

## Database - Bug #2093

### VALEXP support in P2J

03/15/2013 11:57 AM - Ovidiu Maxiniuc

<b>Status:</b>	Closed	<b>Start date:</b>	03/15/2013
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Conversion Support for Server Features	<b>case_num:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			

### History

#### #1 - 03/15/2013 12:06 PM - Ovidiu Maxiniuc

This issue is dedicated to note 80 from Issue #2068.

There is this validation expression for the dvofrstst.dvofrstst-cde field: VALEXP "CAN-FIND(dvofrstst WHERE dvofrstst.dvofrstst-cde = dvofrstst.dvofrstst-cde)" P2J can't seem to convert CAN-FIND in the VALEXP, as the resulted validation is:

```
public logical validateExpression(BaseDataType _newValue_)
{
    final character _dvofrststCde = (character) _newValue_;
    return _dvofrststCde; /* compile error here */
}
```

Later addition (Constantin, note 124):

The above documentation is wrong, and the real VALEXP in the .df file is wrong (the real VALEXP for dvofrstst.dvofrstst-cde is VALEXP("dvofrstst.dvofrstst-cde NE = "" "")), which is corrupted.

Anyway, 4GL doesn't fail on this I think because the field is not used in a UI-update, is used for display only. So, as this doesn't even need to get generated as a Validation inner class, I've instructed Ovidiu to work on this:

- do not emit Validation inner classes in cases the field is not used in interactive UI statements.
- search for cases where P2J doesn't emit the Validation inner class, when VALEXP is used at the field.

## #2 - 03/15/2013 12:42 PM - Eric Faulhaber

- Parent task deleted (#2068)

- Project changed from Liberty to Database

## #3 - 03/15/2013 12:43 PM - Eric Faulhaber

- Target version set to Milestone 4

## #4 - 03/19/2013 04:03 AM - Constantin Asofiei

Ovidiu: with the [#1620](#) changes, I can't see any more schema VALEXP problems in the logs (I think P2J was too aggressive in emitting the schema VALEXP for the frames used in FORM statements). We should make this a low priority. Please document here all your findings and start working on the last notes on #2068.

## #5 - 03/19/2013 07:09 AM - Ovidiu Maxiniuc

Everything seems to work smooth, validation is only generated in schema-validation:

```
<rule>( (parent.type == prog.kw_enable or
  parent.type == prog.kw_prmt_for or
  parent.type == prog.kw_set or
  parent.type == prog.kw_update or
  parent.type == prog.kw_insert or
  parent.type == prog.kw_form) and
  parent.parent.type == prog.statement) or
upPath("STATEMENT/KW_FORM/FORM_ITEM") or
upPath("STATEMENT/KW_FORM/FORM_ITEM/EXPRESSION")
<rule>evalLib("fields")
...
```

Later in @validation\_post@s rules the nodes are dispatched to corresponding frames.

Here is a special cases when P2J fails:

```
form person.schedule validate(person.schedule[1] = "a", "must type 'a'") with frame fl.
```

which will result in a

```
ref = getAst(#(long) widref.getAnnotation("refid"))
      ^ { java.lang.NullPointerException }
```

```
Rule Type : WALK
Source AST: [ validate ] BLOCK/STATEMENT/KW_FORM/FORMAT_PHRASE/KW_VALIDATE/ @34:22 {704374636766}
Copy AST : [ validate ] BLOCK/STATEMENT/KW_FORM/FORMAT_PHRASE/KW_VALIDATE/ @34:22 {704374636766}
Condition : ref = getAst(#(long) widref.getAnnotation("refid"))
Loop : false
--- END RULE REPORT ---
```

```
at com.goldencode.p2j.pattern.PatternEngine.run (PatternEngine.java:732)
at com.goldencode.p2j.convert.ConversionDriver.processTrees (ConversionDriver.java:948)
at com.goldencode.p2j.convert.ConversionDriver.back (ConversionDriver.java:836)
at com.goldencode.p2j.convert.ConversionDriver.main (ConversionDriver.java:1729)
Caused by: com.goldencode.expr.ExpressionException: Expression execution error @1:29 [KW_VALIDATE id=704374636766]
```

The AST look like this:

```
<ast col="1" id="704374636547" line="34" text="form" type="KW_FORM">
  <ast col="0" id="704374636555" line="0" text="form item" type="FORM_ITEM">
    <ast col="0" id="704374636556" line="0" text="expression" type="EXPRESSION">
      <ast col="6" id="704374636557" line="34" text="person.schedule" type="FIELD_CHAR" />
    <ast col="0" id="704374636679" line="0" text="" type="FORMAT_PHRASE">
```

```

    <ast col="0" id="704374636680" line="0" text="" type="KW_VALIDATE">
      <ast col="0" id="704374636681" line="0" text="" type="EXPRESSION" />
      <ast col="0" id="704374636688" line="0" text="" type="EXPRESSION" />
    <ast col="0" id="704374636560" line="0" text="format phrase" type="FORMAT_PHRASE">
      <ast col="22" id="704374636561" line="34" text="validate" type="KW_VALIDATE">
        <ast col="0" id="704374636563" line="0" text="expression" type="EXPRESSION" />
        <ast col="0" id="704374636571" line="0" text="expression" type="EXPRESSION" />

```

The valexp from schema and validation clause are created as ASTs at different levels but in validation.rules at about line 110, widref = copy.parent.getPrevSibling() which is assumed to be a FIELD\_CHAR with refid annotation.

There was an additional case when the generated code was not compilable because some widget accessor of an extend field from a frame was not generated. However, when writing this note I observed that it was my mistake - wrong index of extend was accessed without being added by a form statement to respective frame. In fact, this should be somehow detected and an error/warning be generated (the P4GL does this test at compile time).

#### #6 - 03/19/2013 07:20 AM - Constantin Asofiei

Please check the failed case in the previous note with the latest [#1620](#) fix, if you haven't done already so.

#### #7 - 03/19/2013 07:57 AM - Ovidiu Maxiniuc

The AST is slightly different with last revision from bzz:

```

<ast col="1" id="704374636547" line="5" text="form" type="KW_FORM">
  <ast col="0" id="704374636760" line="0" text="" type="FORM_ITEM">
    <ast col="0" id="704374636761" line="0" text="" type="EXPRESSION">
      <ast col="6" id="704374636764" line="5" text="person.schedule" type="FIELD_CHAR">
        </ast>
      </ast>
    <ast col="0" id="704374636767" line="0" text="format phrase" type="FORMAT_PHRASE">
      <ast col="22" id="704374636768" line="5" text="validate" type="KW_VALIDATE">
        <ast col="0" id="704374636769" line="0" text="expression" type="EXPRESSION">
          <ast col="50" id="704374636770" line="5" text="" type="EQUALS">
            <ast col="31" id="704374636771" line="5" text="person.schedule" type="FIELD_CHAR">
              <annotation datatype="java.lang.Long" key="refid" value="704374636777"/>
            </ast>
          </ast>
        </ast>
      </ast>
    <ast col="0" id="704374636775" line="0" text="expression" type="EXPRESSION">
      </ast>
    </ast>
  </ast>

```

The refid annotation is somewhere else as you can see from the trimmed subtree: as 704374636768 is processed (type prog.kw\_validate and parent.type prog.format\_phrase and lisNote("ignore")), widref is calculated as: copy.parent.getPrevSibling() which is FORM\_ITEM 704374636760. Anyway, that field none of its sub-nodes are not annotated with refid. The only node with this property in the region is 704374636771.

**#8 - 03/23/2013 10:14 PM - Eric Faulhaber**

- *Tracker changed from Support to Bug*

**#9 - 03/29/2013 10:36 AM - Eric Faulhaber**

Is there anything left to do on this issue or can it be closed?

**#10 - 03/29/2013 12:42 PM - Ovidiu Maxiniuc**

At this moment, with the exception of the cases where the expression from the string condition is not valid, I don't have any test-cases that do not work.

**#11 - 03/29/2013 12:43 PM - Eric Faulhaber**

- *% Done changed from 0 to 100*

- *Status changed from New to Closed*

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

- *Target version changed from Milestone 4 to Conversion Support for Server Features*