Harness - Feature #7034

implement FWD appserver client features

01/10/2023 05:42 PM - Greg Shah

Status:	Rejected		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:			**% Done:**	0%
Category:			**Estimated time:**	0.00 hour
Target version:				
billable:	No		**vendor_id:**	GCD
Description				

History

#1 - 01/10/2023 05:47 PM - Greg Shah

The idea is to implement test directives and support that would enable a full suite of FWD appserver tests to be written and executed using the Harness. I imagine the facility to be very similar to the REST/SOAP support except using the FWD replacement open client support.

Constantin: What would you expect the effort to be to write this?

#2 - 01/11/2023 04:26 AM - Constantin Asofiei

OpenClient code can be tested only using Java, I don't see how this can be scripted from Harness. I think JUnit would be better for this. There is the temp-table/data-type parameter tests I wrote for 6129b, which can be re-written (or enhanced) to use JUnit.

#3 - 01/11/2023 04:37 AM - Marian Edu

Greg, if this is about running the code from the 4GL client - CONNECT&RUN ON SERVER - then we don't need anything special nor use Harness. My point about this in the unittest task is that the tests must assume an application server is available and some code must be deployed on it so we can actually get to call the right support procedures otherwise the tests will fail - either because connect errors or missing support procedures.

#4 - 01/11/2023 08:37 AM - Greg Shah

- Status changed from New to Rejected

> OpenClient code can be tested only using Java, I don't see how this can be scripted from Harness.

This is less about testing the Open Client code (at least the generated proxies) and more about being able to automate tests of the server side. I was thinking that we didn't really care about the proxy code and instead could just define the server-side interface dynamically and provide test directives to call it and handle responses.

If this is just a bad idea, then we will choose another path.

> I think JUnit would be better for this. There is the temp-table/data-type parameter tests I wrote for 6129b, which can be re-written (or enhanced) to use JUnit.

OK.

#5 - 01/11/2023 08:38 AM - Greg Shah

My point about this in the unittest task is that the tests must assume an application server is available and some code must be deployed on it so we can actually get to call the right support procedures otherwise the tests will fail - either because connect errors or missing support procedures.

This is true for all converted 4GL unit testing in FWD. In all cases, we must have a FWD application server there with the converted code and the proper configuration. The appserver configuration is certainly more complex, but it is same thing.