempuri-org> needs to use https and also the full URL I think needs to be this: <https://localhost:9443/ws/wsa/wsafwd/wsdl/?targetURI=urn:tempuri-org> (note the /ws/). This should work from the browser directly, too, to get the WSDL file.

Otherwise:

1. focus on the 4GL tests which run SOAP, not SoapUI. Document which tests fail.
2. in SoapUI, there are failures, but not all needs to be addressed in this task; please focus on the Exceptions and Info test suites

#34 - 01/12/2024 08:30 AM - Galya B

Constantin Asofiei wrote:

Galya, <http://localhost:9443/wsa/wsafwd/wsdl/?targetURI=urn:tempuri-org> needs to use https and also the full URL I think needs to be this: <https://localhost:9443/ws/wsa/wsafwd/wsdl/?targetURI=urn:tempuri-org> (note the /ws/). This should work from the browser directly, too, to get the WSDL file.

Is this supposed to be changed in connect.i? It's hard-coded. Anyways, I'll try to figure out the magic of FWD test configurations and paths.

1. focus on the 4GL tests which run SOAP, not SoapUI. Document which tests fail.

I don't mix SOAP and SoapUI :) But I will probably need to run REST tests as well, because I have to figure out how to set clientType for the different supported types of requests (part of the requirements of the task).

#35 - 01/12/2024 08:32 AM - Constantin Asofiei

Galya B wrote:

Constantin Asofiei wrote:

Galya, <http://localhost:9443/wsa/wsafwd/wsdl/?targetURI=urn:tempuri-org> needs to use https and also the full URL I think needs to be this: <https://localhost:9443/ws/wsa/wsafwd/wsdl/?targetURI=urn:tempuri-org> (note the /ws/). This should work from the browser directly, too, to get the WSDL file.

Is this supposed to be changed in connect.i? It's hard-coded. Anyways, I'll try to figure out the magic of FWD test configurations and paths.

Yes, in connect.i.

#36 - 01/12/2024 08:50 AM - Galya B

r14916 Adds conditions for unknown CURRENT-RESPONSE-INFO / CURRENT-REQUEST-INFO.

#37 - 01/12/2024 09:03 AM - Galya B

Constantin Asofiei wrote:

Galya B wrote:

Constantin Asofiei wrote:

Galya, <http://localhost:9443/wsa/wsafwd/wsd/?targetURI=urn:tempuri-org> needs to use https and also the full URL I think needs to be this: <https://localhost:9443/ws/wsa/wsafwd/wsd/?targetURI=urn:tempuri-org> (note the /ws/). This should work from the browser directly, too, to get the WSDL file.

Is this supposed to be changed in connect.i? It's hard-coded. Anyways, I'll try to figure out the magic of FWD test configurations and paths.

Yes, in connect.i.

Lovely, this is the 3rd type of exception, a new one:

```
[24/01/12@16:01:42.597+0200] P-120205 T-000001 1 4GL -- (Procedure: '../..../appsrv/soap/test/4glws/test_raise_s
top.p') Error loading WSDL document https://localhost:9443/ws/wsa/wsafwd/wsd/?targetURI=urn:tempuri-org : W
SDLException: faultCode=OTHER_ERROR: Unable to resolve imported document at 'https://localhost:9443/ws/wsa/wsa
fwd/wsd/?targetURI=urn:tempuri-org': javax.net.ssl.SSLHandshakeException: PKIX path building failed: sun.sec
urity.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested targ
et. (11748)
```

What certificate?

#38 - 01/12/2024 09:06 AM - Galya B

The batch client I run points to:

```
<security>
  <truststore filename="../../server/srv-certs.store" />
  <truststore alias="standard" />
</security>
```

And it has the truststore. All copied from the other clients.

Isn't it enough?

#39 - 01/12/2024 09:09 AM - Constantin Asofiei

No, this is about the HTTPS connection. I'm looking at it now.

#40 - 01/12/2024 09:26 AM - Galya B

Are you sure it should be https?

#41 - 01/12/2024 09:28 AM - Constantin Asofiei

Galya B wrote:

Are you sure it should be https?

Yes.

#42 - 01/12/2024 09:41 AM - Galya B

Galya B wrote:

- argument translation to remote definition type I think is done the same as the CALL statement does it...

What does that mean?

Which types and what CALL?

#43 - 01/15/2024 05:19 AM - Constantin Asofiei

While in deploy/client-chui, I've ran this and I don't get the certificate issue:

```
./client-swing.sh client:cmd-line-option:startup-procedure=appsrv/test/test_raise_stop.p
```

What do you use to run the tests?

#44 - 01/15/2024 05:47 AM - Galya B

appsrv/test/test_raise_stop.p is the legacy appserver test. Try:

```
appsrv/soap/test/4glws/test_return_error.p
appsrv/soap/test/4glws/test_raise_stop.p
appsrv/soap/test/4glws/test_throw_error.p
appsrv/soap/test/4glws/test_return_value.p
```

#45 - 01/15/2024 05:57 AM - Constantin Asofiei

Thanks. There are a couple of issues:

1. <https://localhost:9443/ws/wsa/wsafwd/wsd/!/?targetURI=urn:tempuri-org> does not work and <https://localhost:9443/ws/wsa/wsafwd/wsd/!/?targetURI=urn:tempuri-org> works - you can use the browser directly, this needs to retrieve the WSDL document. Note the missing / before the ? in the case where it works.
2. even with the WSDL URL 'fixed', then this does not discover any services.

This can be used to run it while in deploy/client-swing:

```
./client-swing.sh client:cmd-line-option:startup-procedure=appsrv/soap/test/4glws/test_raise_stop.p net:http_c
lient:disable_ssl_certificate_validation=true ssl-socket:truststore:location=../server/srv-certs.store ssl-soc
ket:truststore:password=[Ugpg637QuJAC\$\*lep95gqZ-Au\`k7nWSiW15j
```

Please look into both issues.

#46 - 01/15/2024 07:11 AM - Marian Edu

Galya B wrote:

Marian, can you help with [#6417-19](#). Something seems wrong with the paths to procedures for appsrv/runall.p.

Gayla, I've somehow missed this... the file was added by Constantin, most probably to run all tests at once, not sure what is wrong with it, do you want me to look at it or problem was already solved by now?

#47 - 01/15/2024 07:32 AM - Marian Edu

Constantin Asofiei wrote:

Galya B wrote:

Constantin Asofiei wrote:

Galya, <http://localhost:9443/wsa/wsafwd/wsdl/?targetURI=urn:tempuri-org> needs to use https and also the full URL I think needs to be this: <https://localhost:9443/ws/wsa/wsafwd/wsdl/?targetURI=urn:tempuri-org> (note the /ws/). This should work from the browser directly, too, to get the WSDL file.

Is this supposed to be changed in connect.i? It's hard-coded. Anyways, I'll try to figure out the magic of FWD test configurations and paths.

Yes, in connect.i.

For the time being this is hardcoded in this include file, this is at least easier to change in only one place instead of all tests that are making an appsrv connection :)

We will see if/how we can parametrise the tests in order to override such hardcoded values - something that should be supported by both ABLUnit and the FWD implementation of it.

For SoapUI tests everything is based on an `endpoint` so the URL there is also hardcoded - we have tests for both classical and pasoe, as far as I remember the easiest way to run the tests was to update the `service endpoint` and `service path` custom properties on the project... sadly this was only done like that for the REST services while the SOAP projects seems to have the hardcoded URL's :(

#48 - 01/15/2024 08:37 AM - Galya B

Constantin Asofiei wrote:

Thanks. There are a couple of issues:

1. `https://localhost:9443/ws/wsa/wsafwd/wsd/!targetURI=urn:tempuri-org` does not work and `https://localhost:9443/ws/wsa/wsafwd/wsd/?targetURI=urn:tempuri-org` works - you can use the browser directly, this needs to retrieve the WSDL document. Note the missing / before the ? in the case where it works.
2. even with the WSDL URL 'fixed', then this does not discover any services.

This can be used to run it while in deploy/client-swing:

[...]

Please look into both issues.

What do you mean by "not discover any services"? I'm still having a certificate issue:

```
Retrieving document at 'https://localhost:9443/ws/wsa/wsafwd/wsd/?targetURI=urn:tempuri-org'.  
Error loading WSDL document https://localhost:9443/ws/wsa/wsafwd/wsd/?targetURI=urn:tempuri-org : WSDLExcept  
ion: faultCode=OTHER_ERROR: Unable to resolve imported document at 'https://localhost:9443/ws/wsa/wsafwd/wsd/?  
targetURI=urn:tempuri-org'.: javax.net.ssl.SSLHandshakeException: PKIX path validation failed: java.security.c  
ert.CertPathValidatorException: Path does not chain with any of the trust anchors. (11748)
```

The relevant section of the wiki is in TBD: [Web Service SSL Connections](#).

Roger got the same issue in #5024-98, but I can't find an explanation how it got resolved.

The server .der certificate is imported in the JVM's cacert. I'm not sure about the appserver account certificates, if they need to be fixed somehow somewhere.

Otherwise the document opens fine in the browser.

#49 - 01/15/2024 08:39 AM - Constantin Asofiei

Galya, the point here is to use `ssl-socket:truststore` to refer to `srv-certs.store` with the FWD server's certificate. This works with the `bzr xfer` testcases.

#50 - 01/15/2024 08:40 AM - Constantin Asofiei

Galya B wrote:

Otherwise the document opens fine in the browser.

Does it open with both wsdl/?... and wsdl?... cases?

#51 - 01/15/2024 08:42 AM - Galya B

Constantin Asofiei wrote:

Galya B wrote:

Otherwise the document opens fine in the browser.

Does it open with both wsdl/?... and wsdl?... cases?

After the fix, it does, without the extra /.

Galya, the point here is to use ssl-socket:truststore to refer to srv-certs.store with the FWD server's certificate. This works with the bzr xfer testcases.

I do use it, that's configured in client.xml:

```
<security>
  <truststore filename="../server/srv-certs.store" />
  <truststore alias="standard" />
</security>
<access>
  <password truststore="Ugpg637QuJAC$*lep95gqZ-Au`k7nWSiW15j" />
</access>
```

#52 - 01/15/2024 08:45 AM - Galya B

Constantin, so you don't get the ssl issue in the output file or in the clientlog <cmd-line-option clientlog="file.log" />?

#53 - 01/15/2024 08:47 AM - Constantin Asofiei

Galya B wrote:

Galya, the point here is to use ssl-socket:truststore to refer to srv-certs.store with the FWD server's certificate. This works with the bzs xfer testcases.

I do use it, that's configured in client.xml:
[...]

No, that's something else. Please test using the command I mentioned in [#6417-45](#)

#54 - 01/15/2024 09:04 AM - Galya B

Constantin Asofiei wrote:

Galya B wrote:

Galya, the point here is to use ssl-socket:truststore to refer to srv-certs.store with the FWD server's certificate. This works with the bzs xfer testcases.

I do use it, that's configured in client.xml:
[...]

No, that's something else. Please test using the command I mentioned in [#6417-45](#)

net:http_client:disable_ssl_certificate_validation=true hides the ssl problem indeed.

Then Port servicePort = getImplicitSoapPort(service.getPorts().values()); is null. The port called fwdObj, isn't it supposed to be defined in directory.xml?

#55 - 01/15/2024 09:08 AM - Constantin Asofiei

I think this code should be executed, as the -Port from connect is set:

```
if (options.getPortName() != null)
{
    servicePort = service.getPort(options.getPortName());
}
```

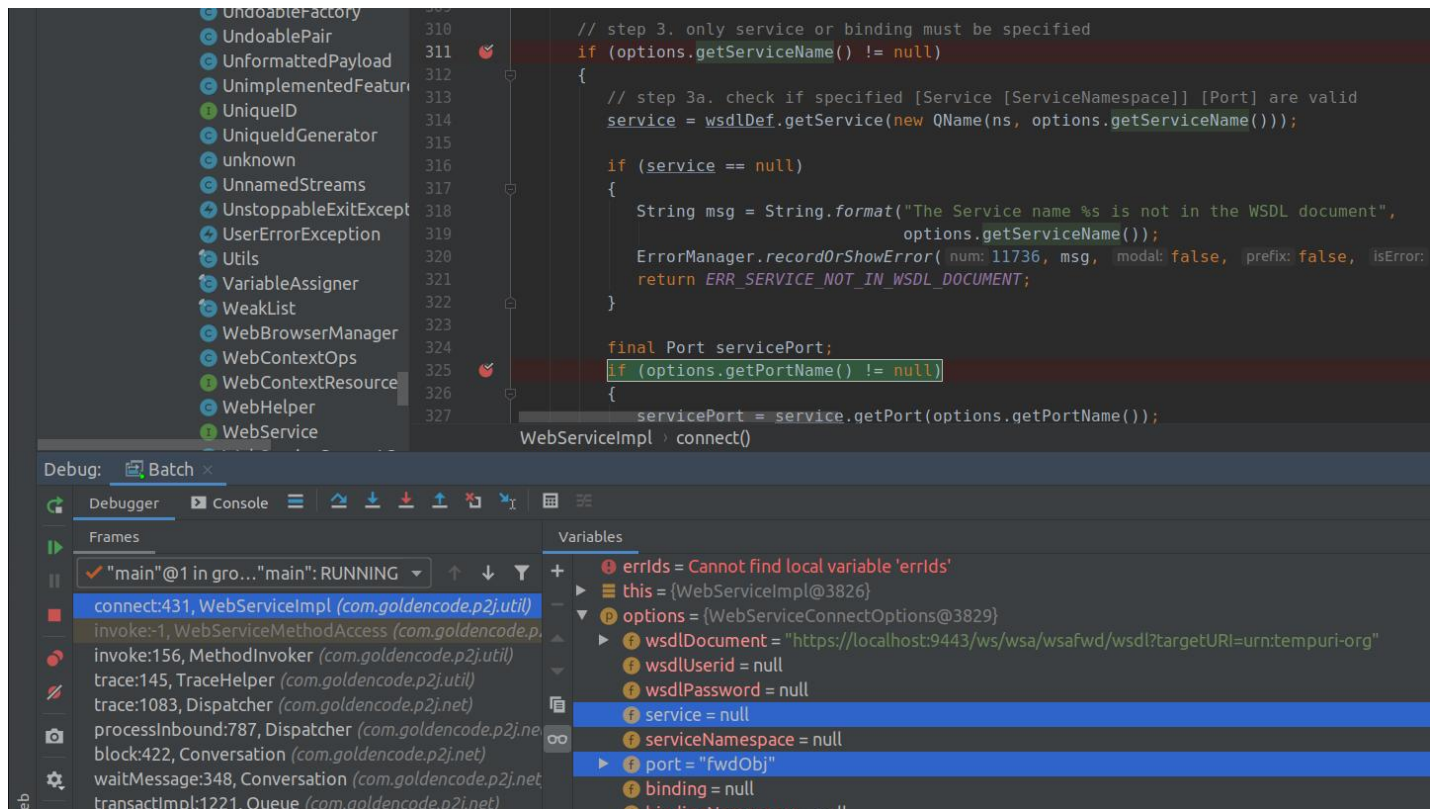
#56 - 01/15/2024 09:16 AM - Galya B

- File no-service-name.png added

Constantin Asofiei wrote:

I think this code should be executed, as the -Port from connect is set:
[...]

It's not.



getServiceName returns service which is null, so the port is fwdObj.

#57 - 01/15/2024 09:17 AM - Galya B

This is executed instead:

```
Port servicePort = getImplicitSoapPort (service.getPorts ().values ());
if (servicePort == null)
{
    noServicesError ();
    return ERR_NO_SERVICE_ERROR;
}
```

#58 - 01/15/2024 09:19 AM - Galya B

P.S. The screenshot can be slightly confusing. Actually the highlighted breakpoint is not hit. Instead the one on line 431 is.

#59 - 01/15/2024 09:21 AM - Constantin Asofiei

Please look into how `getImplicitSoapPort` calculates the implicit port and also compare `fwd.wsdl` from `src/` with `./appsrv/soap/proxygen/fwd.wsdl`, if there are any difference regarding `fwdObj` port.

#60 - 01/15/2024 10:02 AM - Galya B

The problem is that `WebServiceImpl.connect` expects explicit service name or binding name, while there is only a port in the connect options. In this case the code goes for the first service in the `.wsdl` and expects it to have only one port, otherwise an exception is thrown. In the `wsdl` there is one service with two ports, so it doesn't work.

There are a few ways to fix it:

1. in `connect.i` using a service name in the params;
2. in `WebServiceImpl.connect` read the port option, even when no service is explicitly requested, like so:

```
=== modified file 'src/com/goldencode/p2j/util/WebServiceImpl.java'
--- src/com/goldencode/p2j/util/WebServiceImpl.java      2023-05-19 19:46:27 +0000
+++ src/com/goldencode/p2j/util/WebServiceImpl.java      2024-01-15 14:53:02 +0000
@@ -427,7 +427,9 @@
     }
}
```

```
    service = (Service) services.values ().iterator ().next ();
-    Port servicePort = getImplicitSoapPort (service.getPorts ().values ());
+
+    Port servicePort = options.getPortName () != null ? service.getPort (options.getPortName ()) :
+    getImplicitSoapPort (service.getPorts ().values ());
    if (servicePort == null)
    {
        noServicesError ();
    }
}
```

#61 - 01/15/2024 10:20 AM - Galya B

Marian Edu wrote:

Galya B wrote:

Marian, can you help with [#6417-19](#). Something seems wrong with the paths to procedures for appsrv/runall.p.

Gayla, I've somehow missed this... the file was added by Constantin, most probably to run all tests at once, not sure what is wrong with it, do you want me to look at it or problem was already solved by now?

I'm trying to run as much as possible from file-cvt-list.txt.rest_and_appserver. The first issue with missing procedures in the list was described in [#6417-19](#) and I fixed it by adding tests/table/help/get_session_dynamic_table.p.

Now there is another one: (Procedure: '.././appsrv/test/pipe/test_pipe_raw.p') ** "appsrv/test/pipe/test_pipe.i" was not found. (293). I tried to convert appsrv/test/pipe/test_pipe.i (although not sure if .i files are supposed to be converted with convert.list), but it doesn't compile afterwards. Do you know how to run the test appsrv/test/pipe/test_pipe_raw.p?

#62 - 01/15/2024 11:11 AM - Galya B

Something is still off.

ControlFlowOps is throwing errors:

```
if (explicit && h.isUnknown())
{
    String err = null;
    int num = -1;
    final String msg =
        "Could not evaluate procedure handle expression while trying to execute %s";
    err = String.format(msg, name);
    num = 2125;

    ErrorManager.recordOrThrowError(num, err, false);
    return false;
}
```


#63 - 01/16/2024 07:12 AM - Marian Edu

Galya B wrote:

Now there is another one: (Procedure: '././appsrv/test/pipe/test_pipe_raw.p') ** "appsrv/test/pipe/test_pipe.i" was not found. (293). I tried to convert appsrv/test/pipe/test_pipe.i (although not sure if .i files are supposed to be converted with convert.list), but it doesn't compile afterwards. Do you know how to run the test appsrv/test/pipe/test_pipe_raw.p?

I'm not sure what the issue could be here, the include file is there and the procedure compiles just fine in 4GL - only thing that is needed is to have the project's root in propath, you definitively don't need to add the include file to the convert list.

#64 - 01/16/2024 07:14 AM - Galya B

Marian Edu wrote:

Galya B wrote:

Now there is another one: (Procedure: '././appsrv/test/pipe/test_pipe_raw.p') ** "appsrv/test/pipe/test_pipe.i" was not found. (293). I tried to convert appsrv/test/pipe/test_pipe.i (although not sure if .i files are supposed to be converted with convert.list), but it doesn't compile afterwards. Do you know how to run the test appsrv/test/pipe/test_pipe_raw.p?

I'm not sure what the issue could be here, the include file is there and the procedure compiles just fine in 4GL - only thing that is needed is to have the project's root in propath, you definitively don't need to add the include file to the convert list.

The procedure compiles fine, when .i is not in the convert list. But when it's executed, then I get the error (Procedure: '././appsrv/test/pipe/test_pipe_raw.p') ** "appsrv/test/pipe/test_pipe.i" was not found.

#65 - 01/16/2024 07:18 AM - Galya B

It probably is the propath then. I have only:

```
<node class="container" name="standard">
  <node class="container" name="appservers">
    <node class="appserver" name="app_server">
      <node-attribute name="propath" value=".:appsrv/api:"/>
```

appsrv/test is not in. Do you think I miss something else?

#66 - 01/16/2024 07:34 AM - Marian Edu

Galya B wrote:

It probably is the propath then. I have only:
[...]

appsrv/test is not in. Do you think I miss something else?

This is for the procedures that are to be ran from the application server - we deploy on the appsrv only the content of appsrv/api, the file you're referencing is only a test that we run from the client that will call out to the appsrv. When you convert the test procedures - under appsrv/test you only need the `testcases` root in `propath`. The path of the test procedure you're converting looks a bit odd (`../../appsrv/test/pipe/test_pipe_raw.p`)... this is relative to some path in deploy probably and propath only contains "." for the client so the current working folder is somehow important :(

#67 - 01/16/2024 07:46 AM - Galya B

Marian Edu wrote:

The path of the test procedure you're converting looks a bit odd (`../../appsrv/test/pipe/test_pipe_raw.p`)... this is relative to some path in deploy probably and propath only contains "." for the client so the current working folder is somehow important :(

The path in the test procedure itself (appsrv/test/pipe/test_pipe_raw.p) is not odd:

```
{appsrv/test/pipe/test_pipe.i 'pipe_raw.p' 'Mihai' 'character' true 'n/a'}
```

I'm executing it from deploy/client-chui, where the client.xml config is: startup-procedure="../../appsrv/test/pipe/test_pipe_raw.p". Is this not supposed to work?

#68 - 01/16/2024 08:14 AM - Galya B

- File 01-invokeRemoteWithModelImpl.png added

- File 02-invokeImpl.png added

Constantin, I'm debugging ControlFlowOps, but it's too weird. Any hint will be appreciated.

There is execution of invokeImpl with hard-coded null as third argument.

```
ControlFlowOps.java x InvokeConfig.java x
1496     private static void invokeRemoteWithModeImpl(AsyncRequestImpl asyncReq, asyncReq: null
1497         character    name, name: "fwdObj "
1498         handle       portHandle, portHandle: null
1499         handle       remoteHandle, remoteHandle: "1003"
1500         boolean      transactionDistinct, transactionDistinct: false
1501         String       modes, modes: null
1502         Object...    args) args: Object[0]@8272
1503     {
1504         if (portHandle != null)
1505         {
1506             // this is a RUN ... SET ... ON SERVER webService
1507             invokePersistentImpl(asyncReq, remoteHandle, name, persistHandle: null, portHandle, portHandle: null
1508                 transactionDistinct, modes, args); transactionDistinct: false modes: null args: Object[0]@8272
1509         }
1510         else
1511         {
1512             // this is a RUN ... ON SERVER appServer
1513             invokeImpl(asyncReq, remoteHandle, h: null, name, asyncReq: null remoteHandle: "1003" name: "fwdObj "
1514                 function: false, dynamicFunction: false, superCall: false,
1515                 transactionDistinct, modes, args);
1516         }
1517     }
ControlFlowOps invokeRemoteWithModeImpl()
Debug: Server x Batch x
Debugger Console
Frames Variables
"Conversation [0000004..."conversation": RUNNING
invokeImpl:7060, ControlFlowOps (com.goldencode.p2j.util)
invokeRemoteWithModeImpl:1513, ControlFlowOps (com.goldencode.p2j.util)
invokeRemoteWithMode:1471, ControlFlowOps (com.goldencode.p2j.util)
invokeRemoteWithMode:1443, ControlFlowOps (com.goldencode.p2j.util)
invoke:1281, ControlFlowOps (com.goldencode.p2j.util)
run:429, InvokeConfig (com.goldencode.p2j.util)
lambda$execute$3:95, TestThrowError (com.goldencode.testcases.appsvr)
body-1, 191760708 (com.goldencode.testcases.appsvr.soap.test_4glws.T
errids = Cannot find local variable 'errids'
static members of ControlFlowOps
asyncReq = null
name = {character@8270} "fwdObj "
portHandle = null
remoteHandle = {handle@8271} "1003"
transactionDistinct = false
modes = null
args = {Object[0]@8272}
```

But invokeImpl gets into validating h as portHandle in:

```

7044     else
7045     {
7046         if (!validServer(server, h, name)) name: "fwdObj "
7047         {
7048             return unknown.UNKNOWN;
7049         }
7050
7051         ServerImpl s = (ServerImpl) server.get(); s: "ServerImpl{}" server: "1003"
7052         boolean isWebService = s.isWebService(); isWebService: true s: "ServerImpl{}"
7053
7054         if (isWebService && origServer != null && h == null) isWebService: true origServer: "1003" h: null
7055         {
7056             // the target portHandle must always be specified in case of
7057             // RUN port-type ON SERVER h-web-server statements.
7058             // if is missing, the RUN statement is a no-op
7059
7060             return unknown.UNKNOWN;
7061         }
    
```

ControlFlowOps > invokeImpl()

Debug: Server x Batch x

Debugger Console

Frames

- Conversation [0000004... "conversation": RUNNING
- invokeImpl:7060, ControlFlowOps (com.goldencode.p2j.util)
- invokeRemoteWithModelImpl:1513, ControlFlowOps (com.goldencode.p2j.util)
- invokeRemoteWithMode:1471, ControlFlowOps (com.goldencode.p2j.util)
- invokeRemoteWithMode:1443, ControlFlowOps (com.goldencode.p2j.util)
- invoke:1281, ControlFlowOps (com.goldencode.p2j.util)
- run:429, InvokeConfig (com.goldencode.p2j.util)
- lambda\$execute\$3:95, TestThrowError (com.goldencode.testcases.appsrv.soap.test_4glws.TestThrowError)
- body:-1, 191760708 (com.goldencode.testcases.appsrv.soap.test_4glws.TestThrowError)
- body:636, Block (com.goldencode.p2j.util)
- processBody:9195, BlockManager (com.goldencode.p2j.util)
- topLevelBlock:8816, BlockManager (com.goldencode.p2j.util)
- externalProcedure:594, BlockManager (com.goldencode.p2j.util)
- externalProcedure:567, BlockManager (com.goldencode.p2j.util)
- execute:79, TestThrowError (com.goldencode.testcases.appsrv.soap.test_4glws.TestThrowError)
- lambda\$execute\$3:95, TestThrowError (com.goldencode.testcases.appsrv.soap.test_4glws.TestThrowError)

Variables

- errids = Cannot find local variable 'errids'
- static members of ControlFlowOps
- asyncReq = null
- server = {handle@8271} "1003"
- h = null
- name = {character@8270} "fwdObj "
- function = false
- dynamicFunction = false
- superCall = false
- transactionDistinct = false
- modes = null
- args = {Object[0]@8272}
- origServer = {handle@8271} "1003"
- deferred = null
- s = {ServerImpl@8046} "ServerImpl{}"
- isWebService = true

```

// the target portHandle must always be specified in case of
// RUN port-type ON SERVER h-web-server statements.
// if is missing, the RUN statement is a no-op
    
```

#69 - 01/16/2024 08:16 AM - Constantin Asofiei

Galya B wrote:

I'm executing it from deploy/client-chui, where the client.xml config is: startup-procedure="../../appsrv/test/pipe/test_pipe_raw.p". Is this not supposed to work?

The path is: startup-procedure="appsrv/test/pipe/test_pipe_raw.p". Why the two up paths? This name doesn't reference something on disk, but something from name_map.xml.

#70 - 01/16/2024 08:20 AM - Galya B

Constantin Asofiei wrote:

Galya B wrote:

I'm executing it from deploy/client-chui, where the client.xml config is: startup-procedure="../../appsrv/test/pipe/test_pipe_raw.p". Is this not supposed to work?

The path is: startup-procedure="appsrv/test/pipe/test_pipe_raw.p". Why the two up paths? This name doesn't reference something on disk, but something from name_map.xml.

Nice, now I get:

```
(Procedure: 'appsrv/test/pipe/test_pipe_raw.p') ** "appsrv/test/pipe/test_pipe.i" was not found. (293)
```

So the issue is still present.

[#6417-68](#) I asked you about is not related. It happens with any 4glws test.

#71 - 01/16/2024 08:23 AM - Constantin Asofiei

That's because of this statements:

```
copy-lob from file 'appsrv/test/pipe/test_pipe.i' to input_longchar.  
copy-lob from file 'appsrv/test/pipe/test_pipe.i' to input_longchar.
```

This file needs to be made available to the FWD runtime.

Just create the deploy/client/appsrv/test/pipe/test_pipe.i file as a copy of appsrv/test/pipe/test_pipe.i.

#72 - 01/16/2024 08:25 AM - Galya B

Constantin Asofiei wrote:

That's because of this statements:
[...]

This file needs to be made available to the FWD runtime.

Just create the deploy/client/appsrv/test/pipe/test_pipe.i file as a copy of appsrv/test/pipe/test_pipe.i.

Is there no option to add the original location to a client propath?

#73 - 01/16/2024 08:26 AM - Constantin Asofiei

Galya B wrote:

Is there no option to add the original location to a client propath?

You can add the full path for the testcases project to the propath, this will find it.

#74 - 01/16/2024 09:23 AM - Galya B

Constantin Asofiei wrote:

Galya B wrote:

Is there no option to add the original location to a client propath?

You can add the full path for the testcases project to the propath, this will find it.

Is this how it should look:

```
<node class="container" name="standard">
  <node class="container" name="appservers">
    <node class="appserver" name="app_server">
      <node-attribute name="propath" value=".:appsrv/api:/home/gbb/code/testcases:"/>
    </node>
  </node>
  <node class="container" name="runtime">
    <node class="container" name="default">
      <node class="container" name="legacy-system">
        <node class="string" name="propath">
          <node-attribute name="value" value=".:appsrv/api:/home/gbb/code/testcases:"/>
        </node>
      </node>
    </node>
  </node>
</node>
```

Because this doesn't work.

#75 - 01/17/2024 01:32 AM - Galya B

The only way to run `appsrv/test/pipe/test_pipe_raw.p` is as Marian suggested from the root of testcases. I would appreciate if someone confirms that the test can't be run successfully from the deploy dir.

#76 - 01/17/2024 02:40 AM - Marian Edu

Galya B wrote:

The only way to run `appsrv/test/pipe/test_pipe_raw.p` is as Marian suggested from the root of testcases. I would appreciate if someone confirms that the test can't be run successfully from the deploy dir.

I can confirm all paths used in tests and support classes are relative to the project's root so if the `PROPATH` doesn't have the absolute path of this folder but a relative path that will be affected by the current working folder and the procedures/classes/includes (even other support files) referenced won't be resolved.

On our side since we're using PSDOE/Eclipse we've made simple wrapper classes to execute conversion, start server/client and we run those in run/debug configurations (we can override some default parameters if needed) - you can find those in testcases/src-dev. Point is we always run those from the project's root, even the FWD ABLUnit (with a custom tweak applied locally to be able to specify the location of `unit_test.xml` configuration file).

#77 - 01/17/2024 07:14 AM - Galya B

I continue to struggle with the SOAP tests.

Procedure to reproduce the issue (pretty much the connect.i):

```
define variable hWebService as handle no-undo.
define variable hfwdObj as handle no-undo.

create server hWebService.
hWebService:connect('-WSDL https://localhost:9443/ws/wsa/wsafwd/wsd?targetURI=urn:tempuri-org -Service fwdService -Port fwdObj').
run fwdObj set hfwdObj on hWebService no-error.
```

is converted to:

```
public class Soap
{
    private static final InvokeConfig RUN_CALL_SITE_1 = new InvokeConfig().setTarget("fwdObj");

    @LegacySignature(type = Type.VARIABLE, name = "hWebService")
    handle hWebService = TypeFactory.handle();

    @LegacySignature(type = Type.VARIABLE, name = "hfwdObj")
    handle hfwdObj = TypeFactory.handle();

    /**
     * External procedure (converted to Java from the 4GL source code
     * in galya/soap.p).
     */
    @LegacySignature(type = Type.MAIN, name = "galya/soap.p")
    public void execute()
    {
        externalProcedure(Soap.this, new Block((Body) () ->
        {
            ServerFactory.createServer(hWebService);
            hWebService.unwrapConnectable().connect(new character("-WSDL https://localhost:9443/ws/wsa/wsafwd/wsd
```



```
l?targetURI=urn:tempuri-org -Service fwdService -Port fwdObj"));
    RUN_CALL_SITE_1.clone().setProcedureHandle(hfwdObj).setServerHandle(hWebService).run();
  });
}
```

run fwdObj set hfwdObj on hWebService no-error. is converted to code with setProcedureHandle and setServerHandle. On execution in ControlFlowOps.invokeRemoteWithModelImpl this ends up in the second if, although the first if has the comment // this is a RUN ... SET ... ON SERVER webService:

```
if (portHandle != null)
{
  // this is a RUN ... SET ... ON SERVER webService
  invokePersistentImpl(asyncReq, remoteHandle, name, null, portHandle,
    transactionDistinct, modes, args);
}
else
{
  // this is a RUN ... ON SERVER appServer
  invokeImpl(asyncReq, remoteHandle, null, name,
    false, false, false,
    transactionDistinct, modes, args);
}
```

Obviously the portHandle has not been set to the InvokeConfig. How is this supposed to work?

This always ends up hitting:

```
// the target portHandle must always be specified in case of
// RUN port-type ON SERVER h-web-server statements.
// if is missing, the RUN statement is a no-op
```

I can't find a place in the code where InvokeConfig.portHandle is set.

So I just want to know if the code is broken and needs to be fixed, or the tests actually work for someone else (probably with a different connect.i).

#78 - 01/17/2024 07:44 AM - Constantin Asofiei

Galya, this is a regression from back when I rewrote to emit InvokeConfig instead of ControlFlowOps. Please look into rules/convert/control_flow.rules, around this code on line 1234:

```
<action>ref = ref.getImmediateChild(prog.kw_set, null)</action>
<action>runRef = createJavaAst(java.method_call, "setProcedureHandle", runRef)</action>
<action>ref.putAnnotation("peerid", runRef.id)</action>
```

If this has a KW_ON, then instead of setProcedureHandle I think is enough to emit a setPortHandle.

#79 - 01/17/2024 08:44 AM - Galya B

Finally to the next issue Host does not match certificate.:

```
SEVERE | com.goldencode.p2j.net.Dispatcher [Dispatcher.processInbound()] | PID:131472, ThreadName:main, Server
Session:00000068, User:bogus | Unexpected throwable.
java.lang.SecurityException: Host does not match certificate.
    at com.goldencode.p2j.util.LowLevelSocketImpl.createSslSocket(LowLevelSocketImpl.java:400)
```

The service in wsdl has location https://10.39.2.86:9443/ws/wsa/wsafwd. I'm using VPN, so I guess this is the issue. I've added -nohostverify to the connect params, but it does nothing and the client still throws the exception. Is -nohostverify supposed to work?

#80 - 01/17/2024 09:57 AM - Galya B

Without VPN the same bummer, different IP... the local networking one. What do I do, what do I do... Fix the network interfaces or figure out how to use -nohostverify. None sounds good.

#81 - 01/17/2024 09:59 AM - Constantin Asofiei

Doesn't localhost work instead of your IP?

#82 - 01/18/2024 09:30 AM - Galya B

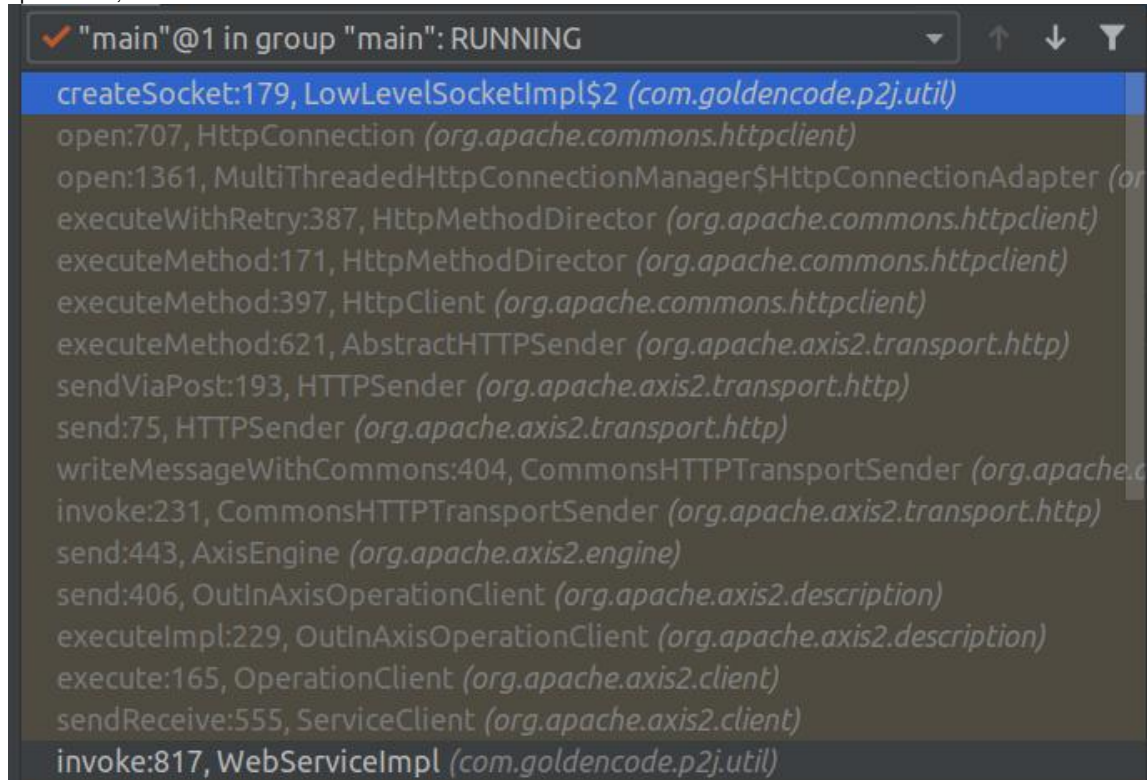
- File losing-P2J_NOHOSTVERIFY.png added

Constantin Asofiei wrote:

Doesn't localhost work instead of your IP?

The IP of eth0 is automatically prepended to the fwdService location in the wsdl, that is available runtime, so I don't seem to have control over it. Since the certificate is for localhost, that's the discrepancy. I need to issue and register a certificate for eth0 IP or make -nohostverify work.

It's official: -nohostverify is broken, because the option LowLevelSocketImpl.P2J_NOHOSTVERIFY passed in to the axis2 ServiceClient never reaches commons' SSLProtocolSocketFactory.createSocket as HttpConnectionParams, where it should be used. The stack has 15 hops purely in apache libs, that's where it's "lost":



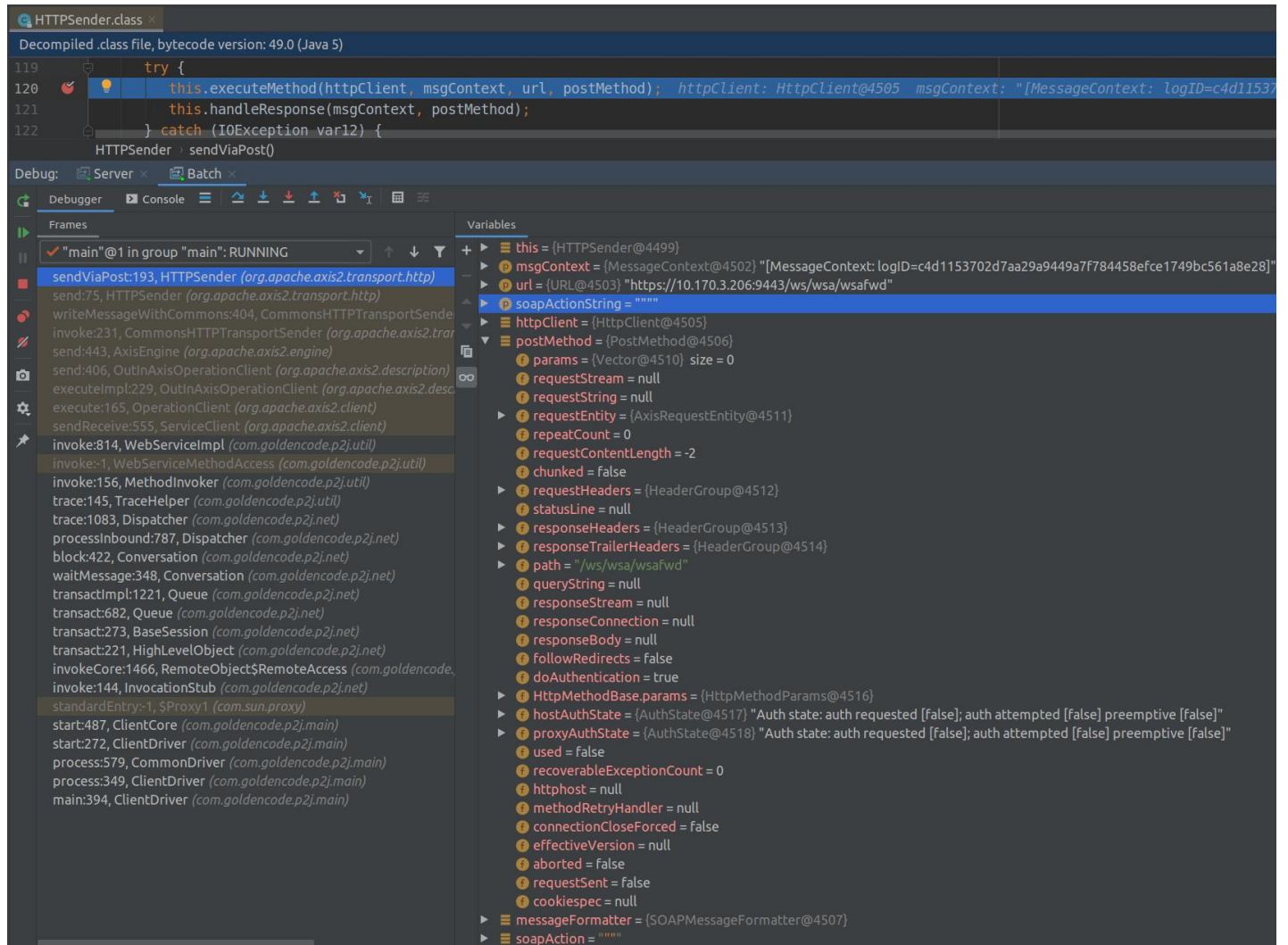
```
✓ "main"@1 in group "main": RUNNING
createSocket:179, LowLevelSocketImpl$2 (com.goldencode.p2j.util)
open:707, HttpConnection (org.apache.commons.httpclient)
open:1361, MultiThreadedHttpConnectionManager$HttpConnectionAdapter (org.apache.commons.httpclient)
executeWithRetry:387, HttpMethodDirector (org.apache.commons.httpclient)
executeMethod:171, HttpMethodDirector (org.apache.commons.httpclient)
executeMethod:397, HttpClient (org.apache.commons.httpclient)
executeMethod:621, AbstractHTTPSender (org.apache.axis2.transport.http)
sendViaPost:193, HTTPSender (org.apache.axis2.transport.http)
send:75, HTTPSender (org.apache.axis2.transport.http)
writeMessageWithCommons:404, CommonsHTTPTransportSender (org.apache.axis2.transport.http)
invoke:231, CommonsHTTPTransportSender (org.apache.axis2.transport.http)
send:443, AxisEngine (org.apache.axis2.engine)
send:406, OutInAxisOperationClient (org.apache.axis2.description)
executeImpl:229, OutInAxisOperationClient (org.apache.axis2.description)
execute:165, OperationClient (org.apache.axis2.client)
sendReceive:555, ServiceClient (org.apache.axis2.client)
invoke:817, WebServiceImpl (com.goldencode.p2j.util)
```

I'll give it the benefit of the doubt. It may have worked in 2015, when implemented, but changes to libraries (although I can't conclude this based on the bsr revisions) broke it. HttpConnectionParams doesn't look as the proper place to put FWD custom options anyways.

I wonder if storing the option in local context will do, but it seems to be on the client's main thread. I need to understand if multiple connections can happen concurrently. Anyways, it should be expected to get fixed.

- File HTTPSender.sendViaPost.png added

The client sends a request to the SOAP endpoint corresponding to the function to be executed on the appserver (e.g. raise_stop), but I don't see any reference to the endpoint in the HTTP request, when debugging. This request returns 302 and I wonder if it hits the endpoint it should at all. On the server SoapHandler.handle is not called, so it further confirms my suspicion. Any ideas?



#84 - 01/22/2024 07:42 AM - Galya B

I'm getting 302 with all SOAP endpoints, when executed from a FWD client (converted procedure). When the endpoints are hit in SoapUI they work as expected. I have no idea what configuration can lead to such outcome. I'm rebasing on latest trunk.

#85 - 01/23/2024 09:27 AM - Galya B

Constantin Asofiei wrote:

There is also a file-cvt-list.txt.rest_and_appserver, to run the plain appserver client tests, but I haven't run these yet. Maybe you can give it a go and post the details here?

I'm out of ideas why the appserver SOAP procedures end up with 302 SOAP fault, when started from the FWD client. I won't be able to run the SOAP tests in FWD client and present the results.

Also it's not clear which set of tests I need to run to complete the task. Only appsv/runall.p is runnable. As I've mentioned before, the file-cvt-list.txt lists include many procedures that require arguments and are not directly runnable.

I still don't understand what this point involves:

argument translation to remote definition type I think is done the same as the CALL statement does it...

Also, I wanted a confirmation I'm understood, when I wrote in [#6417-6](#) that SESSION:LOCAL-VERSION-INFO:OEClientType can't be set to anything but 4GLCLIENT and APPSERVER due to the fact that a FWD client is not limited to one type of connection to the appserver, in contrast with OE.

#86 - 01/23/2024 09:46 AM - Constantin Asofiei

Galya, did you find a way to make -nohostverify work again?

Otherwise, AFAIK FWD server sockets bind to all hosts on the machine; I don't understand why localhost is not working for you in the WSDL URL like <https://localhost:9443/ws/wsa/wsafwd/wsd?targetURI=urn:tempuri-org>. If this works in the browser, it must work from the FWD code, too.

#87 - 01/23/2024 09:51 AM - Galya B

Constantin Asofiei wrote:

Galya, did you find a way to make -nohostverify work again?

SOAP requests seem to be executed synchronously, so I've used the local context to set / get the value, so this is how it works on 6417a.

Otherwise, AFAIK FWD server sockets bind to all hosts on the machine; I don't understand why localhost is not working for you in the WSDL URL like <https://localhost:9443/ws/wsa/wsafwd/wsd?targetURI=urn:tempuri-org>. If this works in the browser, it must work from the FWD code, too.

I get the wsdl fine even from the client, but check the wsdl doc for the url of the endpoints (the location of the service). It uses the eth0 IP.

#88 - 01/23/2024 11:24 AM - Constantin Asofiei

Galya, I have an explanation for the 302 problem. The reason is the trailing '/' char in the request URL; it is not about localhost vs internal IP (at least here). If you use something like <https://localhost:9443/ws/wsa/wsafwd/wsd> in SoapUI, you will see this in the raw log:

[View log...](#)[View log...](#)

```
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "POST /ws/wsa/wsafwd HTTP/1.1[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "Accept-Encoding: gzip,deflate[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "Content-Type: text/xml;charset=UTF-8[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "SOAPAction: ""[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "Content-Length: 277[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "Host: localhost:9443[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "Connection: Keep-Alive[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "User-Agent: Apache-HttpClient/4.5.5 (Java/11.0.21) [\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "<soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/' xmlns:urn='urn:tempuri-org:fdw'>[\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "  <soapenv:Header/>[\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "  <soapenv:Body>[\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "    <urn:pipe_char>[\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "      <urn:inputValue>234</urn:inputValue>[\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "    </urn:pipe_char>[\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "  </soapenv:Body>[\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "</soapenv:Envelope>"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "HTTP/1.1 302 Found[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "Date: Tue, 23 Jan 2024 16:03:32 GMT[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "Location: https://localhost:9443/ws/wsa/wsafwd/[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "Content-Length: 0[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "Server: Jetty(9.4.22.v20191022) [\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "POST /ws/wsa/wsafwd/ HTTP/1.1[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "Accept-Encoding: gzip,deflate[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "Content-Type: text/xml;charset=UTF-8[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "SOAPAction: ""[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "Host: localhost[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "Content-Length: 277[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "Connection: Keep-Alive[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "User-Agent: Apache-HttpClient/4.5.5 (Java/11.0.21) [\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "<soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/' xmlns:urn='urn:tempuri-org:fdw'>[\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "  <soapenv:Header/>[\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "  <soapenv:Body>[\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "    <urn:pipe_char>[\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "      <urn:inputValue>234</urn:inputValue>[\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "    </urn:pipe_char>[\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "  </soapenv:Body>[\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: >> "</soapenv:Envelope>"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "HTTP/1.1 200 OK[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "Date: Tue, 23 Jan 2024 16:03:32 GMT[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "Content-Type: text/xml[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "Transfer-Encoding: chunked[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "Server: Jetty(9.4.22.v20191022) [\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "183[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "<?xml version='1.0' encoding='utf-8'?><SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/' xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xmlns:xsd='http://www.w3.org/2001/XMLSchema'><SOAP-ENV:Body><pipe_charResponse xmlns='urn:tempuri-org:fdw'><result xsi:n
```

```
il="true"/><outputValue>234</outputValue></pipe_charResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "0[\r][\n]"
Tue Jan 23 18:03:32 EET 2024:DEBUG: << "[\r][\n]"
```

I don't think we can make Axis2 to follow the 302 redirect. There is this code being executed when trying to follow redirects:

```
/**
 * Entity enclosing requests cannot be redirected without user intervention
 * according to RFC 2616.
 *
 * @return false.
 *
 * @since 2.0
 */
public boolean getFollowRedirects() {
    return false;
}
```

on this stacktrace:

```
PostMethod(EntityEnclosingMethod).getFollowRedirects() line: 213
HttpMethodDirector.isRedirectNeeded(HttpMethod) line: 819
HttpMethodDirector.executeMethod(HttpMethod) line: 178
HttpClient.executeMethod(HostConfiguration, HttpMethod, HttpContext) line: 397
HTTPSender(AbstractHTTPSender).executeMethod(HttpClient, HttpContext, URL, HttpMethod) line: 621
HTTPSender.sendViaPost(HttpContext, URL, String) line: 193
HTTPSender.send(HttpContext, URL, String) line: 75
CommonsHTTPTransportSender.sendMessageWithCommons(HttpContext, EndpointReference, OMOutputFormat) line: 40
4
CommonsHTTPTransportSender.invoke(HttpContext) line: 231
AxisEngine.send(HttpContext) line: 443
OutInAxisOperationClient.send(HttpContext) line: 406
OutInAxisOperationClient.executeImpl(boolean) line: 229
OutInAxisOperationClient(OperationClient).execute(boolean) line: 165
ServiceClient.sendReceive(QName, OMElement) line: 555
WebServiceImpl.invoke(int, String, String, String) line: 820
WebServiceMethodAccess.invoke(Object, int, Object...) line: not available
```

The redirection is happening on the FWD server-side, somewhere in Jetty. Please debug and find if/how this can be configured, maybe we need to handle implicitly the target with a trailing '/

Otherwise, you can add the trailing '/' in fwd.wsdl at `<soap:address location="/wsa/wsafwd/" />`, and the request gets posted, but there are still errors (something related to namespace, see `SoapHandler.namespaceMatch`). Also, I've used `-WSDL https://localhost:9443/ws/wsa/wsafwd/wsdl?targetURI=urn:tempuri-org -Port fwdObj -nohostverify` for the connection string and this patch:

```
=== modified file 'src/com/goldencode/p2j/soap/SoapHandler.java'
--- old/src/com/goldencode/p2j/soap/SoapHandler.java 2023-05-12 10:05:12 +0000
+++ new/src/com/goldencode/p2j/soap/SoapHandler.java 2024-01-23 16:14:19 +0000
@@ -702,7 +702,7 @@
     return;
 }

-    if (target.endsWith("/wsdl"))
+    if (target.endsWith("/wsdl") || target.endsWith("/wsdl/"))
     {
         String targetURI = request.getParameter("targetURI");
         Document doc = targetURI == null ? null : wsdlDocs.get(targetURI);

=== modified file 'src/com/goldencode/p2j/util/WebServiceImpl.java'
--- old/src/com/goldencode/p2j/util/WebServiceImpl.java 2023-05-19 19:46:27 +0000
+++ new/src/com/goldencode/p2j/util/WebServiceImpl.java 2024-01-23 16:05:43 +0000
@@ -82,6 +82,7 @@
 import org.apache.axis2.context.*;
 import org.apache.axis2.transport.http.*;
 import org.apache.axis2.wsdl.*;
```

```

+import org.apache.commons.httpclient.MultiThreadedHttpConnectionManager;
import org.apache.ws.commons.schema.*;
import javax.wsdl.*;
import javax.wsdl.Binding;
@@ -427,7 +428,7 @@
    }

    service = (Service) services.values().iterator().next();
-    Port servicePort = getImplicitSoapPort(service.getPorts().values());
+    Port servicePort = options.getPortName() != null ? service.getPort(options.getPortName()) : getImplicitSoapPort(service.getPorts().values());
    if (servicePort == null)
    {
        noServicesError();
@@ -805,6 +806,13 @@
        sc.getOptions().setProperty(HTTPConstants.AUTHENTICATE, auth);
    }

+    MultiThreadedHttpConnectionManager httpmgr = new MultiThreadedHttpConnectionManager();
+    httpmgr.getParams().setBooleanParameter(LowLevelSocketImpl.P2J_NOHOSTVERIFY,
+                                           wsd.isNoHostVerify());
+    sc.getServiceContext()
+        .getConfigurationContext()
+        .setProperty(HTTPConstants.MULTITHREAD_HTTP_CONNECTION_MANAGER, httpmgr);
+
    sc.getOptions().setProperty(LowLevelSocketImpl.P2J_NOHOSTVERIFY,
                               wsd.isNoHostVerify());

```

#89 - 01/24/2024 02:15 AM - Galya B

I was thinking about redirect, but I was expecting 301 instead, so didn't dig deeper. In Jetty logs on the server nothing is found related to 301 / 302 / redirect.

Do we expect it to be some kind of misconfiguration in testcases, since prod apps are working fine?

#90 - 01/24/2024 02:16 AM - Constantin Asofiei

Galya B wrote:

Do we expect it to be some kind of misconfiguration in testcases, since prod apps are working fine?

I don't think so. Please try and make both paths (with and without trailing '/' char) handled by SoapHandler - this should fix it.

#91 - 01/24/2024 11:15 AM - Galya B

```
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:S1="urn:soap-fault:details" xmlns:S2="urn:tempuri-org:fwd" xmlns:S3="urn:tempuri-org:persistent" xmlns:prodata="urn:schemas-progress-com:xml-prodata:0001" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:tempuri-org" xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="fwd" targetNamespace="urn:tempuri-org">
```

```
<wsdl:operation name="pipe_char">
  <wsdl:input message="tns:fwd_pipe_char"/>
  <wsdl:output message="tns:fwd_pipe_charResponse"/>
  <wsdl:fault message="tns:FaultDetailMessage" name="fwdFault"/>
</wsdl:operation>
```

pipe_char (called by test_pipe_char.p) is one of the successful procedures, when executed from SoapUI, but not from the FWD client. The operation uses the tns namespace, that is urn:tempuri-org, and it is the namespace of OMElement elRequest (where the variable ns comes from). Later we have a check that should be successful to return a result:

```
if (namespaceMatch(ns, portOp.getInput()) &&
    namespaceMatch(ns, portOp.getOutput()) &&
    namespaceMatch(ns, portOp.getFault("FaultDetailMessage")))
```

In SoapHandler.namespaceMatch the following code msg.getPart(el instanceof Fault ? "FaultDetail" : "parameters").getElementName().getNamespaceURI() returns urn:tempuri-org:fwd, which doesn't match. That is because the actual input, output and fault messages of pipe_char are from other namespaces S1 (urn:soap-fault:details) and S2 (urn:tempuri-org:fwd):

```
<wsdl:message name="FaultDetailMessage">
  <wsdl:part element="S1:FaultDetail" name="FaultDetail"/>
</wsdl:message>
<wsdl:message name="fwd_pipe_char">
  <wsdl:part element="S2:pipe_char" name="parameters"/>
</wsdl:message>
<wsdl:message name="fwd_pipe_charResponse">
  <wsdl:part element="S2:pipe_charResponse" name="parameters"/>
</wsdl:message>
```

Constantin, I have no idea why the check and what can we do about it. Maybe everything used by testcases should be in one namespace?

#92 - 01/25/2024 03:54 AM - Galya B

The SOAP envelop is different.

SoapUI:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>Mihai</urn:inputValue>
    </urn:pipe_char>
  </soapenv:Body>
</soapenv:Envelope>
```

FWD client:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header />
  <soapenv:Body>
    <tns:pipe_char xmlns:tns="urn:tempuri-org">
      <tns:inputValue>Mihai</tns:inputValue>
    </tns:pipe_char>
  </soapenv:Body>
</soapenv:Envelope>
```

#93 - 01/29/2024 11:17 AM - Galya B

The targetNamespace should be used only for the wsdl elements declared without an explicit namespace, but in `WebServiceImpl.invoke` the client prepares the SOAP request by simply adding the targetNamespace to the request and then the server in `SoapHandle.handle` checks if that same target namespace corresponds to the namespaces of input, output of the operation, which they obviously don't according to the wsdl itself.

#94 - 01/30/2024 09:17 AM - Galya B

The root wsdl tag <wsdl:definitions> has one targetNamespace, while the <schema> for the operation has a different one in testcases. The FWD client was building the request body with the targetNamespace from <wsdl:definitions>, so I'm changing it to take the schema one. It seems to work that way.

Someone familiar with how FWD clients are used in prod for accessing soap services needs to validate the behavior in the testing phase.

#95 - 01/31/2024 03:13 AM - Galya B

Help!

Is this statement expected to throw an exception in the client?

```
private static final InvokeConfig RUN_CALL_SITE_15 = new InvokeConfig().setTarget("throw_error").setModes("IO"
);

[..]

silent(() -> RUN_CALL_SITE_15.clone().setInHandle(hfwdObj).run(new integer(9407), soapResult));
```

It's in a silent wrapper, but the external procedure execution still throws client-side. It is a soap request that fails.

Do we need to fix this behavior by informing the client somehow about no-error?

#96 - 01/31/2024 03:16 AM - Constantin Asofiei

Galya, what test are you executing from the 4GL client via soap?

#97 - 01/31/2024 03:17 AM - Galya B

Constantin Asofiei wrote:

Galya, what test are you executing from the 4GL client via soap?

```
<cmd-line-option startup-procedure="appsrv/soap/test/4glws/test_throw_error.p" />
```

```
block-level on error undo, throw.
{common/log_hlp.i}
```

```
define variable outChar as character no-undo.
```

```
run log-it('ABL 4glws throw error procedure : - start').
```

```
{appsrv/soap/test/4glws/connect.i}
```

```
run throw_error in hfwdObj(input 9407, output soapResult) no-error.
assert-err-num(9407,
```

```
substitute({&msg_call_error}, 'throw_error', ''),
{&line-number}).

run log-it('ABL 4glws throw error procedure : - end').

catch e as Progress.Lang.AppError :
  message e:ReturnValue.
end catch.

finally:
{appsrv/soap/test/4glws/disconnect.i}
end finally.
```

#98 - 01/31/2024 03:23 AM - Galya B

I think some legacy 4GL code is converted to SOAP endpoints and is executed from 4GL like WSA, ref in the table [#6417-6](#). Please tell me if there is another one as well, I couldn't find any information in our docs / wikis.

#99 - 01/31/2024 03:26 AM - Galya B

Constantin, you won't be able to run appsrv/soap/test/4glws/test_throw_error.p from 4GL client without the fix for the namespaces in 6417a. I was asking about the expected behavior, because it seems very messy. I wonder if some customers call SOAP from 4GL clients in prod (I've seen some -WSDL connects in one of the customers, but I can't figure out if it's just a test or actually used in prod). I don't want to change core behavior if there are some workarounds working for customers.

#100 - 01/31/2024 03:29 AM - Constantin Asofiei

I'm confused about this test. Galya, I assume you've added the input 9407 argument.

Marian: can you please advise with the above test? I assume error 9407 needs to be thrown by this call, and the catch block does nothing?

#101 - 01/31/2024 03:32 AM - Constantin Asofiei

Galya B wrote:

I think some legacy 4GL code is converted to SOAP endpoints and is executed from 4GL like WSA, ref in the table [#6417-6](#). Please tell me if there is another one as well, I couldn't find any information in our docs / wikis.

What do you mean here? The .wsm file configures which programs are exposed as SOAP.

#102 - 01/31/2024 03:52 AM - Galya B

Constantin Asofiei wrote:

I'm confused about this test. Galya, I assume you've added the input 9407 argument.

Yes, I've added 9407, because the external function `throw_error.p` requires an input param and throws error otherwise for invalid params. I'm also confused about the test.

#103 - 01/31/2024 03:59 AM - Galya B

Constantin Asofiei wrote:

Galya B wrote:

I think some legacy 4GL code is converted to SOAP endpoints and is executed from 4GL like WSA, ref in the table [#6417-6](#). Please tell me if there is another one as well, I couldn't find any information in our docs / wikis.

What do you mean here? The `.wsm` file configures which programs are exposed as SOAP.

`.wsm` configure server-side the endpoints. The client is WSA (Web Services Adapter). Isn't WSA code converted and executed by FWD clients?

#104 - 01/31/2024 04:09 AM - Constantin Asofiei

Galya B wrote:

`.wsm` configure server-side the endpoints. The client is WSA (Web Services Adapter). Isn't WSA code converted and executed by FWD clients?

I think I understand what you mean - when a SOAP request gets executed (regardless if is via SoapUI, another 4GL program, etc), the request is handled by a FWD client - and I think you can name these clients 'WSA'.

#105 - 01/31/2024 04:14 AM - Galya B

Constantin Asofiei wrote:

Galya B wrote:

.wsm configure server-side the endpoints. The client is WSA (Web Services Adapter). Isn't WSA code converted and executed by FWD clients?

I think I understand what you mean - when a SOAP request gets executed (regardless if is via SoapUI, another 4GL program, etc), the request is handled by a FWD client - and I think you can name these clients 'WSA'.

Do you mean the appserver client?

#106 - 01/31/2024 04:14 AM - Marian Edu

Constantin Asofiei wrote:

I'm confused about this test. Galya, I assume you've added the input 9407 argument.

Marian: can you please advise with the above test? I assume error 9407 needs to be thrown by this call, and the catch block does nothing?

Hmm, looks like the `throw_error.p` procedure was changed to take this additional parameter, it was most probably done for REST but then SOAP tests we had before were not updated. The thing I wanted to see there was the result when throwing an `AppError` with `return-value` vs. one with `num-messages`.

The 4GL test needs to be updated to send this new parameter as input - `test_throw_error.p` must send 9470 as input parameter when calling `throw_error` on the server, I will fix that on testcases.

#107 - 01/31/2024 04:16 AM - Marian Edu

Marian Edu wrote:

I will fix that on testcases.

And of course the proxy gen file must be updated and you might probably need the ``wsdl`` file as well.

#108 - 01/31/2024 04:18 AM - Galya B

Constantin, I think we need to fix the FWD client to never throw an exception from a faulty SOAP request but instead leave it to the server, so that no-error takes effect.

#109 - 01/31/2024 04:22 AM - Galya B

Galya B wrote:

Constantin, I think we need to fix the FWD client to never throw an exception from a faulty SOAP request but instead leave it to the server, so that no-error takes effect.

That is in WebServiceImpl.invoke:

```
EntityManager.recordOrThrowError(11506,  
    String.format(msg, operationName, faultMsg), false, true);
```

#110 - 01/31/2024 04:30 AM - Galya B

Marian, can you tell me how AIA and WSA are migrated to FWD? I made an attempt in [#6417-6](#) to describe what I've found in the documentation, but it seems incomplete. These are adapters, AIA from what I read in 4GL doc is front-end lib that allows requests to the appserver. So my assumption is that in Progress there is an actual code written for the client using AIA and WSA.

#111 - 01/31/2024 04:50 AM - Marian Edu

Galya B wrote:

Marian, can you tell me how AIA and WSA are migrated to FWD? I made an attempt in [#6417-6](#) to describe what I've found in the documentation, but it seems incomplete. These are adapters, AIA from what I read in 4GL doc is front-end lib that allows requests to the appserver. So my assumption is that in Progress there is an actual code written for the client using AIA and WSA.

Gayla, I think Constantin was working on FWD implementation - the 4GL client in this case is making a SOAP call and it just happens the SOAP endpoints are running on the AppSrv so it's the WSA involved somewhere in between but this is irrelevant for the client, it can well be any SOAP endpoint. Normally the 4GL client will call in directly on an appsvr but those tests were just one way for Constantin to test the SOAP implementation in FWD.

#112 - 01/31/2024 05:07 AM - Galya B

REST and SOAP endpoints in FWD don't correspond directly to a certain adapter in Progress and these adapters are not converted to code in FWD,

so AIA and WSA are not applicable values for clientType.

#113 - 01/31/2024 06:17 AM - Constantin Asofiei

Galya B wrote:

REST and SOAP endpoints in FWD don't correspond directly to a certain adapter in Progress and these adapters are not converted to code in FWD, so AIA and WSA are not applicable values for clientType.

Exactly. WSA in OpenEdge is part of the middle layer deployed in tomcat - see <https://docs.progress.com/bundle/openedge-application-and-integration-services-117/page/WSA-configuration-and-Web-service-deployment-tools.html>, there is no 'client' equivalent in FWD terms for it.

For AIA, it may be for the 'legacy web', but I don't recall exactly. But, if lets say an appserver agent must identify itself as WSA/AIA for the duration of a request, depending on what kind of request is executes, then we need to implement this. Marian: does this ring a bell?

#114 - 01/31/2024 06:19 AM - Constantin Asofiei

Marian Edu wrote:

The 4GL test needs to be updated to send this new parameter as input - test_throw_error.p must send 9470 as input parameter when calling throw_error on the server, I will fix that on testcases.

Please make the 4GL test which executes the SOAP request more 'explicit', to understand how the remote error propagates to the caller (if applicable).

#115 - 01/31/2024 06:33 AM - Galya B

Constantin Asofiei wrote:

For AIA, it may be for the 'legacy web', but I don't recall exactly. But, if lets say an appserver agent must identify itself as WSA/AIA for the duration of a request, depending on what kind of request is executes, then we need to implement this.

AIA is indeed 'legacy web', I've found a reference to LegacyWebServiceHandler somewhere in the documentation. And in Progress doc I've found AIA has a code (like a library) on the front-end. AIA is to be executed from the front-end.

The issue is that even the request can't be marked as AIA / WSA, because the context can come either from the endpoint conversion, or the client library / sdk / adapter. There is nothing in the endpoint identifying its relationship to one of the original types, and there is nothing in the client.

#116 - 01/31/2024 06:46 AM - Constantin Asofiei

Galya B wrote:

The issue is that even the request can't be marked as AIA / WSA, because the context can come either from the endpoint conversion, or the client library / sdk / adapter. There is nothing in the endpoint identifying its relationship to one of the original types, and there is nothing in the client.

What I mean is that when the appserver agent is executing a request from SoapHandler, that agent can identify 'client-type' as WSA (if this is how 4GL does it). Same for WebServiceHandler, if the request originates from this, then it can report AIA.

#117 - 01/31/2024 08:00 AM - Galya B

In FWD there are:

- LegacyWebServiceHandler is extended by:
 - RestHandler / rest/enabled
 - SoapHandler / soap/enabled
- WebServiceHandler - implementation of OpenEdge.Web.WebHandler that handles OpenEdge.Web.WebRequest / webHandler/enabled.

In OE: <https://community.progress.com/s/article/What-are-the-PASOE-transport>

For Classic UBrowsers, when the client needs to connect through the internet (HTTP/HTTPS), separate and distinct **adapters** **AIA/WSA/REST/Mobile need to be installed** and configured so that the **client** can connect via the AIA adapter to the Appserver for example.

PAS for OpenEdge is a Web server, which means that there is no need to install and configure adapters on a separate Web server. To put it simply: "transports replace adapters":

Transport	URL path
APSV	/apsv
REST	/rest
SOAP	/soap
WEB	/web

Basically, all our endpoints created by the three above-mentioned FWD classes create PASOE endpoints. So we can't mark the requests as coming from an adapter client to a legacy appserver.

AIA is configured in OE with jar in Tomcat and a servlet for the /aia path that calls com.progress.aia.Aia to handle the request: <https://community.progress.com/s/article/P109721> .

<https://docs.progress.com/bundle/openedge-classic-appserver-administration-117/page/AppServer-Internet-Adapter.html>

The AppServer Internet Adapter (AIA) allows ABL clients and Open Clients (.NET and Java™) to access the AppServer or the OpenEdge Adapter for SonicMQ over the Internet.

The AIA is installed and runs as a Java servlet in most any Java container, which provides the Internet access to the AppServer.

So 4GL knows it's AIA when their jar is installed to handle the requests to /aia. **The request itself to the aia endpoint is nothing unique.**

The same with WSA. It's a custom handler mapped to a servlet path /wsa (<https://docs.progress.com/bundle/openedge-webservices-development-117/page/Deploying-a-SOAP-Web-service.html>):

The WSA is a Java servlet running on a Web server or stand-alone Java Servlet Engine (JSE). Situated between the client and AppServer, this Java servlet understands how to pass communications at run time between a Web service client and the AppServer.

WebSpeed is a web server with deployed WebSpeed Messenger (OE code) + logic in SpeedScript executed server-side (similar to .php):

WebSpeed Transaction Server becomes an application server for Web browser clients. The Transaction Server runs the SpeedScript, which generates HTML pages, and returns these Web pages to Web browser clients through the Messenger in response to Post and Get requests from the clients.

The Messenger runs on a Web server [...] and acts as the gateway and translator between Web requests and responses on the Web server side and the corresponding WebSpeed requests and responses on the Transaction Server side.

Again, we don't have that intermediate code that marks normal POST / GET requests as WebSpeed.

If any of those requests originate from OpenEdge.Web.WebHandler, it means the clientType is 4GLCLIENT.

#118 - 01/31/2024 08:14 AM - Greg Shah

We should have full support for WSA/PASOE SOAP and for REST Adapter/PASOE REST. I would expect that we can control any marking for these pretty easily.

We don't have proper websppeed support yet, at least not the "classic" version of websppeed that is implemented as CGI. We do support the webhandler, which is a kind of "do it yourself" web hook. In OE can handle some more complicated web requests which can't otherwise be handled. So I would say that our support for class websppeed and the PASOE WEB transport is not there, other than the webhandler portion (which only works in PASOE? I'm not sure about that). We do intend to add support for websppeed/PASOE WEB in [#6506](#) (at least for the core CGI).

We don't support AIA, which I thought was only the gateway functionality in classic appserver where a web server is used to terminate the SSL connection and it just connects to appserver as a proxy. Right now we have no plans to support this. I thought that AIA doesn't have a PASOE equivalent for this, though perhaps I am mis-remembering.

#119 - 01/31/2024 08:19 AM - Galya B

In FWD the three types of legacy endpoints handlers are simply generated by openapi.openedge.export, and get converted each to a LegacyService with a type (REST, SOAP, WEBHANDLER). Again those types don't correspond to the above adapters or any specific client. In OE the clients of openapi are ABL / Java / .NET. It doesn't have to do with how the endpoints are generated.

#120 - 01/31/2024 08:24 AM - Greg Shah

My point is that if we need to know whether we are in classic REST, PASOE REST, WSA (classic SOAP), PASOE SOAP... we can determine this. These services are configured explicitly and we can mark them as classic or PASOE (if we don't already).

#121 - 01/31/2024 10:00 AM - Galya B

The value of clientType can be applied to requests:

- 4GLCLIENT when an external procedure is running on legacy appserver from 4GL code
- JAVACLIENT when an external procedure is running on legacy appserver from Java code
- WSA when an external procedure is running on legacy appserver after -WSDL connect.
- APPSERVER when a startup procedure is running on legacy appserver
- MULTI-SESSION-AGENT when a startup procedure is running on pasoe appserver
- ? when an endpoint is executed without session (i.e. from SoapUI) (on PASOE only?)

clientType can have two values divided by comma, because only 4GL/Java/.NET are actual "clients", while AIA, WSA, WebSpeed are web adapters and both can co-exist.

AIA is neither REST, nor SOAP. It's one endpoint /aia with OE servlet handling custom HTTP requests generated by 4GL / OE Java / .NET clients. The so called Open Clients Java / .Net use OE classes - com.progress.open4gl.javaproxy.OpenAppObject / Progress.Open4GL.Proxy.OpenAppObject. 4GL clients probably have to use a custom implementation of OpenEdge.Web.WebHandler.

The annotation we have in our testcases openapi.openedge.export is probably used to open OpenEdge procedures to the servlet-adapter. In FWD we create separate endpoints for each procedure, while this is not the case in OE, where it's only one endpoint.

AIA is not FWD's WEBHANDLER. WebServiceHandler is simply implementation of OpenEdge.Web.WebHandler, that is (<https://documentation.progress.com/output/oehttpclient/oe117/index.html>):

Abstract class to use as a basis for building WebSpeed for PASOE request handlers

I don't understand how / if WEBHANDLER is used currently in FWD.

WEBSPEED is not REST or SOAP, it's a script server-side (like php) + custom GET/POST requests.

There is also the RA or REST Management Agent, but it's not in the clientType.

#122 - 01/31/2024 10:12 AM - Galya B

See, the paradox displayed by testcases:

appsrv/soap/test/4glws/test_throw_error.p connects to the SOAP agent with appsrv/soap/test/4glws/connect.i:

```
hWebService:connect ('-WSDL https://localhost:9443/ws/wsa/wsafwd/wsd1/?targetURI=urn:tempuri-org -Service fwdS  
ervice -Port fwdObj').
```

Then `appsrv/soap/test/4glws/test_throw_error.p` runs `throw_error`, that is generated by `appsrv/api/throw_error.p` annotated as REST:

```
@openapi.openedge.export FILE(type="REST", executionMode="external", useReturnValue="false", writeDataSetBeforeImage="false").
```

This 'REST' endpoint ends up in the WSDL. So at this point we can't even determine the correct protocol by the `openapi` annotation.

The only way to mark `clientType` as WSA is to suppose all SOAP connections to legacy appserver are WSA.

#123 - 02/01/2024 02:55 AM - Galya B

I need to know where `openapi.openedge.export` is used as annotation in the original OE? I've found some references in their documentation, but it's not clear when it's used for what purpose and if it's manually or auto generated.

#124 - 02/01/2024 03:04 AM - Constantin Asofiei

Galya B wrote:

I need to know where `openapi.openedge.export` is used as annotation in the original OE? I've found some references in their documentation, but it's not clear when it's used for what purpose and if it's manually or auto generated.

AFAIK 4GL annotations have no meaning at runtime, even the compiler treats them as comments. I don't recall if these are auto generated or hand-written (maybe they can be both?).

For note [#6417-122](#) - you can not rely on the 4GL code being executed (a 4GL procedure can be annotated for both SOAP and REST, there is no restriction). What matters is who calls this code: modify the `invoke` code at the `soap/rest/web` handlers to pass a parameter which specifies a 'client-type override'; this will be passed to the agent and, when the override is set, it will report this client-type.

#125 - 02/01/2024 06:13 AM - Galya B

I've found an example of how the annotation `openapi.openedge.export` is created: <https://community.progress.com/s/question/0D54Q00007gckNxSAI/creating-an-openedge-abl-service-using-openedge-1170>. The example is for the /web transport (WebSpeed service) in PASOE servers. Business entity is created with resource url and crud operations and thus a `.cls` file is generated with annotations including `openapi.openedge.export` (REST). A Data Object service is created by importing the `.cls` and the annotation `progress.service.resource` defines the url path and name of the json object.

Another way to generate REST annotations for `.p` or `.cls` files in Project Explorer (legacy appserver): <https://docs.progress.com/bundle/openedge-developer-studio-olh-117/page/Defining-REST-service-annotations.html>. The OE tool 'REST Expose Editor' lists resources annotated with REST to associate an operation with a verb for a REST resource.

On the other hand SOAP doesn't seem to use the same annotations. <https://docs.progress.com/bundle/openedge-webservices-development-117/page/Web-service-objects-and-the-WSDL-file.html> :

During Web service deployment, the WSA generates a WSDL file that defines one or more Web service objects to describe the Web service.

These Web service objects conform to the Open Client object model.

The legacy appserver WSA adapter is supposed to be used server-side, while clients are Java/.NET open clients. The server and client interfaces are generated with ProxyGen and rely on objects AppObject and SubAppObject.

For PASOE the soap services are deployed with a script deploySOAP.[bat|sh] using .wsm file generated by ProxyGen.

#126 - 02/01/2024 09:56 AM - Galya B

Question: AIA is simply allowing to connect to legacy appserver over http(s). This is mind-blowing. So the question: Is allowing the appserver to be accessed from another host via http enabled by default in FWD? Should server:connect("-URL http//IP/aia") work without any FWD configurations?

#127 - 02/01/2024 10:00 AM - Greg Shah

We don't support AIA and have no plans to do so.

#128 - 02/01/2024 10:04 AM - Galya B

Greg Shah wrote:

We don't support AIA and have no plans to do so.

AIA is enabling server:connect("-URL http//IP/aia") generally speaking and we already have it. We may not have the servlet adapter and Java / .NET clients with their interfaces, but the purpose of AIA is simply allowing internet connections to the legacy appserver, so we have that fulfilled. **My question was if it's hidden behind configuration similar to OE or not and I think it's not.**

<https://docs.progress.com/bundle/openedge-classic-appserver-administration-117/page/AppServer-Internet-Adapter.html>

The AppServer Internet Adapter (AIA) allows ABL clients and Open Clients (.NET and Java™) to access the AppServer or the OpenEdge Adapter for SonicMQ over the Internet.

#129 - 02/01/2024 10:16 AM - Galya B

This is my final conclusion after a lot of research and this is to be implemented:

Value	OE Client Type	FWD
"4GLCLIENT"	ABL Client	an external procedure is running on legacy appserver from 4GL code
"JAVACLIENT"	Java Open Client	an external procedure is running on legacy

		appserver from Java code
".NETCLIENT"	NET Open Client	Not supported
"AIA"	AIA (AppServer Internet Adapter)	an external procedure is running on legacy appserver after -URL connect
"WSA"	WSA (Web Services Adapter)	an external procedure is running on legacy appserver after -WSDL connect
"ESB"	ESB (Adapter for Sonic ESB)	Not supported
"APPSERVER"	AppServer agent	startup procedure is running on legacy appserver
"WEBSPEED"	WebSpeed agent	Not supported
"MULTI-SESSION-AGENT"	Progress Application Server agent	startup procedure is running on PASOE appserver
Unknown value (?)	Other special-purpose clients	Any other client type

#130 - 02/05/2024 04:38 AM - Galya B

I need some clarification on the resource/proxy/classloader thing: SOAPFaultImpl inherits HandleResource and the parent constructor calls ProxyFactory.isProxyClass, checking if ProxyFactory.loader (AsmClassLoader) is the same as SOAPFaultImpl.getClassLoader (AppClassLoader).

First, I don't understand what this is all about.

Then, ProxyFactory.loader is null, because AsmClassLoader hasn't been instantiated, so the client can't send SOAPFaultImpl.

#131 - 02/05/2024 06:52 AM - Constantin Asofiei

Galya B wrote:

I need some clarification on the resource/proxy/classloader thing: SOAPFaultImpl inherits HandleResource and the parent constructor calls ProxyFactory.isProxyClass, checking if ProxyFactory.loader (AsmClassLoader) is the same as SOAPFaultImpl.getClassLoader (AppClassLoader).

You mean this code?

```
// this will ensure the INSTANTIATING-PROCEDURE attribute is setup accordingly and also that
// static resources will be registered with the correct top-level block
if (processResource() && !ProxyFactory.isProxyClass(this.getClass()))
{
    // for proxy cases, explicit registration is needed
    pm.processResource(this, dynamic);
}
```

First, I don't understand what this is all about.

You have buffers which are proxies, and the processing can't be done when the buffer proxy is instantiated.

Then, ProxyFactory.loader is null, because AsmClassLoader hasn't been instantiated, so the client can't send SOAPFaultImpl.

Do you have -Djava.system.class.loader=com.goldencode.p2j.classloader.MultiClassLoader ?

#132 - 02/05/2024 07:21 AM - Galya B

Constantin Asofiei wrote:

Then, ProxyFactory.loader is null, because AsmClassLoader hasn't been instantiated, so the client can't send SOAPFaultImpl.

Do you have -Djava.system.class.loader=com.goldencode.p2j.classloader.MultiClassLoader ?

It fixes the NPE, but I wonder why is MultiClassLoader needed to return a SOAP fault? Is it something customers know about when using SOAP from ABL?

#133 - 02/05/2024 07:23 AM - Constantin Asofiei

Galya B wrote:

... but I wonder why is MultiClassLoader needed to return a SOAP fault? Is it something customers know about when using SOAP from ABL?

I don't understand what you mean. MultiClassLoader is a requirement for the FWD server. How is this affecting the SOAP fault processing?

#134 - 02/05/2024 07:25 AM - Galya B

Constantin Asofiei wrote:

Galya B wrote:

... but I wonder why is MultiClassLoader needed to return a SOAP fault? Is it something customers know about when using SOAP from ABL?

I don't understand what you mean. MultiClassLoader is a requirement for the FWD server. How is this affecting the SOAP fault processing?

The issue I'm describing is in the FWD client, when a procedure is called via SOAP on the appserver. So the above-mentioned check is done client-side. I'm adding MultiClassLoader as class loader to the client run script.

#135 - 02/05/2024 07:29 AM - Constantin Asofiei

Galya B wrote:

The issue I'm describing is in the FWD client, when a procedure is called via SOAP on the appserver. So the above-mentioned check is done client-side. I'm adding MultiClassLoader as class loader to the client run script.

OK, I understand now what you mean. SoapFaultImpl is instantiated on client-side, in WebServiceImpl.invoke. This isn't right. This code needs to be fixed so this will happen on server-side.

#136 - 02/05/2024 07:33 AM - Galya B

Constantin Asofiei wrote:

Galya B wrote:

The issue I'm describing is in the FWD client, when a procedure is called via SOAP on the appserver. So the above-mentioned check is done client-side. I'm adding MultiClassLoader as class loader to the client run script.

OK, I understand now what you mean. SoapFaultImpl is instantiated on client-side, in WebServiceImpl.invoke. This isn't right. This code needs to be fixed so this will happen on server-side.

Do we have customers calling soap endpoints from ABL? I wonder what I'm getting into making such core changes.

#137 - 02/07/2024 03:53 AM - Galya B

I need some clarification on the test in `appsrv/soap/test/4glws/test_throw_error.p`:

```
block-level on error undo, throw.
{common/log_hlp.i}

define variable outChar as character no-undo.

run log-it('ABL 4glws throw error procedure : - start').

{appsrv/soap/test/4glws/connect.i}

run throw_error in hfwdObj(input 9407, output soapResult) no-error.
assert-err-num(9407,
  substitute({&msg_call_error}, 'throw_error', ''),
  {&line-number}).

run log-it('ABL 4glws throw error procedure : - end').

catch e as Progress.Lang.AppError :
  message e:ReturnValue.
end catch.

finally:
{appsrv/soap/test/4glws/disconnect.i}
end finally.
```

The error thrown by the external procedure should be silenced by `no-error`, then what `AppError` is caught after the end log?

#138 - 02/07/2024 07:40 AM - Marian Edu

Galya B wrote:

The error thrown by the external procedure should be silenced by `no-error`, then what `AppError` is caught after the end log?

Normally none, it's just that we used to have this `finally` block in all tests just to make sure we don't miss anything :)

You can safely ignore that in this case, as said most probably was only added as a default.

#139 - 02/07/2024 12:59 PM - Galya B

On QuitConditionException the appserver agent gets disconnected, but then AgentPool.idsToAppserver gets empty and consecutive requests from external clients fail due to NPE.

The OE doc says:

When QUIT is executed from within a procedure running on an AppServer, it terminates the ABL session running on the AppServer, causing the AppServer agent to shut down and returns to the ABL client session from which it was spawned.

So I guess this is the behavior disconnect in Agent is simulating. The issue is that Agent disconnects the AgentPool, but AppServerHelper for the LegacyServiceWorker remains connected. I think I've fixed it by disconnecting the helper in SoapHandler, but it may need to be fixed for other handlers too.

#140 - 02/08/2024 10:47 AM - Galya B

There are several common error messages for the SOAP requests, but it's difficult to compile requirements for when each one should be returned, because FWD has a very different execution environment / process. So I'll try to summarize here what shows up in the tests:

- An error was detected while executing the Web Service request. (10893)
 - classic test cases: dataset, exceptions, primitives, table, type
 - PASOE test cases: same
- An error was detected in the Web Service request. (10894)
 - classic test cases: primitives, table, exceptions
 - PASOE test cases: primitives, table, type
- An internal Web Service error occurred in the Web Services Adapter. (10902)
 - classic test cases: dataset, exceptions, type
 - PASOE test cases: not relevant
- Exception running ABL procedure:
 - we have it only in PASOE tests, but in webtests_from_logs dir there are also examples from classic SOAP requests
 - classic test logs: dataset
 - PASOE test cases: dataset, exceptions, primitives, type

There is already an attempt to match some errors to exceptions in the code, but it's far from sufficient. I may need to focus on everything else left in this task, because this issue is quite the puzzle.

#141 - 03/18/2024 09:08 AM - Galya B

- Status changed from WIP to Review

- versioninfo fields are read-only, but writing them is a runtime error and not compile error in FWD.
- session/local_version_info.p - pass.
 - oerequestinfo: adapterType requires enum, which is not supported by FWD yet
- appsrv/test/dataset/test_get_current_request_info.p - pass.
- appsrv/test/dataset/test_get_current_response_info.p - pass.
- skeleton change required on merge, ref [#6417-15](#).
 - agentId, sessionId, threadId: these are 'plugged in' but the Agent doesn't set the to anything (remain unknown).
- Values added to sessionId and threadId.
 - VersionInfo:clientType supports only 4GLCLIENT and APPSERVER at this time
- TO DO: implement [#6417-129](#).
 - SESSION:CURRENT-REQUEST/RESPONSE-INFO don't know yet how this can be unknown or not
- the condition found and implemented.
 - some errors throw by SERVER:CONNECT are not the same
- tests in testcases/appsrv/test/connect_errors/:
 - appsrv/test/connect_errors/direct_connect.p - pass.
 - appsrv/test/connect_errors/host_name.p - pass.
 - appsrv/test/connect_errors/ip_address.p - pass.
 - appsrv/test/connect_errors/port_number.p - pass.
 - appsrv/test/connect_errors/service_name.p - pass.
 - appsrv/test/connect_errors/session_model_incorrect.p - pass.
 - appsrv/test/connect_errors/session_model_mismatch.p - needs more work.
 - argument translation to remote definition type I think is done the same as the CALL statement does it...
- not sure what this is.
 - OO instance serialization and transmission to remote side (GES: are we already handling this in [#4658](#)?)
- to be handled in [#4658](#).
 - a QUIT on the remote side quits the client
- appsrv/test/test_raise_quit.p - pass.
 - misc errors related to the ERROR, STOP, RETURN ERROR, raised by the remote side
- appsrv/test/test_throw_error.p - pass with a fix in testcases, ref [#6417-24](#).
- appsrv/test/test_return_error.p - pass.
- appsrv/test/test_raise_stop.p - pass.

There is more than that on the branch: the changes are quite into the core features and overall very important in my opinion, so I want to merge 6417a with these fixes soon and continue on another branch with the other (mostly rest / soap) tests failing.

6417a r15080 (re)based on trunk r15062 ready for review.

#142 - 03/25/2024 06:32 AM - Galya B

I'm not sure if my reasoning is correct, but it looks like the same appserver agent can execute all types of appserver tasks (rest, soap, webhandler, classic, etc). The tasks get into a queue and are running in sequence. This means client type for the session (the connection between the agent and the server) should be set on task start and reset on task complete. Unfortunately the task itself is presented by Runnable and can't reveal where it comes from (the client type). And this makes simply setting an attr to the session not so simple.

#143 - 03/25/2024 06:40 AM - Galya B

Actually there several places where appserver tasks are executed in a different way. I've found these:

- Agent.AgentCommand.doExecuteImpl
- LegacyServiceWorker.run

#144 - 03/25/2024 09:05 AM - Greg Shah

I'm not sure if my reasoning is correct, but it looks like the same appserver agent can execute all types of appserver tasks (rest, soap, webhandler, classic, etc). The tasks get into a queue and are running in sequence.

This is correct.

Unfortunately the task itself is presented by Runnable and can't reveal where it comes from (the client type). And this makes simply setting an attr to the session not so simple.

We could require the usage of a new interface that extends Runnable and has a method that returns an enum instance which describes the type.

#145 - 03/28/2024 09:44 AM - Galya B

Test in OE for client type:

Procedure running in batch client:

```
def var h as handle.
```

```
create server h.
```

```
h:connect("-DirectConnect -S 3111 -H localhost -sessionModel Session-Free", "", "").
```

```
message "session:local-version-info:OEClientType " session:local-version-info:OEClientType view-as alert-box error.
```

```
message "session:current-request-info " session:current-request-info view-as alert-box error.
```

```
message "session:current-response-info " session:current-response-info view-as alert-box error.
```

```
message "h:request-info:OEClientType " h:request-info:VersionInfo:OEClientType view-as alert-box error.
```

```
message "h:response-info:OEClientType " h:response-info:VersionInfo:OEClientType view-as alert-box error.
```

```
run typeonapp.p on server h.
```

```
h:disconnect().  
delete object h.
```

Results in clientlog:

```
4GLCLIENT  
?  
?  
4GLCLIENT  
JAVACLIENT
```

External procedure typeonapp.p running on appserver:

```
message "session:local-version-info:OEClientType " session:local-version-info:OEClientType view-as alert-box e  
rror.  
message "session:current-request-info " session:current-request-info view-as alert-box error.  
message "session:current-response-info " session:current-response-info view-as alert-box error.
```

Results in appserver log:

```
APPSERVER  
4GLCLIENT  
APPSERVER
```

The same results with AIA (connecting to appserver via http url), when using localhost, i.e.:

```
h:connect("-URL http://localhost:8080/aia/Aia?AppService=gbb -sessionModel Session-Free").
```

As you can see the types in the request and response are different. Also JAVACLIENT came up from nowhere in the batch 4GL client. Also, AIA adapter is not indicating AIA client, at least testing with localhost.

#146 - 03/28/2024 09:49 AM - Galya B

- Status changed from Review to WIP

#147 - 04/30/2024 08:24 AM - Galya B

Marian, do you have any idea why this test executed as batch in 4GL gives the following result:

```
def var h as handle.  
  
create server h.  
h:connect("-DirectConnect -S 3111 -H localhost -sessionModel Session-Free", "", "").  
  
message h:response-info:VersionInfo:OEClientType.  
  
h:disconnect().  
delete object h.
```

Message displayed: JAVACLIENT.

I have no idea how it can be replicated in FWD if it's not explainable.

#148 - 04/30/2024 08:49 AM - Marian Edu

Galya B wrote:

Marian, do you have any idea why this test executed as batch in 4GL gives the following result:

[...]

Message displayed: JAVACLIENT.

I have no idea how it can be replicated in FWD if it's not explainable.

Gayla, we don't have the 'classic' application server available anymore so we can only test using the 'pacific' (pasoe) Tomcat based version... but anyway, that 'RESPONSE-INFO' is available even before connect - to set the 'CLIENT-PRINCIPAL' mainly. If you run that from 4GL client I don't see how it could ever be 'JAVACLIENT' unless it just somehow points to the last request served by the app server agent. In you test you didn't actually ran anything on the app server, try to do a call and see if that changes anything.

Files

no-service-name.png	265 KB	01/15/2024	Galya B
01-invokeRemoteWithModelImpl.png	290 KB	01/16/2024	Galya B
02-invokelImpl.png	309 KB	01/16/2024	Galya B
losing-P2J_NOHOSTVERIFY.png	126 KB	01/18/2024	Galya B
HTTPSender.sendViaPost.png	465 KB	01/22/2024	Galya B