

Base Language - Bug #2300

:: operator used in UI statements

05/09/2014 03:41 AM - Ovidiu Maxiniuc

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			
Related issues:			
Related to Base Language - Feature #2050: ATTR_POLY/METH_POLY used as rvalue ...			Closed

History

#1 - 05/09/2014 04:15 AM - Ovidiu Maxiniuc

The conversion error is revealed by the following simple testcase:

```
def var hb as handle.
hb = buffer book:handle.
display hb::book-id hb::isbn.
```

When conversion is run the following output is generated:

```
EXPRESSION EXECUTION ERROR:
-----
bdt = create(cls)
      ^ { java.lang.InstantiationException }
-----
EXPRESSION EXECUTION ERROR:
-----
fmt = execLib("format_for_type", ref)
      ^ { Expression execution error @1:7 }
-----
EXPRESSION EXECUTION ERROR:
-----
fmtst1 = execLib("get_override_format_string", copy)
          ^ { Expression execution error @1:7 }
-----
ERROR:
java.lang.RuntimeException: ERROR!  Active Rule:
-----
      RULE REPORT
-----
Rule Type :    WALK
Source AST:  [ expression ] BLOCK/STATEMENT/KW_DISP/EXPRESSION/ @0:0 {639950127300}
Copy AST   :  [ expression ] BLOCK/STATEMENT/KW_DISP/EXPRESSION/ @0:0 {639950127300}
Condition  :  bdt = create(cls)
Loop       :  false
--- END RULE REPORT ---

at com.goldencode.p2j.pattern.PatternEngine.run(PatternEngine.java:1004)
at com.goldencode.p2j.convert.ConversionDriver.processTrees(ConversionDriver.java:915)
at com.goldencode.p2j.convert.ConversionDriver.generateFrames(ConversionDriver.java:845)
at com.goldencode.p2j.convert.ConversionDriver.back(ConversionDriver.java:810)
at com.goldencode.p2j.convert.ConversionDriver.main(ConversionDriver.java:1740)
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:601)
    at com.intellij.rt.execution.application.AppMain.main(AppMain.java:134)
Caused by: com.goldencode.expr.ExpressionException: Expression execution error @1:10 [EXPRESSION id=6399501273
00]
    at com.goldencode.p2j.pattern.AstWalker.walk(AstWalker.java:225)
    at com.goldencode.p2j.pattern.AstWalker.walk(AstWalker.java:160)
    at com.goldencode.p2j.pattern.PatternEngine.apply(PatternEngine.java:1500)
    at com.goldencode.p2j.pattern.PatternEngine.processAst(PatternEngine.java:1398)
    at com.goldencode.p2j.pattern.PatternEngine.processAst(PatternEngine.java:1346)
    at com.goldencode.p2j.pattern.PatternEngine.run(PatternEngine.java:972)
    ... 9 more
Caused by: com.goldencode.expr.ExpressionException: Expression execution error @1:10
    at com.goldencode.expr.Expression.execute(Expression.java:434)
    at com.goldencode.p2j.pattern.Rule.apply(Rule.java:401)
    at com.goldencode.p2j.pattern.Rule.executeActions(Rule.java:640)
    at com.goldencode.p2j.pattern.Rule.coreProcessing(Rule.java:609)
    at com.goldencode.p2j.pattern.Rule.apply(Rule.java:440)
Elapsed job time: 00:00:00.119
    at com.goldencode.p2j.pattern.RuleContainer.apply(RuleContainer.java:530)
    at com.goldencode.p2j.pattern.RuleSet.apply(RuleSet.java:1)
    at com.goldencode.p2j.pattern.AstWalker.walk(AstWalker.java:212)
    ... 14 more
Caused by: com.goldencode.expr.ExpressionException: Expression execution error @1:7
    at com.goldencode.expr.Expression.execute(Expression.java:434)
    at com.goldencode.p2j.pattern.Rule.apply(Rule.java:401)
    at com.goldencode.p2j.pattern.Rule.executeActions(Rule.java:640)
    at com.goldencode.p2j.pattern.Rule.coreProcessing(Rule.java:609)
    at com.goldencode.p2j.pattern.Rule.apply(Rule.java:440)
    at com.goldencode.p2j.pattern.Rule.executeActions(Rule.java:640)
    at com.goldencode.p2j.pattern.Rule.coreProcessing(Rule.java:609)
    at com.goldencode.p2j.pattern.Rule.apply(Rule.java:440)
    at com.goldencode.p2j.pattern.NamedFunction.execute(NamedFunction.java:393)
    at com.goldencode.p2j.pattern.AstSymbolResolver.execute(AstSymbolResolver.java:655)
    at com.goldencode.p2j.pattern.CommonAstSupport$Library.execLib(CommonAstSupport.java:1097)
    at com.goldencode.expr.CE5605.execute(Unknown Source)
    at com.goldencode.expr.Expression.execute(Expression.java:341)
    ... 21 more
Caused by: com.goldencode.expr.ExpressionException: Expression execution error @1:7
    at com.goldencode.expr.Expression.execute(Expression.java:434)
    at com.goldencode.p2j.pattern.Rule.apply(Rule.java:401)
    at com.goldencode.p2j.pattern.NamedFunction.execute(NamedFunction.java:393)
    at com.goldencode.p2j.pattern.AstSymbolResolver.execute(AstSymbolResolver.java:655)
    at com.goldencode.p2j.pattern.CommonAstSupport$Library.execLib(CommonAstSupport.java:1097)
    at com.goldencode.expr.CE5624.execute(Unknown Source)
    at com.goldencode.expr.Expression.execute(Expression.java:341)
    ... 33 more
Caused by: java.lang.InstantiationException
    at sun.reflect.InstantiationExceptionConstructorAccessorImpl.newInstance(InstantiationExceptionConstructor
AccessorImpl.java:48)
    at java.lang.reflect.Constructor.newInstance(Constructor.java:525)
    at com.goldencode.p2j.pattern.CommonAstSupport$Library.create(CommonAstSupport.java:589)
    at com.goldencode.p2j.pattern.CommonAstSupport$Library.create(CommonAstSupport.java:512)
    at com.goldencode.expr.CE5631.execute(Unknown Source)
    at com.goldencode.expr.Expression.execute(Expression.java:341)

```

This is caused when in the ExpressionConversionWorker.java, "BaseDataType" is returned when heuristic detection of an :: expression fails so the frame definition cannot extract the format for this generic datatype.

The main question here is: as the frame definition is computed at compile time in 4GL, how does 4GL handle cases when the :: operator is used? And, can it be used in update statements? Some little testing on lndev01 abends the 4GL with "Core Dump" message when trying to edit it, so a first impression is that they didn't design 4GL to use the :: operator in UI update statements. The test which abends in 4GL when you try to change the h::f1 value is:

```

def temp-table ttl field f1 as int.
create ttl.
ttl.f1 = 10.

def var h as handle.
h = buffer ttl:handle.

display h::f1 ttl.f1 with frame f1.
enable all with frame f1.

```

wait-for close of current-window.

Continuing the investigation on windev01. The testcase did not always crash the application, however here are some new and maybe interesting findings:

- attempting to modify the h::f1 widget content will display following Error messages:  
SYSTEM ERROR: fmpmnt - invalid data type. (62) message on CHUI client when editing a character field.  
When editing f1 int field, the app did not crashed, but any key except cursor were ignored.  
Incompatible datatypes found during runtime conversion. (5729) in an error dialog on GUI client. No crashes here.  
The messages were the same for a second character field f2 (my assumption was that character datatype allows any change).
- I added the equivalent  
h:BUFFER-FIELD("f1"):BUFFER-VALUE()  
to display statement. As expected, the same behavior as above.
- the 100-point test:
  - created function get-f2:  
function get-f2 return char : return tt1.f2. end.
  - added a dynamic call to it in same display statement:  
display DYNAMIC-FUNCTION("get-f2") h:BUFFER-FIELD("f2"):BUFFER-VALUE() h::f2 tt1.f1 with frame f1.The same behavior, again, on GUI and CHUI. Calling the function directly allows modification of the displayed value.

#### **Preliminary conclusions**

1. The crash is not specific to :: operator, but related to 4GL poly datatype.
2. Result of :: is only a 'rvalue' with unknown type, not a reference to a location to store the result and it does not matter what the changes are.
3. This kind of rvalues widgets should not be enabled for editing. It doesn't make sense and probably OpenEdge is built with the assumption that P4GL programs won't do it.

#### **#2 - 05/09/2014 08:10 AM - Greg Shah**

Was this found in standalone testcases or does this actually occur in customer code?

#### **#3 - 05/09/2014 08:19 AM - Ovidiu Maxiniuc**

It was a standalone case I used for investigation of some other issue.

The temporary fix was the cause of the regression in the customer's code mentioned in yesterday's mail discussion.