User Interface - Feature #1783

properly handle empty UNDERLINE statements

10/30/2012 09:34 AM - Greg Shah

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Eugenie Lyzenko	**% Done:**	100%
Category:		**Estimated time:**	16.00 hours
Target version:	GUI Support for a Complex ADM2 App		
billable:	No	**vendor_id:**	GCD

Description

History

#1 - 10/31/2012 02:56 PM - Greg Shah

- Target version set to Milestone 12

#2 - 05/06/2014 02:42 PM - Greg Shah

- Assignee set to Eugenie Lyzenko

- Estimated time changed from 4.00 to 16.00

The UNDERLINE statement can be specified like this:

```
...
DEFINE FRAME my-frame my-var1 my-var2.

UNDERLINE WITH FRAME my-frame 5 DOWN.
```

Normally, one or more widgets are specified like this:

```
UNDERLINE my-var2 WITH FRAME my-frame.
```

When these widgets are not listed, we call this an "empty" underline statement. I believe this has no runtime effect (nothing gets underlined) and it can only really affect the static frame definition (the frame phrase can have unique options that aren't specified elsewhere). Please test this out and make changes that ensure that empty underline statements don't emit into the business logic (assuming they in fact have no runtime effect).

#3 - 05/07/2014 01:38 PM - Eugenie Lyzenko

The current P2J code converts

```
...
UNDERLINE WITH FRAME my-frame 5 DOWN.
...
```

into

```
...
            myFrameFrame.openScope();
            frame0.openScope();

            ;
...
```

This is correct or not? The converted code should not contain NOP(;)? The UI frame generated is:

```
...
   public static class UnderlineTest0MyFrameDef
   extends WidgetList
   {
      FillInWidget myVar1 = new FillInWidget();

      FillInWidget myVar2 = new FillInWidget();

      public void setup(CommonFrame frame)
      {
         frame.setDown(5);
         myVar1.setDataType("character");
         myVar2.setDataType("integer");
         myVar1.setLabel("my-var1");
         myVar1.setFormat("x(8)");
         myVar2.setLabel("my-var2");
      }

      {
         addWidget("myVar1", "my-var1", myVar1);
         addWidget("myVar2", "my-var2", myVar2);
      }
   }
...
```

Just no underline references. The converted application does not have any underlining.

The second question is for

```
UNDERLINE my-var2 WITH FRAME my-frame.
```

statement. Do we need to debug it to verify the same behavior as in 4GL? The 4GL shows the frame with underlined entry:

```
...
+--------------------+
|my-var1     my-var2|
|-------- ----------|
|         ----------|
|                    |
|                    |
|                    |
|                    |
+--------------------+
...
```

Note there is no value and underline attribute is placed to the same cell as data.

#4 - 05/07/2014 02:01 PM - Greg Shah

> This is correct or not?

It is not 100% correct.

> The converted code should not contain NOP?

Right. It should not contain that.

Make sure that this testcase works the same way as in the 4GL:

```
def var txt as char init "something".
define frame f0 txt.
underline with frame f0 5 down.
pause.
display txt with frame f0.
```

> Do we need to debug it to verify the same behavior as in 4GL? The 4GL shows the frame with underlined entry:

It should already be working, but please do check it. If there are problems, document them here and fix them.

#5 - 05/07/2014 03:17 PM - Eugenie Lyzenko

```
...
underline with frame f0 5 down.
...
```

This line converts to:

```
...
;
...
```

However the behavior is the same(NOP just do nothing).

#6 - 05/07/2014 03:18 PM - Greg Shah

Is the behavior (especially AFTER the pause) the same as the 4GL?

#7 - 05/07/2014 04:03 PM - Eugenie Lyzenko

> Is the behavior (especially AFTER the pause) the same as the 4GL?

Yes, P2J does exactly the same as 4GL.

#8 - 05/07/2014 05:59 PM - Greg Shah

Good. Then the important thing to do is to eliminate the output of that ;. Take a close look at the AST and JAST files for this case. Identify the node that is being generated in the JAST that causes the ;. Track that back to it's "peer" in the AST. Then put the necessary protection rules in the TRPL ruleset(s) to detect this "empty" case and avoid emiting that JAST node.

#9 - 05/07/2014 08:10 PM - Eugenie Lyzenko

- File evl_upd20140507a.zip added

The jast node to be eliminated is:

```
...
                      <ast col="0" id="5746666242091" line="0" text="underline" type="METHOD_CALL">
                        <annotation datatype="java.lang.Long" key="peerid" value="5742371274781"/>
                      </ast>
...
```

Ast peer is:

```
...
      <ast col="1" id="5742371274781" line="3" text="underline" type="KW_UNDERLIN">
        <annotation datatype="java.lang.Long" key="scope-id" value="5742371274817"/>
```

```
        <annotation datatype="java.lang.Long" key="frame-id" value="5742371274818"/>
        <annotation datatype="java.lang.Long" key="block_par_id" value="5746666242072"/>
        <annotation datatype="java.lang.Long" key="peerid" value="5746666242091"/>
        <ast col="11" hidden="true" id="5742371274784" line="3" text="with" type="FRAME_PHRASE">
          <annotation datatype="java.lang.Long" key="frame-id" value="5742371274818"/>
          <ast col="16" hidden="true" id="5742371274787" line="3" text="frame" type="KW_FRAME">
            <ast col="22" id="5742371274790" line="3" text="f0" type="WID_FRAME"/>
          </ast>
          <ast col="27" id="5742371274793" line="3" text="down" type="KW_DOWN">
            <ast col="0" id="5742371274794" line="0" text="expression" type="EXPRESSION">
              <ast col="25" id="5742371274795" line="3" text="5" type="NUM_LITERAL">
                <annotation datatype="java.lang.Boolean" key="use64bit" value="false"/>
              </ast>
            </ast>
          </ast>
        </ast>
      </ast>
...
```

So I would suggest to modify convert/ui_statements.rules line 962 this way:

```
...
         <!-- underline -->
         <rule>type == prog.kw_underlin and !descendant(prog.frame_phrase, 1)
            <action>methodText = "underline"</action>

            <!-- if there is only a single child, then the instance method
                 name above will be used, find the root for the field list
                 -->
            <action>
               ref = this.getImmediateChild(prog.content_array, null)
            </action>

            <!-- check if we use the frame form or not -->
            <rule>ref.numImmediateChildren > 1
               <action>
                  refName = #(java.lang.String) execLib("get_framename", this)
               </action>
            </rule>
         </rule>
...
```

Additional condition check(not having frame phrase just after underline statement) removes JAST node and eliminates NOP in converted code.

#10 - 05/08/2014 08:17 AM - Greg Shah

Your solution is not correct. The 2nd parameter to descendant() is the number of levels of children (1 for the immediate children, 2 to include grandchildren...) to search. With your approach, all UNDERLINE statements that have a frame phrase will never emit.

It is also possible to have no children at all (this is also a NOP), as in:

```
UNDERLINE.
```

Try this:

```
<!-- avoid emitting code for empty underline statements (those with no widgets listed) -->
<rule>type == prog.kw_underlin and copy.numImmediateChildren &gt; 0 and copy.getChildAt(0).type != prog.frame_
phrase
...
```

#11 - 05/08/2014 01:05 PM - Eugenie Lyzenko

- File evl_upd20140508a.zip added

```
    <rule>type == prog.kw_underlin and copy.numImmediateChildren &gt; 0 and copy.getChildAt(0).type != prog.fr
    ame_phrase
```

This works OK, including eliminating NOP generation for:

```
UNDERLINE.
```

Also the testcase:

```
...
UNDERLINE my-var2 WITH FRAME my-frame.
...
```

converted to:

```
...
myFrameFrame.widgetMyVar1().underline();
...
```

#12 - 05/08/2014 01:16 PM - Greg Shah

Code Review 0508a

I am fine with the change.

Are all the testcases working the same way in the 4GL and P2J? If so, then go ahead and run conversion regression testing. There is no need for runtime regression testing.

#13 - 05/08/2014 02:02 PM - Eugenie Lyzenko

- File testcases_evl_20140508a.zip added

Unfortunately there are the difference in handling "real" underline statements like(underline_test2.p):

```
...
def var my-var1 as character initial "Hello".
def var my-var2 as integer initial 314.

DEFINE FRAME my-frame my-var1 my-var2.
UNDERLINE WITH FRAME my-frame 5 DOWN.

UNDERLINE my-var2 WITH FRAME my-frame.
DISPLAY my-var1 my-var2 WITH FRAME my-frame.

pause.

UNDERLINE my-var1 WITH FRAME my-frame.
DISPLAY my-var1 my-var2 WITH FRAME my-frame.
...
```

I have attached the screenshots to compare 4Gl and P2J outputs. The pictures *_0_4gl.jpg and *_0_p2j.jpg reflects the state before pause, *_1_4gl.jpg and *_1_p2j.jpg - after pause. You can see the difference. Is this the issue we need to fix?

#14 - 05/08/2014 02:25 PM - Greg Shah

Yes, fix this.

I suspect this has something to do with the DOWN processing and the afterDown variable used in GenericFrame.underline(). The UNDERLINE is like an implicit DOWN, so it should cause some special behavior (like making the DISPLAY appear in the next iteration of the down frame). DOWN processing also causes pending output to be flushed. But normally on an interactive terminal (not redirected), flushing is not an issue.

I think you must carefully debug into the first DISPLAY statement (the missing one) to see where it "goes wrong".

#15 - 05/11/2014 05:00 PM - Eugenie Lyzenko

The intermediate debugging results. The main issue is happening on the first UNDERLINE statement when there is nothing yet on the screen.

The control flow for not first underline and display is:
When underline processed it turn on frame repainting. The FrameImpl code looks for the underlined row and draws the horizontal line. Then when handling display statement FrameImpl code look for the down body and displays the widget content. So we have two separate drawing code and display command draw only widget values.

But on the first time when there is nothing yet on the screen the underline command does not make any output on the screen because there is no bitmap to render to. Then when handling display command FrameImpl code looks for the first objects to draw and these objects are the underlining, not the widget values. So in this case display command behaves as usual underline command and we need second painting to display the widget values after underlining.

Continue debugging.

#16 - 05/13/2014 03:10 PM - Greg Shah

Do you have an update on this? It seems like you should be pretty close to a solution, based on your findings.

#17 - 05/13/2014 03:50 PM - Eugenie Lyzenko

> Do you have an update on this? It seems like you should be pretty close to a solution, based on your findings.

Yes, I.m pretty close but still not have code that works. Looks like we need very special one time fix for cases when underline is called when frame is not on the screen. Unfortunately at this time the body of the down frame has no correct widget values, so adjustDownFrame() with underline does not work. I'm looking for solution to handle this situation on the first display call after this special underline. Hope to find solution later today(the code for frame repainting has a lot of very special processing I need to take into account to avoid regressions).

#18 - 05/14/2014 06:51 PM - Eugenie Lyzenko

Finally I have the fix. The client side modification is not enough. We need to perform implicit down processing on the server side even if there is no frame on the screen during update command. Otherwise the data sent to client are not sufficient. And the client side needs to consider this special case additionally. I'll prepare and upload the update in a hour.

#19 - 05/14/2014 08:10 PM - Eugenie Lyzenko

- File evl_upd20140514a.zip added

The suggested fix for you to review merged with the recent code base. The idea is to make full down processing on the server side and adjust the position on the client side when underline was made when frame is not yet visible.

#20 - 05/15/2014 10:23 AM - Greg Shah

Code Review 0514a

The changes look good. I do wonder what happens if the Point.y value of the widget becomes negative during relocateComponent(). Are there any bad results that could happen?

#21 - 05/15/2014 10:41 AM - Eugenie Lyzenko

> I do wonder what happens if the Point.y value of the widget becomes negative during relocateComponent(). Are there any bad results that could happen?

I'm expecting the changes I made will work only for down frames that will have underlined widgets when the frame is not on the screen. I guess in usual case(when the frame is already visible) this special processing should be bypassed. But I agree, the negative y coords can produce screen mess or the widget will not be seen because location is out of the viewport rectangles. May be the changes need to be tested additionally.

#22 - 05/15/2014 10:57 AM - Greg Shah

> can produce screen mess

If our clipping implementation is working properly, then a negative y value should not be a problem. However, there may be a flaw in our logic somewhere.

> or the widget will not be seen because location is out of the viewport rectangles

Yes, perhaps this is an issue if the "current" row is ever able to be moved to a location that is invalid.

My primary concerns:

1. Can an unexpected y value (especially a negative one) cause problems in our code?
2. Can the general approach lead to unexpected state that breaks code currently working? In particular, can the "current" row (which is the row that can be edited and which should always be visible) ever be in a condition (moved to a location) which is not displayable in the down frame?

I think the best way to check this is to try to create testcases:

- that have the first row moved to an invalid y value
- that have the current row moved to an invalid y value

In other words: try to break this implementation. Report your results here. If you cannot break it, then go ahead into both conversion and runtime testing.

#23 - 05/15/2014 11:14 AM - Eugenie Lyzenko

> In other words: try to break this implementation. Report your results here. If you cannot break it, then go ahead into both conversion and runtime testing.

OK.

#24 - 05/15/2014 06:06 PM - Eugenie Lyzenko

I have tested two possibilities:
- perform UP command for the first row in down frame several times.
- manual setting widget row attribute to negative value.

The result is:
- it is not possible to set row to negative by the UP command - 4GL moves the widget to the bottom of the frame when ~~1 coord is requested.~~
it is not possible to manually move the widget to negative position by changing the widget's ROW attribute - the 4GL displays the message and coord is not changing.

The P2J works almost the same with this testing. UP command moves the first row to the bottom of the frame and manual setting row to negative does not change anything. The only exception is P2J does not display the warning message that the Y coord can not be negative while 4GL displays the message. But I think this in not related to the current update changes. Or this is something we need to fix with this task too?

So I'm going to perform one more test involving the code that underlines the hidden widgets. This way we can be sure the modified code will be included into the process.

#25 - 05/15/2014 08:47 PM - Eugenie Lyzenko

Well, the testcase to partially break implementation:

```
...
def var my-var1 as character initial "Hello".
def var my-var2 as integer initial 314.

DEFINE FRAME my-frame my-var1 my-var2.
UNDERLINE WITH FRAME my-frame 3 DOWN.

UNDERLINE my-var1 WITH FRAME my-frame.
DISPLAY my-var1 my-var2 WITH FRAME my-frame.
message "press any key to one line up.".
pause.

UP 1 WITH FRAME my-frame.
DISPLAY my-var1 my-var2 WITH FRAME my-frame.

message "press any key to two lines up.".
pause.
```

```
UP 2 WITH FRAME my-frame.
DISPLAY my-var1 my-var2 WITH FRAME my-frame.

message "press any key to one line up.".
pause.

UP 1 WITH FRAME my-frame.
DISPLAY my-var1 my-var2 WITH FRAME my-frame.

message "press any key to set row to -7 and -5.".
pause.

my-var1:row = -7.
my-var2:row = -5.
DISPLAY my-var1 my-var2 WITH FRAME my-frame.
...
```

The handling of this test differs in 4GL and P2J with last changes and without(when first line underlining does not work properly). Does it mean the changes we are going to test has side effects? Moving UP the first row in 4GL does not change anything while in P2J it clears underlining and shifts underlined row one line UP(because the real widget values was on the second row).

The main issue I think is the P2J represents underline character as one of the element in down body(along with the widget itself). So when we have first line as underlined - we really have down body with two elements, the first is the underline and the second is the widget line. And if we moving them up - everything is moving I think. One I can tell is the negative coord is not set.

What do you think? Do we need to find and fix all underline deviations in this task? May be there is some subset of cases we need to provide the compatibility? I'm asking to just not to fix anything we will never encounter. What is our policy here?

#26 - 05/16/2014 08:30 AM - Greg Shah

> The only exception is P2J does not display the warning message that the Y coord can not be negative while 4GL displays the message.

Please create a small testcase that shows this problem and open a new redmine task. Make sure to properly describe the problem and attach the testcase.

> Moving UP the first row in 4GL does not change anything while in P2J it clears underlining and shifts underlined row one line UP.

Create a new redmine task for this problem, you can attach the testcase from note 25 as the recreate.

> Do we need to find and fix all underline deviations in this task?

We will try to fix the problems with UNDERLINE. But the two problems above are more about UP than UNDERLINE, so we will defer those.

> May be there is some subset of cases we need to provide the compatibility? I'm asking to just not to fix anything we will never encounter. What is our policy here?

As a general policy, we don't put limits on our compatibility. However, during the work on any given task we may choose to defer some work/bugs that are not immediately critical.

The primary issue here is that you are not really exercising the new code very much. Try something like this:

```
def var my-var1 as character initial "Hello".
def var my-var2 as integer initial 314.

DEFINE FRAME my-frame my-var1 my-var2.
UNDERLINE my-var1 WITH FRAME my-frame.
UNDERLINE my-var1 WITH FRAME my-frame.
UNDERLINE my-var1 WITH FRAME my-frame.
UNDERLINE my-var1 WITH FRAME my-frame.
UNDERLINE my-var1 WITH FRAME my-frame.
DISPLAY my-var1 my-var2 WITH FRAME my-frame.
```

Or this:

```
def var my-var1 as character initial "Hello".
def var my-var2 as integer initial 314.

DEFINE FRAME my-frame my-var1 my-var2.
UNDERLINE my-var1 WITH FRAME my-frame.
VIEW FRAME my-frame.
HIDE FRAME my-frame.
UNDERLINE my-var1 WITH FRAME my-frame.
VIEW FRAME my-frame.
HIDE FRAME my-frame.
UNDERLINE my-var1 WITH FRAME my-frame.
VIEW FRAME my-frame.
HIDE FRAME my-frame.
UNDERLINE my-var1 WITH FRAME my-frame.
VIEW FRAME my-frame.
HIDE FRAME my-frame.
UNDERLINE my-var1 WITH FRAME my-frame.
DISPLAY my-var1 my-var2 WITH FRAME my-frame.
```

Both of these are designed to trigger your code more than once, in an attempt to break it.

#27 - 05/16/2014 10:48 AM - Eugenie Lyzenko

Greg Shah wrote:
...

> The primary issue here is that you are not really exercising the new code very much. Try something like this:
>
> [...]
>
> Or this:
>
> [...]
>
> Both of these are designed to trigger your code more than once, in an attempt to break it.

I have tested these cases with the my changes for this task. The screen looks are absolutely the same for 4GL and P2J. So we can start the regression testing and fix the problems with UP(discovered in note 25) as another task. What do you think?

#28 - 05/16/2014 11:47 AM - Greg Shah

> So we can start the regression testing and fix the problems with UP as another task. What do you think?

Yes.

#29 - 05/16/2014 02:30 PM - Eugenie Lyzenko

The conversion test completed. The converted sources are binary equal. Continue with runtime regression testing.

#30 - 05/16/2014 07:52 PM - Eugenie Lyzenko

- File evl_upd20140516a.zip added

The regression has been found. Testcase TC-ITEM-MASTER-103 the difference in page size for report generation. The expected file:

```
...
Item            Description                              Unit Cost      On Hand      Unit Cost
--------------- ---------------------------------------- -------------- ------------ --------------
                                                                                      --------------
                                                                                           0.00000
...
```

but we have:

```
...
Item            Description                                Unit Cost      On Hand      Unit Cost
--------------- ---------------------------------------- -------------- ------------ --------------
                                                                                          0.00000
...
```

The underline processing change should be bypassed for redirected output because in this case the rendering is different. The suggested fix has been attached for you to review. I'll restart the runtime regression soon. Please let me know if there are any notes/objections.

#31 - 05/16/2014 08:35 PM - Eugenie Lyzenko

- File deleted (evl_upd20140516a.zip)

#32 - 05/16/2014 08:39 PM - Eugenie Lyzenko

- File evl_upd20140516a.zip added

Sorry, the suggested fix does not work. Need to check ThinClient redirection, not simple Frame one. The fix has been re-uploaded for you to review.

#33 - 05/17/2014 11:07 PM - Eugenie Lyzenko

The main part completed without regressions in a single run. Continue testing with CTRL-C part.

#34 - 05/19/2014 08:20 AM - Greg Shah

Code Review 0516a

I am fine with the changes. When CTRL-C processing passes, please check it in and distribute it.

#35 - 05/21/2014 01:56 PM - Eugenie Lyzenko

Finally I have finished CTRL-C testing with all passed results. Not in single run.due to multiple false negative results. The problem is well known issue with:

```
...
|        ATT+---------------------- Message -----------------------+061A
|        P.O| An error has occurred attempting to connect to the   |
|        Wil|              selected database system.               |
|           | ------------------------------------------------- |
...
```

It is strange because I can not reproduce this manually. I have stopped the servers and uploaded the results to shared dir:
10532_d2814f6_20140521_evl.zip

So shortly the update will be committed and distributed.

#36 - 05/21/2014 02:15 PM - Eugenie Lyzenko

The update 0516a has been committed to bzr as 10536.

#37 - 05/21/2014 02:23 PM - Greg Shah

- % Done changed from 0 to 100

- Status changed from New to Closed

#38 - 11/16/2016 12:12 PM - Greg Shah

- Target version changed from Milestone 12 to GUI Support for a Complex ADM2 App

Files

evl_upd20140507a.zip	20.1 KB	05/08/2014	Eugenie Lyzenko
evl_upd20140508a.zip	20.2 KB	05/08/2014	Eugenie Lyzenko
testcases_evl_20140508a.zip	44.3 KB	05/08/2014	Eugenie Lyzenko
evl_upd20140514a.zip	237 KB	05/15/2014	Eugenie Lyzenko
evl_upd20140516a.zip	237 KB	05/17/2014	Eugenie Lyzenko