Base Language - Bug #2637

batch client needs to detect if the client process has output redirected and change error handling

08/13/2015 11:04 AM - Greg Shah

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

Description

History

#1 - 08/13/2015 11:06 AM - Greg Shah

- Target version deleted (Milestone 12)

- Start date deleted (08/13/2015)

When a _progres executible launches, it can detect if its standard output is redirected. In batch mode, this changes certain behavior because the client is automatically put into redirected terminal mode. There is no need for the 4GL code to redirect the terminal in this case. This means it allows output to be written to the terminal (which can be the unnamed stream) and that output does not generate an error. In the case of a batch process that does not have its output redirected, any output to the terminal is an error. Currently we implement this error in all cases and we have no way to automatically shift into this redirected terminal mode. In other words, P2J current expects the 4GL code to do the redirecting, but it needs to honor the prior operating-system level redirection that may be there from the client process launch.

#2 - 08/13/2015 12:18 PM - Constantin Asofiei

I don't think the OutputManager.activateBatch(false); calls throughout ThinClient and Window are properly balanced - this is set to false for example in TC.putScreen but is never restored, if batch mode was enabled.

#3 - 02/04/2016 07:49 AM - Greg Shah

A useful email discussion:

-------- Forwarded Message --------
Subject: Re: batch issue?
Date: Thu, 4 Feb 2016 02:03:28 +0300
From: Eugenie V. Lyzenko ...
To: Greg Shah ..., Ovidiu Maxiniuc ...
CC: Asofiei, Constantin ...

Greg,

        The message display statement must be redirected STDOUT aware.

        We don't want to make each output statement aware of this. As far as I know, all output would behave the same way.

        The code causes exception runs before TC detects if the STDOUT already redirected(using TC.isRedirected()).

The issue in #2637 is that TC.isRedirected() will return FALSE when the user runs the batch client like this:

java .... ClientDriver .... > my_log_file 2&>1

ThinClient has no idea that the user (or a script) has already redirected stdout/stderr. So it acts like redirection did not occur since the 4GL code did not do the redirection. In OpenEdge, this problem doesn't occur.

I see. In this case the P2J currently has no idea is the output is redirected because redirection is performed by external OS tools, I guess. And this means TC.isRedirected() returns FALSE.

I don't know if the #2637 issue is the issue that Ovidiu is seeing.

Yes, we need to know what is the client execution command syntax. If redirection made by OS - the issue is different.

To be able to know about such redirection for particular session we need to implement some call to OS to find out the current STDOUT redirection status. As far as I know this is not implemented in current P2J.

Greg

Regards,
Eugenie.

On 02/03/2016 05:22 PM, Eugenie V. Lyzenko wrote:

Greg,

Also please see #2637 which is a known issue. The idea is that in the 4GL, it can detect if STDOUT is already redirected but P2J cannot. It is a bug to fix. I'm not sure it is the cause of this.

Yes, I think it is related issue. The message display statement must be redirected STDOUT aware. The code causes exception runs before TC detects if the STDOUT already redirected(using TC.isRedirected()).

However we need to confirm the 4GL is able to display message in redirected mode. Because the 4GL can silently ignore this cal for redirected mode. Or at least for some OS in some conditions(redirected batch mode).

Greg

Regards,
Eugenie.

On 02/03/2016 04:11 PM, Eugenie V. Lyzenko wrote:

Greg, Ovidiu,

I think this is expected to happen when a batch program tries to write to STDOUT and STDOUT is not redirected.

Eugenie: what do you think?

Yes, the console is not redirected and system is Linux or Windows with foreground batch run.

Thanks,
Greg

Regards,

Eugenie.

On 02/03/2016 02:28 PM, Ovidiu Maxiniuc wrote:

Hi,

A quick question, I hope.
I am testing #2794:
com.something.generatedTest.EntityMethodAPITest.runTest[Entity=|+VOID| TestType=|METHOD|
TestSubType=|RECORD_DEPENDENT| xmlSuffix=|0|]

with freshly generated code. I am getting some exceptions caused by this:

Caused by: com.goldencode.p2j.util.UnstoppableExitException
    at com.goldencode.p2j.ui.client.chui.driver.batch.BatchHelper.outputOperation(BatchHelper.java:62)
    at com.goldencode.p2j.ui.client.chui.driver.batch.BatchPrimitives.updateScreen(BatchPrimitives.java:238)
    at com.goldencode.p2j.ui.chui.DirectPrimitives.updateScreen(DirectPrimitives.java:70)
    at com.goldencode.p2j.ui.chui.DoubleBufferedTerminal.sync(DoubleBufferedTerminal.java:232)
    at com.goldencode.p2j.ui.client.OutputManager.sync(OutputManager.java:932)
    at com.goldencode.p2j.ui.client.widget.AbstractWidget.sync(AbstractWidget.java:1659)
    at com.goldencode.p2j.ui.client.chui.MessageAreaImpl.setMessageText(MessageAreaImpl.java:331)
    at com.goldencode.p2j.ui.client.chui.MessageAreaImpl.setMessageText(MessageAreaImpl.java:238)
    at com.goldencode.p2j.ui.client.Window.setMessageText(Window.java:1447)
    at com.goldencode.p2j.ui.client.Window.displayMessage(Window.java:2425)
    at com.goldencode.p2j.ui.client.Window.message(Window.java:1881)
    at com.goldencode.p2j.ui.client.Window.message(Window.java:1815)
    at com.goldencode.p2j.ui.client.Window.splitMessage(Window.java:2462)
    at com.goldencode.p2j.ui.client.Window.message(Window.java:1850)
    at com.goldencode.p2j.ui.client.Window.message(Window.java:2020)
    at com.goldencode.p2j.ui.chui.ThinClient.message(ThinClient.java:6745)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:497)
    at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
    at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:705)
    at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
    at com.goldencode.p2j.net.Conversation.waitMessage(Conversation.java:257)
    at com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1128)
    at com.goldencode.p2j.net.Queue.transact(Queue.java:585)
    at com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:223)
    at com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:163)
    at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1425)
    at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:97)
    at com.sun.proxy.$Proxy5.standardEntry(Unknown Source)
    at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:281)
    at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:102)
    at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:201)
    at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:398)
    at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:95)
    at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:267)

What do I do wrong? is it a configuration issue ?

Thanks,
Ovidiu

#4 - 02/04/2016 07:50 AM - Greg Shah

To be able to know about such redirection for particular session we need to implement some call to OS to find out the current STDOUT redirection status. As far as I know this is not implemented in current P2J.

Correct. Fixing that problem is the purpose of this task.