User Interface - Bug #3723

Pressing ESC key twice doesn't allow the web client to log out properly

09/21/2018 09:03 AM - Sergey Ivanovskiy

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

Description

History

#1 - 09/21/2018 09:06 AM - Sergey Ivanovskiy

- Subject changed from Pressing ESC key twice doesn't allow the web client to be logged out properly to Pressing ESC key twice doesn't allow the web client to log out properly

Found that if we executed a simple GUI program in the web client and press ESC key, then "Procedure complete ..." status is displayed. If we press ESC at the second time instead of pressing SPACE, then the connection with the web client will be broken, and the web client doesn't log out properly.

#2 - 04/30/2019 05:17 PM - Greg Shah

- Assignee set to Alexei Kaigorodov

- Start date deleted (09/21/2018)

#3 - 05/30/2019 06:48 AM - Alexei Kaigorodov

Sergey Ivanovskiy wrote:

> Found that if we executed a simple GUI program in the web client and press ESC key, then "Procedure complete ..." status is displayed.

1. which web client - descktop or embedded?
2. Could not find in documentation how to execute simple GUI program. Only found how to run hotel GUI demo in web mode, but server fails when trying to execute spawner - its path is not configured and equals to "[spawner-path]", instead of correct path "/opt/spawner/spawn". How to set spawner path when starting the server?

#4 - 05/30/2019 07:47 AM - Sergey Ivanovskiy

- File web_test_list_8 added

- File directory.xml added

- File build_testcase.sh added

Alexei Kaigorodov wrote:

> Sergey Ivanovskiy wrote:
>
> > Found that if we executed a simple GUI program in the web client and press ESC key, then "Procedure complete ..." status is displayed.
>
> 1. which web client - descktop or embedded?

The web client.

> 2. Could not@@ find in documentation how to execute simple GUI program. Only found how to run hotel GUI demo in web mode, but server fails when trying to execute spawner - its path is not configured and equals to "[spawner-path]", instead of correct path "/opt/spawner/spawn". How to set spawner path when starting the server?

getting_started_with_p2j_development.html has a paragraph about a working testcases project for developers.

In this example it is supposed that all projects are under ~/projects.
Please checkout testcases project

```
cd ~/projects
bzr co ~/secure/code/p2j_repo/testcases/
```

and set up its environment. All programs to convert are in uast folder ~/project/testcases/uast
Create p2j link to your branch or to trunc branch.

```
cd uast
ln -s ~/projects/p2j p2j
```

In my practice I am using this simple script ~/bin/build_testcase.sh to build the subset of the converted programs from this uast folder:

```
cd ~/projects/testcases/uast
build_testcases.sh web_test_list_8
```

web_test_list_8 has a list of 4GL programs to convert and to compile into jar testcases.jar that will be used to start simple server

```
cd ~/projects/testcases/simple/server
```

This folder contains directory.xml. Thus I advice you to set up this directory.xml to run a standalone program and to prepare a file of 4GL programs to convert. Also please add this project to your IDE environment and setup simple server to run from this environment. I am using eclipse now. Later I will find out helpful resources from my email or docs if nobody else will do this for you.

#5 - 05/30/2019 11:36 AM - Alexei Kaigorodov

I already have project testcases installed, and can run testusing 4WD server and swing client. All I need is to run a program with web client. So I copied your directory.xml to testcases/simple/server, started server with my old script and tried to point browser to https://localhost:7449/gui, but got message "Firefox can't establish a connection to the server at localhost:7449." I set port 7449 because it is mentioned in the directory.xml.

So questions remain:
- how to start web server?
- which url to open?
and where is the file getting_started_with_p2j_development.html? I know https://proj.goldencode.com/projects/p2j/wiki/Getting_Started but it says nothing about web client.

#6 - 05/30/2019 12:40 PM - Sergey Ivanovskiy

Alexei Kaigorodov wrote:

> I already have project testcases installed, and can run testusing 4WD server and swing client. All I need is to run a program with web client. So I copied your directory.xml to testcases/simple/server, started server with my old script and tried to point browser to https://localhost:7449/gui, but got message "Firefox can't establish a connection to the server at localhost:7449." I set port 7449 because it is mentioned in the directory.xml.
>
> So questions remain:
> - how to start web server?
> - which url to open?
> and where is the file getting_started_with_p2j_development.html? I know https://proj.goldencode.com/projects/p2j/wiki/Getting_Started but it says nothing about web client.

Alexei, the attached directory.xml is an example. It needs to change some parameters that are fitted to your environment and to build p2j with these keys
-Dspawn.install.folder and -Dsrv.certs

```
ant all -Dpost.build=yes -Dspawn.install.folder=/home/sbi/opt -Dsrv.certs=/home/sbi/projects/testcases/simple/server/srv-certs.store
```

It needs to change the OS user, p2j-entry and the web client settings:

```
          <node class="string" name="spawner">
            <node-attribute name="value" value="/home/sbi/opt/spawner/spawn"/>
          </node>
```

In my example they are

```
<node class="container" name="webClient">
          <node class="boolean" name="enabled">
            <node-attribute name="value" value="TRUE"/>
          </node>
          <node class="boolean" name="virtualDesktopEnabled">
            <node-attribute name="value" value="TRUE"/>
          </node>
          <node class="boolean" name="embedded">
            <node-attribute name="value" value="FALSE"/>
          </node>
          <node class="boolean" name="graphicsCached">
            <node-attribute name="value" value="TRUE"/>
          </node>
          <node class="string" name="host">
            <node-attribute name="value" value="192.168.1.37"/>
          </node>
          <node class="integer" name="port">
            <node-attribute name="value" value="7449"/>
```

```
          </node>
          <node class="string" name="uploadSingleFile">
            <node-attribute name="value" value="200M"/>
          </node>
          <node class="string" name="uploadTotalFiles">
            <node-attribute name="value" value="1000M"/>
          </node>
          <node class="container" name="portsRange">
            <node class="string" name="namePrefix">
              <node-attribute name="value" value="client"/>
            </node>
            <node class="integer" name="from">
              <node-attribute name="value" value="7449"/>
            </node>
            <node class="integer" name="to">
              <node-attribute name="value" value="7450"/>
            </node>
          </node>
          <node class="integer" name="maxBinaryMessage">
            <node-attribute name="value" value="32894"/>
          </node>
          <node class="integer" name="maxIdleTime">
            <node-attribute name="value" value="90000"/>
          </node>
          <node class="integer" name="watchdogTimeout">
            <node-attribute name="value" value="120000"/>
          </node>
          <node class="container" name="desktop">
            <node class="string" name="background">
              <node-attribute name="value" value="0x000000"/>
            </node>
            <node class="string" name="background-image">
              <node-attribute name="value" value="fwd.png"/>
            </node>
            <node class="string" name="image-mime-type">
              <node-attribute name="value" value="image/png"/>
            </node>
            <node class="string" name="background-origin">
              <node-attribute name="value" value="border-box"/>
            </node>
            <node class="string" name="background-position">
              <node-attribute name="value" value="center"/>
            </node>
            <node class="string" name="background-size">
              <node-attribute name="value" value="100% 100%"/>
            </node>
            <node class="string" name="background-repeat">
              <node-attribute name="value" value="no-repeat"/>
            </node>
          </node>
        </node>
```

```
        <node class="container" name="clientConfig">
          <node class="string" name="spawner">
            <node-attribute name="value" value="/home/sbi/opt/spawner/spawn"/>
          </node>
          <node class="string" name="workingDir">
            <node-attribute name="value" value="/home/sbi/projects/testcases/uast/demo"/>
          </node>
          <node class="string" name="jvmArgs">
            <node-attribute name="value" value="-Xmx512m -XX:MaxPermSize=64m -Djava.awt.headless=true -Xdebug
-Xnoagent -Djava.compiler=NONE -Xrunjdwp:transport=dt_socket,address=9999,server=y,suspend=n"/>
          </node>
          <node class="boolean" name="underline-keyboard-shortcuts">
            <node-attribute name="value" value="FALSE"/>
          </node>
        </node>
```

and
the default OS user

```
          <node class="string" name="defaultAccount">
            <node-attribute name="value" value="sbi"/>
```

```
        </node>
```

and

```
      <node class="container" name="trustedspawner">
        <node class="container" name="000100">
          <node class="resource" name="resource-instance">
            <node-attribute name="reference" value="sbi"/>
            <node-attribute name="reftype" value="TRUE"/>
          </node>
          <node class="trustedSpawnerRights" name="rights">
            <node-attribute name="allow" value="true"/>
          </node>
          <node class="strings" name="subjects">
            <node-attribute name="values" value="ehotel"/>
          </node>
        </node>
      </node>
```

are not required and these setting are absent in my directory.xml, but for trusted spawner the OS user setup is required.

Finally the web server

```
      <node class="integer" name="adminPort">
        <node-attribute name="value" value="7443"/>
      </node>
```

must be available at https://192.168.1.37:7443/gui and the server must respond with the login form in which it needs to fill with your user login and password, then the server will redirect to new address https://192.168.1.37:7449 (depending on the settings and the number of running clients)

#7 - 05/30/2019 02:36 PM - Greg Shah

> and where is the file getting_started_with_p2j_development.html?

Getting Started with FWD Development

Please note this is about the testcases project, not the web client.

> 2. Could not@@ find in documentation how to execute simple GUI program. Only found how to run hotel GUI demo in web mode, but server fails when trying to execute spawner - its path is not configured and equals to "[spawner-path]", instead of correct path "/opt/spawner/spawn". How to set spawner path when starting the server?

In the Hotel GUI you must follow the instructions carefully. I think you never ran the prepare_hotel.sh as documented there.

#8 - 05/31/2019 11:35 AM - Alexei Kaigorodov

> This folder contains directory.xml.

which folder?

> Thus I advice you to set up this directory.xml to run a standalone program

How to set up directory.xml? Which standalone program to run? How at all xml file can run a program?

> Also please add this project to your IDE environment

I added this project in Inellij Idea, but still use scripts to recompile and run.

> and setup simple server to run from this environment.

What server do you mean? com.goldencode.p2j.main.ServerDriver or some arbitrary web server? I tried to add a run configuration to start com.goldencode.p2j.main.ServerDriver but failed - too many parameters have to be set up correctly. Instead, I run the script testcases/simple/server/server.sh and attach debugger if needed.

cd ~/projects/testcases/simple/server

This folder contains directory.xml.

which folder?

I think he is clearly referencing ~/projects/testcases/simple/server in this case. But a simple use of find in the Hotel GUI project would find the file deploy/server/directory.xml. You can run the web client in either the testcases/uast/ project or in the Hotel GUI project.

Thus I advice you to set up this directory.xml to run a standalone program

How to set up directory.xml?

#3723-6 provides specific details of the changes to the directory.xml, which you can make with an editor.

As an alternative, if you are using Hotel GUI, you can run prepare_hotel.sh to make the equivalent edits to the deploy/server/directory.xml.

In either case, the web client should then be configured properly in the FWD server. Of course, Sergey's other advice about building FWD in #3723-6 must also be followed.

Which standalone program to run? How at all xml file can run a program?

Change the p2j-entry value in the directory to set the Java class (and execute method) of the converted program that will be executed by a user's session.

and setup simple server to run from this environment.

What server do you mean? com.goldencode.p2j.main.ServerDriver or some arbitrary web server? I tried to add a run configuration to start com.goldencode.p2j.main.ServerDriver but failed - too many parameters have to be set up correctly. Instead, I run the script testcases/simple/server/server.sh and attach debugger if needed.

He means the FWD server, which you can run from testcases/simple/server/server.sh or in Hotel GUI from deploy/server/server.sh.

#10 - 06/10/2019 07:50 AM - Alexei Kaigorodov

Cannot reproduce.

Sergey Ivanovskiy wrote:

> Found that if we executed a simple GUI program in the web client and press ESC key, then "Procedure complete ..." status is displayed. If we press ESC at the second time instead of pressing SPACE, then the connection with the web client will be broken, and the web client doesn't log out properly.

I could not to start a simple GUI program. I could only execute hotel gui demo in virtual desktop mode (https://localhost:7443/gui). After login and playing with buttons, I pressed ESC key, but no "Procedure complete ..." status was displayed. Instead, the application returned to first login screen, asking for O/S credentials. Once the credentials are provided, the next login screen appears etc, that is, connection is not broken.

If you want me to investigate the question further, please provide a working standalone application which I could build and run without much efforts.

#11 - 06/10/2019 12:31 PM - Sergey Ivanovskiy

This issue can be reproduced with this program testcases/uast/demo/demo_widgets.p. Alexei, could you convert this program from testcases project and run this program standalone?

#12 - 06/10/2019 12:44 PM - Sergey Ivanovskiy

I didn't describe how to create runnable configuration of simple/server and simple/client clearly. It needs to refresh my memory. You can add the package demo/demo_widgets.p to your hotel_gui 4gl source code (abl directory) and then convert and compile it like hotel_gui. Then please change p2j-entry to com.goldencode.hotel.demo.DemoWidgets.execute.

#13 - 06/10/2019 12:47 PM - Sergey Ivanovskiy

Please change p2j-entry to com.goldencode.hotel.demo.DemoWidgets.execute.

#14 - 06/12/2019 08:40 AM - Alexei Kaigorodov

So I took hotel_gui project and replaced the abl code with demo/demo_widgets.p taken from testcases/uast/, built and started the server.

Then, when opening https://localhost:7443/embedded I see web page named "Embedded Web Client Demo" asking for login/password. After login/password are put, the page prints "Loading..." and hangs. The file server.log was appended with single line:
No root access permissions for srv-certs.store.
without any information about the class/method or code file/line number. I could not find where this text is printed.

When opening https://localhost:7443/gui I see web page named "Operating System Login (FWD GUI Web client)" asking for login/password. After login/password are put, the window of demo/demo_widgets.p appeared.

After pressing ESC key twice, it hangs for couple of minutes and then returns to the O/S login page.
Interesting, the browser (Firefox) beeps when the mouse cursor hovers the demo_widget window.
The file server.log was appended with two lines:

com.goldencode.p2j.util.EndConditionException: END-ERROR key function
[06/12/2019 09:42:34 NOVT] (TransactionManager.handleDeferredError:SEVERE) {00000005:00000016:bogus} <depth = 0; trans_level = -1; trans_label = null; rollback_scope = -1; rollback_label = null; rollback_pending = false; in_quit = false; retry_scope = -1; retry_label = null; ignore_err = false> Throwing deferred error (END-ERROR key function)

Then, unexpected troubles began. I tried to find out how ESC key is handled. I could not trace it in debugger, because first I have to make a mouse click on the application window, which also makes breakpoit stop, and to let program run further I have left the application window and to go to the debugger window. I decided simply pring all the events which come to server, but everything I tried had no effect:

```
String msg = "processInbound:" + key.toString();
          System.out.println(msg);
          System.err.println(msg);
          LOG.log(Level.INFO, msg);
          SessionManager.LOG.log(Level.INFO, msg);
```

as if printing was turned off completely. I set a breakpoint to this code, and I see it is executed, but no output seen.

Then I tried to find where the string "Procedure complete. Press space bar to continue." is printed. This string is the constant ThinClient.PBE_DEFAULT_TEXT, and the only place where it is used is ThinClient.ThinClient(). I set a breakpoint there, but it did not work.

Need an advice in which direction to move further.

#15 - 06/13/2019 02:28 PM - Hynek Cihlar

When ESC is pressed an end-condition is raised in the legacy stack to interrupt any routines that wait for user input (wait-for, pause, etc.).

The problem here is that when the second end-condition is raised (after second ESC pressed), the client driver logic bypasses the proper shutdown. During proper shutdown GuiWebDriver.shutdown() is called, which sends MSQ_QUIT message to the JS part of the driver and gracefully stops the embedded web server. Either of these (whichever comes first) cause a redirect to the login page (messageHandler or ws.onclose in p2j.socket.js). When proper shutdown is bypassed, GuiWebDriver.shutdown() is not called and instead the Java client process is exited in CommonDriver.process() with the call System.exit(-13).

I think the solution in this case is to protect ClientCore.start() and do the proper shutdown even when the second end condition is raised.

Btw. this problem goes beyond the Web driver. During the abrupt exit, the FWD network session is not closed either, see session.terminate(); in ClientCore.start().

#16 - 06/13/2019 03:00 PM - Alexei Kaigorodov

Sorry I do not understand anything.

Hynek Cihlar wrote:

> When ESC is pressed an end-condition is raised in the legacy stack to interrupt any routines that wait for user input (wait-for, pause, etc.).

- what is end condition ?

- how it is raised?
- what is the legacy stack?
- I know how a thread can be interrupted but do not know how to interrupt routines that wait for user input
- when ESC is pressed, some callback in JS code in browser is called, I think. Then this callback routine sends some message to the server, I think. But it is received by server in different way than ordinary user input, right? Which way?

> The problem here is that when the second end-condition is raised (after second ESC pressed), the client driver logic bypasses the proper shutdown.

- I thought client driver works in Java client programs, and in the web mode, the role of client driver plays Javascript code running in the browser

> During proper shutdown GuiWebDriver.shutdown() is called, which sends MSQ_QUIT message to the JS part of the driver and gracefully stops the embedded web server.

That is, you want to say that the embedded web server is located in the JS part of the driver?

> Either of these (whichever comes first) cause a redirect to the login page (messageHandler or ws.onclose in p2j.socket.js). When proper shutdown is bypassed, GuiWebDriver.shutdown() is not called and instead the Java client process is exited in CommonDriver.process() with the call System.exit(-13).
>
> I think the solution in this case is to protect ClientCore.start()

- what do you mean with "protect"?

> and do the proper shutdown even when the second end condition is raised.
>
> Btw. this problem goes beyond the Web driver. During the abrupt exit, the FWD network session is not closed either, see session.terminate(); in ClientCore.start().

- I believe there should exist the description of the protocol between all parts of the program, and the program code should follow that protocol. Otherwise, similar bugs will appear every now and then.

#17 - 06/13/2019 04:45 PM - Hynek Cihlar

Alexei Kaigorodov wrote:

> Hynek Cihlar wrote:
>
> > When ESC is pressed an end-condition is raised in the legacy stack to interrupt any routines that wait for user input (wait-for, pause, etc.).
>
> - what is end condition ?

I suggest you read the Progress documentation on this concept.
https://documentation.progress.com/output/ua/OpenEdge_latest/index.html#page/dverr%2Fintroduction-to-condition-handling.html%23

> - how it is raised?

Raising the conditions are in FWD implemented with ConditionException and its sub classes.

> - what is the legacy stack?

By legacy stack I meant the 4GL code call stack - statements, methods, functions, etc. For example for the following code sample:

```
define frame f...
on "x" anywhere do:
   enable all with frame f.
   wait-for go of frame f.
end.

wait-for close of current-window.
```

When user presses "x", the call stack will contain two wait-for statements with wait-for go on the top.

> - I know how a thread can be interrupted but do not know how to interrupt routines that wait for user input

One way to interrupt the wait-for go above is by pressing the GO button in frame f. Another way is an end condition.

> - when ESC is pressed, some callback in JS code in browser is called, I think. Then this callback routine sends some message to the server, I think. But it is received by server in different way than ordinary user input, right? Which way?

ESC keys are sent from JS to Java client process as any other key. Put a break point in WebClientProtocol.processBinaryMessage() to see the message. Eventually the ESC event will be transformed in an end condition. The transformation process is pretty complex and is out of scope for this issue. But you can trace the key event in the client process to find more details.

> > The problem here is that when the second end-condition is raised (after second ESC pressed), the client driver logic bypasses the proper shutdown.
>
> - I thought client driver works in Java client programs, and in the web mode, the role of client driver plays Javascript code running in the browser

We have multiple client drivers or screen drivers. Web GUI driver is one of them. See GuiWebDriver, which is the main driver class. This class runs in

the FWD client process. The driver starts and embedded web server (Jetty) that serves the web resources (html, ccs, js) to the HTTP agent (browser). When we refer to "FWD client" we mean the Java process that runs the driver. When we refer to "GUI driver" or "driver" or perhaps "Java driver" we mean GuiWebDriver and the related Java classes. When we refer to "JS driver" we refer to the Javascript code that is served by the embedded web server started by the Web GUI driver.

Beware of the class ClientDriver. This class doesn't belong to the screen drivers concept above. This class is responsible for initializing the Java client process.

> During proper shutdown GuiWebDriver.shutdown() is called, which sends MSQ_QUIT message to the JS part of the driver and gracefully stops the embedded web server.

> That is, you want to say that the embedded web server is located in the JS part of the driver?

See above.

> Either of these (whichever comes first) cause a redirect to the login page (messageHandler or ws.onclose in p2j.socket.js). When proper shutdown is bypassed, GuiWebDriver.shutdown() is not called and instead the Java client process is exited in CommonDriver.process() with the call System.exit(-13).
>
> I think the solution in this case is to protect ClientCore.start()

> - what do you mean with "protect"?

By protect, I mean catching the end condition exception, handle the finalization and rethrowing the exception.

> and do the proper shutdown even when the second end condition is raised.
>
> Btw. this problem goes beyond the Web driver. During the abrupt exit, the FWD network session is not closed either, see session.terminate(); in ClientCore.start().

> - I believe there should exist the description of the protocol between all parts of the program, and the program code should follow that protocol.

Agree.

> Otherwise, similar bugs will appear every now and then.

In this case the problem is not in the protocol, but the finalization logic of the FWD client process.

#18 - 06/14/2019 10:35 AM - Alexei Kaigorodov

Hynek Cihlar wrote:

> ESC keys are sent from JS to Java client process as any other key. Put a break point in WebClientProtocol.processBinaryMessage() to see the message.

Such a breakpoint did not work. Probably, this method works inside other process, not that which was launched by the script hotel_gui/deploy/server/server.sh.

Following breakpoints worked:

listenWorker:1235, RouterSessionManager (com.goldencode.p2j.net)

run:1437, RouterSessionManager$Incoming (com.goldencode.p2j.net)

listenWorker:1235, RouterSessionManager (com.goldencode.p2j.net)

run:555, Protocol$Writer (com.goldencode.p2j.net)

run:439, Protocol$Reader (com.goldencode.p2j.net)

can you explain what the roles these points play in the client-server communication?

#19 - 06/14/2019 11:38 AM - Greg Shah

> Probably, this method works inside other process, not that which was launched by the script hotel_gui/deploy/server/server.sh.

The server.sh starts the FWD server. This has nothing to do with the web client support. You need to connect to the FWD client for that.

Please see https://proj.goldencode.com/projects/p2j/wiki/Runtime_Architecture#Distributed-Application-Protocol-DAP

It is not useful to be looking at the problem at the DAP level. The DAP is the low level transport for implementing bidirectional remote object calls between the FWD server and the FWD client. As noted in the runtime architecture document, the converted 4GL code runs on the FWD server and the FWD client is a presentation engine and local platform delegate. In web client mode there is an extra tier for the browser which is javascript based.

The built-in 4GL event processing here can be more properly seen inside the FWD client itself. The DAP is not used between the FWD client and the JS code in the browser. That connection between the FWD client and the browser is by web socket as the runtime architecture document shows.

#20 - 06/17/2019 08:05 AM - Alexei Kaigorodov

Greg Shah wrote:

> The server.sh starts the FWD server. This has nothing to do with the web client support. You need to connect to the FWD client for that.

In Swing GUI mode, I manually start FWD client as a java program and can tell jvm to open debugging port.
In web mode, however, I start only server, and then work through the browser. I do not control launching of FWD client and cannot tell jvm that I want to debug the FWD client.

#21 - 06/17/2019 08:20 AM - Constantin Asofiei

Alexei Kaigorodov wrote:

> In web mode, however, I start only server, and then work through the browser. I do not control launching of FWD client and cannot tell jvm that I want to debug the FWD client.

In directory.xml, find the clientConfig node and add this child:

```
          <node class="string" name="jvmArgs">
            <node-attribute name="value" value="-Xmx512m -XX:MaxPermSize=64m -Djava.awt.headless=true -Xdebug
-Xnoagent -Djava.compiler=NONE -Xrunjdwp:transport=dt_socket,address=2998,server=y,suspend=n"/>
          </node>
```

If you don't terminate the client from the web browser, or from the attached debugging, you need to manually kill the ClientDriver java process before you are able to connect again.

#22 - 06/17/2019 09:07 AM - Alexei Kaigorodov

so I edited hotel_gui/deploy/server/directory.xml, found lientConfig and replaced jvmArgs with new value. Then ran hotel_gui/deploy/server/server.sh and tried to connect debugger to the port 2998, but IDE said "Connection refused". server.log and stdout.log contain no error messages, and do not mention 2998.

#23 - 06/17/2019 09:16 AM - Constantin Asofiei

Alexei Kaigorodov wrote:

> ... and tried to connect debugger to the port 2998, but IDE said "Connection refused".

This setting is for the Web client, not the FWD server. So login to a FWD Web client first, and after that try to connect.

Files

directory.xml	53.6 KB	05/30/2019	Sergey Ivanovskiy
build_testcase.sh	217 Bytes	05/30/2019	Sergey Ivanovskiy
web_test_list_8	651 Bytes	05/30/2019	Sergey Ivanovskiy