

## Base Language - Bug #2367

### re-connecting an appserver agent after a hard client termination

08/08/2014 12:43 PM - Ovidiu Maxiniuc

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Constantin Asofiei	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Cleanup and Stablization for Server Features	<b>case_num:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			

#### History

##### #1 - 08/08/2014 01:01 PM - Ovidiu Maxiniuc

When a client abruptly terminate (killed or otherwise disconnected because of network failure) the appserver enters a state from where it will not allow any more connections.

The only solution is to stop the server and a clean re-start.

This is the output of SimpleApiTest when it tries to reconnect:

```
java.lang.IllegalStateException: The appserver is not connected - check the connection result!
    at com.goldencode.p2j.util.AppServerHelper.checkConnected(AppServerHelper.java:1580)
    at com.goldencode.p2j.util.AppServerHelper.invoke(AppServerHelper.java:946)
    at com.goldencode.p2j.util.AppServerHelper.invoke(AppServerHelper.java:812)
    at SimpleApiTest.searchApiTest(SimpleApiTest.java:69)
    at SimpleApiTest.main(SimpleApiTest.java:204)
Caused by: com.goldencode.p2j.NumberedException: Connection refused by Application Server
    at com.goldencode.p2j.util.AppServerHelper.connect(AppServerHelper.java:354)
    at com.goldencode.p2j.util.AppServerHelper.connect(AppServerHelper.java:183)
    at SimpleApiTest.getConnection(SimpleApiTest.java:118)
    at SimpleApiTest.searchApiTest(SimpleApiTest.java:64)
    ... 1 more
Caused by: com.goldencode.p2j.NumberedException: No servers available
    at com.goldencode.p2j.util.AgentPool$BoundPool.newConnection(AgentPool.java:829)
    at com.goldencode.p2j.util.AgentPool.connect(AgentPool.java:321)
    at com.goldencode.p2j.util.AppServerManager.connect(AppServerManager.java:394)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:693)
    at com.goldencode.p2j.net.Dispatcher.dispatch(Dispatcher.java:809)
    at com.goldencode.p2j.net.Dispatcher$DispatcherStub.run(Dispatcher.java:1621)
    at java.lang.Thread.run(Thread.java:744)
java.lang.IllegalStateException: The appserver is not connected - check the connection result!
    at com.goldencode.p2j.util.AppServerHelper.checkConnected(AppServerHelper.java:1580)
    at com.goldencode.p2j.util.AppServerHelper.disconnect(AppServerHelper.java:537)
    at SimpleApiTest.searchApiTest(SimpleApiTest.java:89)
    at SimpleApiTest.main(SimpleApiTest.java:204)
Caused by: com.goldencode.p2j.NumberedException: Connection refused by Application Server
    at com.goldencode.p2j.util.AppServerHelper.connect(AppServerHelper.java:354)
    at com.goldencode.p2j.util.AppServerHelper.connect(AppServerHelper.java:183)
    at SimpleApiTest.getConnection(SimpleApiTest.java:118)
    at SimpleApiTest.searchApiTest(SimpleApiTest.java:64)
    ... 1 more
Caused by: com.goldencode.p2j.NumberedException: No servers available
    at com.goldencode.p2j.util.AgentPool$BoundPool.newConnection(AgentPool.java:829)
    at com.goldencode.p2j.util.AgentPool.connect(AgentPool.java:321)
    at com.goldencode.p2j.util.AppServerManager.connect(AppServerManager.java:394)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
```

```
at sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke (Method.java:606)
at com.goldencode.p2j.util.MethodInvoker.invoke (MethodInvoker.java:76)
at com.goldencode.p2j.net.Dispatcher.processInbound (Dispatcher.java:693)
at com.goldencode.p2j.net.Dispatcher.dispatch (Dispatcher.java:809)
at com.goldencode.p2j.net.Dispatcher$DispatcherStub.run (Dispatcher.java:1621)
at java.lang.Thread.run (Thread.java:744)
```

The server log does not read anything, in fact it is somewhat blocked. The new attempts to connect are not visible.

#### #2 - 08/08/2014 01:03 PM - Constantin Asofiei

Ovidiu, please try to duplicate again, and capture the server thread dump (use jstack for this). I'm curious where the server got stuck... there should be some kind of deadlock somewhere.

#### #3 - 08/08/2014 01:08 PM - Eric Faulhaber

- Target version set to Milestone 11

Ovidiu: this was achieved with min\_agents and max\_agents both set to 1, correct?

#### #4 - 08/08/2014 01:49 PM - Ovidiu Maxiniuc

- File blocked-server-threads.txt added

Eric: yes, both set to 1:

```
<node-attribute name="initial_agents" value="1"/>
<node-attribute name="min_agents" value="1"/>
<node-attribute name="max_agents" value="1"/>
```

If a larger number would have been used after n disconnects the server would also get blocked, but the issue would not be visible so fast.

Costel: Attached is the list of threads of a blocked server. Also, the server dump can be found at devsrv01://<temp\_path>/server.dump.7z.

LE: GES removed specific customer name and pathing from this entry.

#### #5 - 08/26/2014 04:47 PM - Constantin Asofiei

- Status changed from New to WIP

- % Done changed from 0 to 90
- File ca\_upd20140806b.zip added

This one was tricky to duplicate and even trickier to solve. The cause is this: if the appserver client terminates the P2J session BEFORE the appserver connection has been fully established, there was no notification to release the agent (i.e. call the appserver disconnect code). Thus, the agent remained assigned to that appserver connection, even if the P2J session associated with it was no longer alive.

I didn't duplicate this in the server project, but I assume this is what Ovidiu was doing: the appserver client was being killed during the connection call.

4GL calls the Disconnect procedure only if the connection was fully established. The 4GL rules implemented are:

- if the P2J session was terminated before or while the appserver connection call, then the agent is just released/cleaned up and nothing else is done
- else, if it was terminated after the appserver connection call, then normal processing is needed: the appserver disconnect code needs to be executed normally, as if it was explicitly called by the user.

The fix for this was tricky because deadlocks kept popping up, as the connect code had to keep track of **when** the P2J session was being terminated, and the session termination listener needed to wait until the appserver connection code was fully finished, before proceeding.

Attached update fixes this.

Eric: as Ovidiu is away, please do a quick test on the server project, by killing the client while:

1. the connect procedure is being executed
2. the normal appserver API is being executed
3. the disconnect procedure is being executed

If all these pass, then I can release it.

#### #6 - 08/26/2014 05:52 PM - Greg Shah

Code Review 0806b

Wow! That is tricky code with lots of subtlety. You documented it well and the approach makes sense. I have no problems with the changes. Good work!

#### #7 - 09/01/2014 10:40 AM - Ovidiu Maxiniuc

Constantin Asofiei wrote:

Eric: as Ovidiu is away, please do a quick test on the server project, by killing the client while:

1. the connect procedure is being executed
2. the normal appserver API is being executed
3. the disconnect procedure is being executed

If all these pass, then I can release it.

Eric: I understand you didn't have the time to test the update.

I am available now and I am about to start the validation of the update by executing the tests Constantin enumerated above.

#### #8 - 09/01/2014 04:26 PM - Ovidiu Maxiniuc

Constantin,

I tested on Linux/PostgreSQL with bzt rev 10603 and your update 0806b, initial\_agents, min\_agents and max\_agents all set to 1.

When client is killed during the API execution and disconnection everything seems normal. (I hope I suspended the server in the right locations, I did multiple tests here and these points look fine now). However, things change when the SimpleApiTest client is killed before the connection is completed. Any subsequent connection of the client to appserver will print the following stack (note that the server.log is clean):

```
java.lang.IllegalStateException: The appserver is not connected - check the connection result!
    at com.goldencode.p2j.util.AppServerHelper.checkConnected(AppServerHelper.java:1581)
    at com.goldencode.p2j.util.AppServerHelper.invoke(AppServerHelper.java:947)
    at com.goldencode.p2j.util.AppServerHelper.invoke(AppServerHelper.java:813)
    at SimpleApiTest.searchApiTest(SimpleApiTest.java:70)
    at SimpleApiTest.main(SimpleApiTest.java:225)
Caused by: com.goldencode.p2j.NumberedException: Connection refused by Application Server
    at com.goldencode.p2j.util.AppServerHelper.connect(AppServerHelper.java:355)
    at com.goldencode.p2j.util.AppServerHelper.connect(AppServerHelper.java:184)
    at SimpleApiTest.getConnection(SimpleApiTest.java:119)
    at SimpleApiTest.searchApiTest(SimpleApiTest.java:65)
    ... 1 more
Caused by: com.goldencode.p2j.NumberedException: No servers available
    at com.goldencode.p2j.util.AgentPool$BoundPool.newConnection(AgentPool.java:851)
    at com.goldencode.p2j.util.AgentPool.connect(AgentPool.java:328)
    at com.goldencode.p2j.util.AppServerManager.connect(AppServerManager.java:475)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:693)
    at com.goldencode.p2j.net.Dispatcher.dispatch(Dispatcher.java:809)
    at com.goldencode.p2j.net.Dispatcher$DispatcherStub.run(Dispatcher.java:1621)
    at java.lang.Thread.run(Thread.java:744)
```

Apparently it is the same error, only the line numbers are shifted in the AgentPool.java source.

I did not investigate the process, but when a new agent is requested, in getFromMainPool, mainPool is empty so null is returned.

**#9 - 09/02/2014 04:53 AM - Constantin Asofiei**

Ovidiu, you are correct, if the client is killed just before the session listener is registered, we still have problems.

But you can test something else: please place a breakpoint in the 4GL program which acts as the connect procedure, then kill the SimpleApiTest app. This should clean up properly.

**#10 - 09/02/2014 06:20 AM - Constantin Asofiei**

- File *ca\_upd20140902a.zip* added

Ovidiu, attached update should fix the remaining problem. The solution was to make the (queue state check, add session listener) code atomic, so that the queue can't be stopped while adding the listener; also, the registration fails if the queue is already stopped.

**#11 - 09/02/2014 09:59 AM - Ovidiu Maxiniuc**

Constantin Asofiei wrote:

Ovidiu, attached update should fix the remaining problem. The solution was to make the (queue state check, add session listener) code atomic, so that the queue can't be stopped while adding the listener; also, the registration fails if the queue is already stopped.

Indeed, all my test were successful, I could not reproduce the previous stack traces. I believe you can release the update.

**#12 - 09/02/2014 10:19 AM - Greg Shah**

Code Review 0902a

The changes look good.

**#13 - 09/03/2014 06:47 AM - Constantin Asofiei**

Greg Shah wrote:

Code Review 0902a

The changes look good.

Committed to rev 10604.

**#14 - 09/03/2014 03:37 PM - Greg Shah**

- Status changed from WIP to Closed

- % Done changed from 90 to 100

**#15 - 11/16/2016 12:06 PM - Greg Shah**

- Target version changed from Milestone 11 to Cleanup and Stabilization for Server Features

**Files**

---

blocked-server-threads.txt	18.8 KB	08/08/2014	Ovidiu Maxiniuc
ca_upd20140806b.zip	15.5 KB	08/26/2014	Constantin Asofiei
ca_upd20140902a.zip	22.2 KB	09/02/2014	Constantin Asofiei