

Build and Source Control - Support #7850

questions and answers about ABLUnit support in FWD

09/27/2023 03:43 PM - Greg Shah

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

History

#1 - 04/12/2024 01:51 PM - Roger Borrello

I get this when the converted test suite (1 test) tries to compile:

```
[javac] /home/rfb/projects/testcases_v2_dev2/src/com/goldencode/testcases/tests/base_language/stanza_ini/Suite.java:14: error: incompatible types: Suite cannot be converted to Annotation
[javac] @Suite
[javac] ^
[javac] 1 error
```

The 4GL for Suite.cls is:

```
block-level on error undo, throw.
```

```
@TestSuite(classes="tests.base_language.stanza_ini.Test1").
```

```
class tests.base_language.stanza_ini.Suite:
end class.
```

#2 - 04/19/2024 11:27 AM - Roger Borrello

I was attempting to build ABL unit for FWD using `ant -Dtests.module=base_language ablunit-tests-gui`. Is that correct? I received an error when it was processing DBConnection.

```
rfb@rfb:~/projects/testcases_v2_dev$ ant -Dtests.module=base_language ablunit-tests-gui
```

Buildfile: /home/rfb/projects/testcases_v2_dev/build.xml

ablunit-tests-gui:

```
[taskdef] Could not load definitions from resource PCT.properties. It could not be found.  
[typedef] Could not load definitions from resource types.properties. It could not be found.
```

BUILD FAILED

```
/home/rfb/projects/testcases_v2_dev/build.xml:1056: The following error occurred while executing this line:  
/home/rfb/projects/testcases_v2_dev/build_ablunit.xml:56: Problem: failed to create task or type DBConnection  
Cause: The name is undefined.
```

Action: Check the spelling.

Action: Check that any custom tasks/types have been declared.

Action: Check that any <presetdef>/<macrodef> declarations have taken place.

#3 - 04/19/2024 07:42 PM - Roger Borrello

I was finally able to figure out how to get a test for stanza_ini to run under ABL Unit. I have a single test in the suite:

```
using Progress.Lang.*.  
using OpenEdge.Core.Assert from propath.  
using support.test.*.  
  
block-level on error undo, throw.  
  
/* Core tests for stanza INI file usage, refs #8353 */  
  
define variable gSupWarns as logical no-undo.  
define variable gIniName as character.  
define variable gDir as character.  
  
@Setup.  
procedure setUp:  
    assign  
        gIniName = "env.ini"  
        gDir = "./"  
        gSupWarns = session:suppress-warnings.  
        session:suppress-warnings = false.  
  
    os-delete ./env.ini no-error.  
end procedure.  
  
@Test.  
procedure Test1:  
  
    define variable errNums as integer extent 1 no-undo.  
    define variable errMsg as character extent 1 no-undo.  
  
    /* Case 1 load should fail. No oddities */  
    assign  
        errNums[1] = 4449  
        errMsg[1] = "The LOAD of " + string(gIniName) + " in path " + string(gDir) + " failed. (" + string(er  
rNums[1]) + ")".  
  
    LOAD gIniName DIR gDir no-error.  
    support.test.AssertExt:Errors(errNums, errMsg, true).  
  
end procedure. /* Test */  
  
@TearDown.  
procedure TearDown:
```

```
os-delete ./env.ini no-error.  
assign session:suppress-warnings = gSupWarns.  
end procedure.  
  
/* EOF */
```

The LOAD of the ini is supposed to take place on the client. I am getting an NPE reported, but the 4GL Unit Testing wiki [section on running the client](#) has a note:

Note: there is probably no need to run the client in debug mode, since all unit test code is executed on the server, and nothing interesting happens at the client side.

The command I'm running is:

```
java -Djava.library.path=../../p2j/build/lib -jar /home/rfb/projects/junit-platform-console-standalone-1.9.0.jar --class-path ./lib/testcases.jar:../../p2j/build/lib/p2j.jar --select-class com.goldencode.testcases.tests.base_language.stanza_ini.TestSuiteClass
```

I build this command by trial-and-error, and it turns out I do need to debug the client, in order to determine why I was getting an NPE. By adding `-Xdebug -Xrunjdpw:transport=dt_socket,server=y,suspend=n,address=2999` to my command, I was able to debug the client. Perhaps the wiki needs to strike that?

I also would like to know if I am invoking the jUnit console properly, as the wiki mentions using `junit-platform-console-1.9.x.jar`, but that doesn't include all the dependencies required.

#4 - 04/20/2024 10:52 AM - Greg Shah

Note: there is probably no need to run the client in debug mode, since all unit test code is executed on the server, and nothing interesting happens at the client side.

This is not correct. Anything that normally happens on the FWD client (redirected UI for reports, access to the file system, stream processing,

sockets, native library calls...) will still happen on the client.

Please edit the documentation to correct that.

I'll let Vladimir answer your other question about invoking junit.

#5 - 04/20/2024 02:48 PM - Vladimir Tsichevski

Roger Borrello wrote:

I get this when the converted test suite (1 test) tries to compile:
[...]
The 4GL for Suite.cls is:
[...]

You should not name your 4gl class as Suite, otherwise you have a name clash in the converted Java code between the your Java class name and the annotation name.

#6 - 04/20/2024 03:32 PM - Vladimir Tsichevski

Roger Borrello wrote:

The LOAD of the ini is supposed to take place on the client. I am getting an NPE reported, but the 4GL Unit Testing wiki [section on running the client](#) has a note:

Note: there is probably no need to run the client in debug mode, since all unit test code is executed on the server, and nothing interesting happens at the client side.

The key word is "probably". I myself do run both the server and client parts from eclipse, unless I need to debug FWD in native ChUI mode. From my experience, I rarely need the Debug Eclipse launching configuration for the client process, and use the Run Eclipse command instead.

The command I'm running is:
[...]

I build this command by trial-and-error, and it turns out I do need to debug the client, in order to determine why I was getting an NPE. By adding `-Xdebug -Xrunjwp:transport=dt_socket,server=y,suspend=n,address=2999` to my command, I was able to debug the client. Perhaps the wiki needs to strike that?

The wiki just defines minimum requirement for the Java process options, set be the JUnit5 console launcher and FWD. You can debug the client Java proces as any other Java process, there are no specific requirements or restriction. The wiki does not describe Java debugging, since this is unrelated.

In your case, I would use `suspend=y` instead of `suspend=n`, so the process will pause until you connect with debugger.

I also would like to know if I am invoking the junit console properly, as the wiki mentions using `junit-platform-console-1.9.x.jar`, but that doesn't include all the dependencies required.

I've update the page: added more details about the FWD client classpath and removed the explicit reference to `junit-platform-console-1.9.x.jar`.