Base Language - Bug #4427

Unable to convert keyword input when used to index into an extent

11/22/2019 04:31 PM - Roger Borrello

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	100%
Category:		**Estimated time:**	16.00 hours
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	
Description			

History

#1 - 11/22/2019 04:42 PM - Roger Borrello

General

Expression execution error when there is an input statement withing the brackets of an extent which is also the parameter to another input. If you remove the first input, we are successful.

Update: Committed rules/annotations/input_builtin_prep.rules to 4207a-11367.
~~Update: This still has work to do, as the Java produced does not compile. See #note-11~~
Update: Fixed in 4207a-11349.

P2J Branch

This is being handled in Branch **4207a**

Testcase

Checked into testcase_repo:
uast/extent_subscript_using_input.p

```
def buffer mybuff for Person.

form
    mybuff.schedule[1]
    mybuff.schedule[2]
    mybuff.schedule[3]
    mybuff.schedule[4]
    mybuff.schedule[5]
    mybuff.emp-num
    with frame myframe.

find first mybuff no-error.
if not available mybuff then
  create mybuff.

do with frame myframe:
  do while true:
  update
    mybuff.schedule[1]
    mybuff.schedule[2]
    mybuff.schedule[3]
    mybuff.schedule[4]
    mybuff.schedule[5]
    mybuff.emp-num
  edit-loop:
    editing:
      readkey.
      if input schedule[input emp-num] = "" then
        message "number not in use".
      else
        message "number in use".
    end.
  end.
end.
```

Artifacts

Execution Log

```
     [java] ------------------------------------------------------------------------------
     [java] Core Code Conversion
     [java] ------------------------------------------------------------------------------
     [java] 
     [java] Optional rule set [customer_specific_conversion] not found.
     [java] ./abl/extent_subscript_using_input.p
     [java] Elapsed job time:  00:00:00.964
     [java] EXPRESSION EXECUTION ERROR:
     [java] ---------------------------
     [java] throwException(errmsg, this)
     [java] ^  { Unsupported builtin function KW_INPUT. [FUNC_POLY id <197568495865> 28:25] }
     [java] ---------------------------
     [java] ERROR:
     [java] com.goldencode.p2j.pattern.TreeWalkException: ERROR!  Active Rule:
     [java] -----------------------
     [java]       RULE REPORT      
     [java] -----------------------
     [java] Rule Type :   WALK
     [java] Source AST:  [ input ] BLOCK/INNER_BLOCK/BLOCK/INNER_BLOCK/BLOCK/STATEMENT/KW_UPDATE/EDITING_BLOCK
/BLOCK/STATEMENT/KW_IF/EXPRESSION/EQUALS/FUNC_POLY/LBRACKET/FUNC_POLY/ @28:25 {197568495865}
     [java] Copy AST  :  [ input ] BLOCK/INNER_BLOCK/BLOCK/INNER_BLOCK/BLOCK/STATEMENT/KW_UPDATE/EDITING_BLOCK
/BLOCK/STATEMENT/KW_IF/EXPRESSION/EQUALS/FUNC_POLY/LBRACKET/FUNC_POLY/ @28:25 {197568495865}
     [java] Condition :  throwException(errmsg, this)
     [java] Loop      :  false
     [java] --- END RULE REPORT ---
     [java] 
     [java] 
     [java] 
     [java]     at com.goldencode.p2j.pattern.PatternEngine.run(PatternEngine.java:1070)
     [java]     at com.goldencode.p2j.convert.TransformDriver.processTrees(TransformDriver.java:542)
     [java]     at com.goldencode.p2j.convert.ConversionDriver.back(ConversionDriver.java:580)
     [java]     at com.goldencode.p2j.convert.TransformDriver.executeJob(TransformDriver.java:876)
     [java]     at com.goldencode.p2j.convert.ConversionDriver.main(ConversionDriver.java:983)
     [java] Caused by: com.goldencode.expr.ExpressionException: Expression execution error @1:1 [FUNC_POLY id=
197568495865]
     [java]     at com.goldencode.p2j.pattern.AstWalker.walk(AstWalker.java:275)
     [java]     at com.goldencode.p2j.pattern.AstWalker.walk(AstWalker.java:210)
     [java]     at com.goldencode.p2j.pattern.PatternEngine.apply(PatternEngine.java:1633)
     [java]     at com.goldencode.p2j.pattern.PatternEngine.processAst(PatternEngine.java:1531)
     [java]     at com.goldencode.p2j.pattern.PatternEngine.processAst(PatternEngine.java:1479)
     [java]     at com.goldencode.p2j.pattern.PatternEngine.run(PatternEngine.java:1034)
     [java]     ... 4 more
     [java] Caused by: com.goldencode.expr.ExpressionException: Expression execution error @1:1
     [java]     at com.goldencode.expr.Expression.execute(Expression.java:484)
     [java]     at com.goldencode.p2j.pattern.Rule.apply(Rule.java:497)
     [java]     at com.goldencode.p2j.pattern.Rule.executeActions(Rule.java:745)
     [java]     at com.goldencode.p2j.pattern.Rule.coreProcessing(Rule.java:712)
     [java]     at com.goldencode.p2j.pattern.Rule.apply(Rule.java:534)
     [java]     at com.goldencode.p2j.pattern.Rule.executeActions(Rule.java:745)
     [java]     at com.goldencode.p2j.pattern.Rule.coreProcessing(Rule.java:712)
     [java]     at com.goldencode.p2j.pattern.Rule.apply(Rule.java:534)
     [java]     at com.goldencode.p2j.pattern.Rule.executeActions(Rule.java:745)
     [java]     at com.goldencode.p2j.pattern.Rule.coreProcessing(Rule.java:712)
     [java]     at com.goldencode.p2j.pattern.Rule.apply(Rule.java:534)
     [java]     at com.goldencode.p2j.pattern.RuleContainer.apply(RuleContainer.java:585)
     [java]     at com.goldencode.p2j.pattern.RuleSet.apply(RuleSet.java:98)
     [java]     at com.goldencode.p2j.pattern.RuleContainer.apply(RuleContainer.java:585)
     [java]     at com.goldencode.p2j.pattern.RuleSet.apply(RuleSet.java:98)
     [java]     at com.goldencode.p2j.pattern.AstWalker.walk(AstWalker.java:262)
     [java]     ... 9 more
     [java] Caused by: com.goldencode.p2j.pattern.CommonAstSupport$UserGeneratedException: Unsupported builtin
 function KW_INPUT. [FUNC_POLY id <197568495865> 28:25]
```

```
     [java]     at com.goldencode.p2j.pattern.CommonAstSupport$Library.throwException(CommonAstSupport.java:28
17)
     [java]     at com.goldencode.expr.CE12445.execute(Unknown Source)
     [java]     at com.goldencode.expr.Expression.execute(Expression.java:391)
     [java]     ... 24 more

BUILD FAILED
```

AST Snippet

```
      <ast col="7" id="197568495778" line="28" text="if" type="KW_IF">
        <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
        <ast col="0" id="197568495780" line="0" text="expression" type="EXPRESSION">
          <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
          <ast col="40" id="197568495781" line="28" text="=" type="EQUALS">
            <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
            <ast col="10" id="197568495785" line="28" text="input" type="FUNC_POLY">
              <annotation datatype="java.lang.Long" key="oldtype" value="691"/>
              <annotation datatype="java.lang.Boolean" key="builtin" value="true"/>
              <annotation datatype="java.lang.Boolean" key="returnsunknown" value="false"/>
              <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
              <annotation datatype="java.lang.Long" key="frame-id" value="197568495829"/>
              <annotation datatype="java.lang.String" key="methodtxt" value="getScheduleArray"/>
              <ast col="16" hidden="true" id="197568495787" line="28" text="schedule" type="FIELD_CHAR">
                <annotation datatype="java.lang.Long" key="oldtype" value="2781"/>
                <annotation datatype="java.lang.String" key="schemaname" value="p2j_test.person.schedule"/>
                <annotation datatype="java.lang.String" key="bufname" value="p2j_test.mybuff"/>
                <annotation datatype="java.lang.String" key="dbname" value="p2j_test"/>
                <annotation datatype="java.lang.Long" key="recordtype" value="12"/>
                <annotation datatype="java.lang.String" key="name" value="schedule"/>
                <annotation datatype="java.lang.Long" key="type" value="402"/>
                <annotation datatype="java.lang.String" key="format" value="&quot;x(64)&quot;"/>
                <annotation datatype="java.lang.Long" key="extent" value="5"/>
                <annotation datatype="java.lang.Long" key="param_index" value="0"/>
                <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
                <annotation datatype="java.lang.Boolean" key="is_meta" value="false"/>
                <annotation datatype="java.lang.String" key="fieldname" value="schedule"/>
                <annotation datatype="java.lang.String" key="bufrefkey" value="p2j_test.person,p2j_test.mybuff
,197568495627"/>
                <annotation datatype="java.lang.Long" key="bufreftype" value="18"/>
                <annotation datatype="java.lang.String" key="uniquename" value="p2j_test.mybuff_p2j_test.perso
n"/>
                <annotation datatype="java.lang.String" key="methodtxt" value="getSchedule"/>
                <annotation datatype="java.lang.String" key="check_accessors" value=""/>
                <annotation datatype="java.lang.Boolean" key="dynamicindex" value="true"/>
                <annotation datatype="java.lang.String" key="getter" value="getScheduleArray"/>
                <annotation datatype="java.lang.String" key="setter" value="setScheduleArray"/>
                <annotation datatype="java.lang.String" key="javaname" value="scheduleArray"/>
                <annotation datatype="java.lang.String" key="accessor" value="widgetScheduleArray"/>
                <annotation datatype="java.lang.String" key="widgettype" value="FillInWidget"/>
                <annotation datatype="java.lang.Long" key="frame-id" value="197568495829"/>
                <ast col="24" id="197568495788" line="28" text="[" type="LBRACKET">
                  <annotation datatype="java.lang.Boolean" key="emit" value="false"/>
                  <ast col="25" id="197568495789" line="28" text="input" type="FUNC_POLY">
                    <annotation datatype="java.lang.Long" key="oldtype" value="691"/>
                    <annotation datatype="java.lang.Boolean" key="builtin" value="true"/>
                    <annotation datatype="java.lang.Boolean" key="returnsunknown" value="false"/>
                    <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
                    <annotation datatype="java.lang.Long" key="frame-id" value="197568495829"/>
                    <annotation datatype="java.lang.String" key="methodtxt" value="getEmpNum"/>
                    <ast col="31" hidden="true" id="197568495791" line="28" text="emp-num" type="FIELD_INT">
                      <annotation datatype="java.lang.Long" key="oldtype" value="2781"/>
                      <annotation datatype="java.lang.String" key="schemaname" value="p2j_test.person.emp-num"
/>
                      <annotation datatype="java.lang.String" key="bufname" value="p2j_test.mybuff"/>
                      <annotation datatype="java.lang.String" key="dbname" value="p2j_test"/>
                      <annotation datatype="java.lang.Long" key="recordtype" value="12"/>
                      <annotation datatype="java.lang.String" key="name" value="emp-num"/>
                      <annotation datatype="java.lang.Long" key="type" value="410"/>
                      <annotation datatype="java.lang.String" key="format" value="&quot;9999999&quot;"/>
                      <annotation datatype="java.lang.String" key="columnlabel" value="&quot;Emp Num Column&qu
ot;"/>
```

```
                      <annotation datatype="java.lang.Long" key="param_index" value="0"/>
                      <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
                      <annotation datatype="java.lang.Boolean" key="is_meta" value="false"/>
                      <annotation datatype="java.lang.String" key="fieldname" value="empNum"/>
                      <annotation datatype="java.lang.String" key="bufrefkey" value="p2j_test.person,p2j_test.
mybuff,197568495627"/>
                      <annotation datatype="java.lang.Long" key="bufreftype" value="18"/>
                      <annotation datatype="java.lang.String" key="uniquename" value="p2j_test.mybuff_p2j_test
.person"/>
                      <annotation datatype="java.lang.String" key="methodtxt" value="getEmpNum"/>
                      <annotation datatype="java.lang.String" key="check_accessors" value=""/>
                      <annotation datatype="java.lang.String" key="getter" value="getEmpNum"/>
                      <annotation datatype="java.lang.String" key="setter" value="setEmpNum"/>
                      <annotation datatype="java.lang.String" key="javaname" value="empNum"/>
                      <annotation datatype="java.lang.String" key="accessor" value="widgetEmpNum"/>
                      <annotation datatype="java.lang.String" key="widgettype" value="FillInWidget"/>
                      <annotation datatype="java.lang.Long" key="frame-id" value="197568495829"/>
                    </ast>
                  </ast>
                </ast>
              </ast>
              <ast col="24" id="197568495864" line="28" text="[" type="LBRACKET">
                <annotation datatype="java.lang.Boolean" key="emit" value="false"/>
                <ast col="25" id="197568495865" line="28" text="input" type="FUNC_POLY">
                  <annotation datatype="java.lang.Long" key="oldtype" value="691"/>
                  <annotation datatype="java.lang.Boolean" key="builtin" value="true"/>
                  <annotation datatype="java.lang.Boolean" key="returnsunknown" value="false"/>
                  <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
                  <annotation datatype="java.lang.Long" key="frame-id" value="197568495829"/>
                  <ast col="31" id="197568495866" line="28" text="emp-num" type="FIELD_INT">
                    <annotation datatype="java.lang.Long" key="oldtype" value="2781"/>
                    <annotation datatype="java.lang.String" key="schemaname" value="p2j_test.person.emp-num"/>
                    <annotation datatype="java.lang.String" key="bufname" value="p2j_test.mybuff"/>
                    <annotation datatype="java.lang.String" key="dbname" value="p2j_test"/>
                    <annotation datatype="java.lang.Long" key="recordtype" value="12"/>
                    <annotation datatype="java.lang.String" key="name" value="emp-num"/>
                    <annotation datatype="java.lang.Long" key="type" value="410"/>
                    <annotation datatype="java.lang.String" key="format" value="&quot;9999999&quot;"/>
                    <annotation datatype="java.lang.String" key="columnlabel" value="&quot;Emp Num Column&quot
;"/>
                    <annotation datatype="java.lang.Long" key="param_index" value="0"/>
                    <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
                    <annotation datatype="java.lang.Boolean" key="is_meta" value="false"/>
                    <annotation datatype="java.lang.String" key="fieldname" value="empNum"/>
                    <annotation datatype="java.lang.String" key="bufrefkey" value="p2j_test.person,p2j_test.my
buff,197568495627"/>
                    <annotation datatype="java.lang.Long" key="bufreftype" value="18"/>
                    <annotation datatype="java.lang.String" key="uniquename" value="p2j_test.mybuff_p2j_test.p
erson"/>
                    <annotation datatype="java.lang.String" key="methodtxt" value="getEmpNum"/>
                    <annotation datatype="java.lang.String" key="check_accessors" value=""/>
                    <annotation datatype="java.lang.String" key="getter" value="getEmpNum"/>
                    <annotation datatype="java.lang.String" key="setter" value="setEmpNum"/>
                    <annotation datatype="java.lang.String" key="javaname" value="empNum"/>
                    <annotation datatype="java.lang.String" key="accessor" value="widgetEmpNum"/>
                    <annotation datatype="java.lang.String" key="widgettype" value="FillInWidget"/>
                    <annotation datatype="java.lang.Long" key="frame-id" value="197568495829"/>
                  </ast>
                </ast>
              </ast>
            </ast>
            <ast col="42" id="197568495792" line="28" text="&quot;&quot;" type="STRING">
              <annotation datatype="java.lang.Boolean" key="is-literal" value="true"/>
              <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
            </ast>
          </ast>
        </ast>
        <ast col="45" id="197568495794" line="28" text="then" type="KW_THEN">
          <ast col="0" id="197568495796" line="0" text="block" type="BLOCK">
            <annotation datatype="java.lang.Boolean" key="recordscoping" value="false"/>
            <ast col="0" id="197568495797" line="0" text="statement" type="STATEMENT">
              <ast col="9" id="197568495798" line="29" text="message" type="KW_MSG">
                <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
                <ast col="0" id="197568495800" line="0" text="" type="CONTENT_ARRAY">
                  <ast col="0" id="197568495801" line="0" text="expression" type="EXPRESSION">
```

```
                    <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
                    <ast col="17" id="197568495802" line="29" text="&quot;number not in use&quot;" type="STRING">
                      <annotation datatype="java.lang.Boolean" key="is-literal" value="true"/>
                      <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
                    </ast>
                  </ast>
                </ast>
              </ast>
            </ast>
          </ast>
        </ast>
        <ast col="7" id="197568495805" line="30" text="else" type="KW_ELSE">
          <ast col="0" id="197568495807" line="0" text="block" type="BLOCK">
            <annotation datatype="java.lang.Boolean" key="recordscoping" value="false"/>
            <ast col="0" id="197568495808" line="0" text="statement" type="STATEMENT">
              <ast col="9" id="197568495809" line="31" text="message" type="KW_MSG">
                <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
                <ast col="0" id="197568495811" line="0" text="" type="CONTENT_ARRAY">
                  <ast col="0" id="197568495812" line="0" text="expression" type="EXPRESSION">
                    <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
                    <ast col="17" id="197568495813" line="31" text="&quot;number in use&quot;" type="STRING">
                      <annotation datatype="java.lang.Boolean" key="is-literal" value="true"/>
                      <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
                    </ast>
                  </ast>
                </ast>
              </ast>
            </ast>
          </ast>
        </ast>
      </ast>
```

#3 - 11/26/2019 03:53 PM - Roger Borrello

I cannot duplicate the issue using 4207a-11342 using the testcase nor customer code.

#4 - 11/26/2019 06:02 PM - Roger Borrello

Roger Borrello wrote:

I cannot duplicate the issue using 4207a-11342 using the testcase nor customer code.

Here is the Java that is generated. It doesn't look like I would expect, although I'm a little green in this area.

```
package com.goldencode.testcases.abl;

import com.goldencode.p2j.util.*;
import com.goldencode.testcases.ui.abl.*;
import com.goldencode.p2j.ui.*;
import com.goldencode.p2j.persist.*;
import com.goldencode.testcases.dmo.p2j_test.*;
import com.goldencode.p2j.persist.lock.*;

import static com.goldencode.p2j.util.BlockManager.*;
import static com.goldencode.p2j.util.logical.*;
import static com.goldencode.p2j.util.ErrorManager.*;

/**
 * Business logic (converted to Java from the 4GL source code
 * in abl/extent_subscript_using_input.p).
 */
public class ExtentSubscriptUsingInput
{
   Person.Buf mybuff = RecordBuffer.define(Person.Buf.class, "p2j_test", "mybuff", "mybuff");

   ExtentSubscriptUsingInputMyframe myframeFrame = GenericFrame.createFrame(ExtentSubscriptUsingInputMyframe.c
lass, "myframe");

   /**
    * External procedure (converted to Java from the 4GL source code
    * in abl/extent_subscript_using_input.p).
    */
   public void execute()
   {
      externalProcedure(ExtentSubscriptUsingInput.this, TransactionType.FULL, new Block((Body) () ->
      {
         myframeFrame.openScope();
         RecordBuffer.openScope(mybuff);
         silent(() -> new FindQuery(mybuff, (String) null, null, "mybuff.siteId asc, mybuff.empNum asc").first
());

         if (_not(mybuff.available()))
         {
            mybuff.create();
         }

         doBlock("blockLabel0", new Block((Body) () ->
         {
            loopLabel0:
            while (true)
            {
            }
         }));
      }));
   }
}
```

#5 - 11/26/2019 06:06 PM - Greg Shah

It is missing all output for this:

```
  update
    mybuff.schedule[1]
    mybuff.schedule[2]
    mybuff.schedule[3]
    mybuff.schedule[4]
    mybuff.schedule[5]
    mybuff.emp-num
  edit-loop:
    editing:
      readkey.
      if input schedule[input emp-num] = "" then
        message "number not in use".
      else
        message "number in use".
    end.
  end.
```

This is a regression from my changes in rev 11342.

#6 - 11/26/2019 07:31 PM - Greg Shah

The regression is fixed in rev 11343. Now this testcase fails again. Success!

#7 - 11/27/2019 12:48 PM - Roger Borrello

This same testcase has the issue, so buffers are not the only thing leading to it. It could be 2 separate issues, or both the same root-cause.

```
define variable mybuff as character extent 1 no-undo.
define variable mynum as integer no-undo.

form
    mybuff[1]
    mynum
    with frame myframe.

do with frame myframe:
  do while true:
  update
    mybuff[1]
    mynum
  edit-loop:
    editing:
      readkey.
      if input mybuff[input mynum] = "" then
        message "number not in use".
      else
        message "number in use".
    end.
  end.
end.
```

#8 - 12/02/2019 12:02 PM - Roger Borrello

Roger Borrello wrote:

> This same testcase has the issue, so buffers are not the only thing leading to it. It could be 2 separate issues, or both the same root-cause.
>
> [...]

In order to determine what is wrong, I created 2 separate testcases.

1. extent_subscript_using_input/extent_subscript_using_input.p - Just uses variables
2. extent_subscript_using_input/extent_subscript_using_input-buffer.p - Uses a buffer

The end results is 2 ast files that have the same issue.

FUNC_POLY (input) will have either FIELD_CHAR (schedule) or VAR_CHAR (mybuff) as children. Those children will both have LBRACKET -> FUNC_POLY (input) as children, with subsequent FIELD_INT (emp-num) or VAR_INT (mynum) as children.

Both cases, there is another LBRACKET -> FUNC_POLY (input) with subsequent FIELD_INT (emp-num) or VAR_INT (mynum) as an additional parameter to the original FUNC_POLY (input).

Still investigating.

#9 - 12/02/2019 07:07 PM - Roger Borrello

~~This has been fixed in 4207a-11349~~

#10 - 12/09/2019 02:48 PM - Roger Borrello

- Estimated time set to 16.00

- Status changed from New to WIP

- % Done changed from 0 to 100

#11 - 12/17/2019 05:31 PM - Roger Borrello

This still is not fixed, as the java code does not compile. Below is the compilation error of the testcase:

```
    [javac] testcases/uast/uast/buildarea/src/com/goldencode/testcases/abl/ExtentSubscriptUsingInput.java:45: 
error: method getMynum in interface ExtentSubscriptUsingInputMyframe cannot be applied to given types;
```

```
    [javac]                 if (_isEqual(myframeFrame.getMybuffArray((myframeFrame.getMynum(myframeFrame.widgetMynum())).subscript()), ""))
    [javac]                                                                                    ^
    [javac]   required: no arguments
    [javac]   found: FillInWidget
    [javac]   reason: actual and formal argument lists differ in length
```

#12 - 12/20/2019 04:02 PM - Roger Borrello

I looked at a much simpler case of using input mynum then input mybuff[mynum] instead of the more complicated, nested case input mybuff[input mynum]].

The generated Java compiled:

```
            updateEditing(myframeFrame, elementList0, "editingLabel0", new Block((Body) () ->
            {
               KeyReader.readKey();
               myframeFrame.getMynum();

               if (_isEqual(myframeFrame.getMybuffArray((mynum).subscript()), ""))
```

Trying to simplify it one more layer, and just using input mybuff[1] generated Java using the more direct myframeFrame.getMybuffArray0() instead of the (mynum).subscript() method, so that didn't help.

The generated code from input mybuff[input mynum] becomes:

```
            updateEditing(myframeFrame, elementList0, "editingLabel0", new Block((Body) () ->
            {
               KeyReader.readKey();

               if (_isEqual(myframeFrame.getMybuffArray((myframeFrame.getMynum(myframeFrame.widgetMynum())).subscript()), ""))
```

I'm wondering why the inner [input mynum] did not result in myframeFrame.getMynum() like it did when it was on its own line. Instead it became myframeFrame.widgetMynum() which was passed as a parameter to the myframeFrame.getMynum() method. That method doesn't take a parameter.

The correct code should be myframeFrame.getMybuffArray(((myframeFrame.getMynum()).subscript())) shouldn't it? It compiles, in any case.

#13 - 12/20/2019 04:54 PM - Roger Borrello

Roger Borrello wrote:

> I looked at a much simpler case of using input mynum then input mybuff[mynum] instead of the more complicated, nested case input mybuff[input mynum]].
>
> The generated Java compiled:
> [...]
>
> Trying to simplify it one more layer, and just using input mybuff[1] generated Java using the more direct myframeFrame.getMybuffArray0() instead of the (mynum).subscript() method, so that didn't help.
>
> The generated code from input mybuff[input mynum] becomes:
>
> [...]
>
> I'm wondering why the inner input mynum] did not result in myframeFrame.getMynum() like it did when it was on its own line. Instead it became myframeFrame.widgetMynum() which was passed as a parameter to the myframeFrame.getMynum() method. That method doesn't take a parameter.
>
> The correct code should be myframeFrame.getMybuffArray(((myframeFrame.getMynum()).subscript())) shouldn't it? It compiles, in any case.

I went to a basic input mybuff[x] where x was just an integer. It generated:

```
                if (_isEqual(myframeFrame.getMybuffArray((x).subscript()), ""))
```

#14 - 12/23/2019 11:53 AM - Roger Borrello

What is dynamicindex annotation used for? I'm trying to determine why the LBRACKET subtree is duplicated in input_builtin_prep.rules

#15 - 12/24/2019 10:22 AM - Roger Borrello

Roger Borrello wrote:

> What is dynamicindex annotation used for? I'm trying to determine why the LBRACKET subtree is duplicated in input_builtin_prep.rules

Still not understanding why the AST has duplicate LBRACKET node created as a child of FUNC_POLY(input). And if that is correct, why isn't there another created, since we have a 2nd input in an LBRACKET?

#16 - 12/24/2019 10:55 AM - Constantin Asofiei

Roger Borrello wrote:

> I went to a basic input mybuff[x] where x was just an integer. It generated:

I don't think the subscript needs to be generated here. This is required in cases like xVar[idx], to emit as subscript(xVar, idx). In your case, the widget is already emitted correctly, so the subscript doesn't need to be emitted.

#17 - 12/24/2019 10:59 AM - Constantin Asofiei

Constantin Asofiei wrote:

> Roger Borrello wrote:
>
>> I went to a basic input mybuff[x] where x was just an integer. It generated:
>
> I don't think the subscript needs to be generated here. This is required in cases like xVar[idx], to emit as subscript(xVar, idx). In your case, the widget is already emitted correctly, so the subscript doesn't need to be emitted.

I take this back, the subscript is required to make the index 0-based instead of 1-based, as the frame widgets for an extent var will be numbered 0-based, not 1-based, during conversion.

#18 - 12/24/2019 11:11 AM - Roger Borrello

Thanks for looking. I had a much more complicated testcase originally, which has been greatly reduced:

```
define variable mybuff as character extent 1 no-undo.
define variable mynum as integer no-undo.
define variable x as integer no-undo.

form
    mybuff[1]
    mynum
    with frame myframe.

do with frame myframe:
  update
    mybuff[1]
    mynum
  editing:
    readkey.
    if input mybuff[input mynum] = "" then
      message "Blank".
    else
      message "Not blank".
  end.
end.
```

Greg had me add pre and post processing to annotations/input_builtin_prep.rules to push the id of the kw_input node so it could be used to access the original node's methodtext and override annotations, and add them into the current node.

That allowed us to complete parsing, but the generated Java doesn't compile, as noted above. I am having trouble understanding of the problems is in annotations, or convert.

#19 - 12/24/2019 11:16 AM - Constantin Asofiei

There's something else going on. This case parses/converts properly:

```
def var c as int extent 5.
def var i as int.

form c i with frame f1.

def var ch as char.
ch = input c[input i].
ch = input c[i].

if input c[input i] = "" then message "1".
if input c[i] = "" then message "1".
```

but if you change the type of c var to character, it no longer converts. The conversion was with 3809e.

#20 - 12/24/2019 11:17 AM - Constantin Asofiei

Roger Borrello wrote:

> Thanks for looking. I had a much more complicated testcase originally, which has been greatly reduced:

Same with this test: if you change mybuff to integer, it will parse and convert properly with 3809e. So, we might have a problem with some special processing for character widgets.

#21 - 12/24/2019 11:44 AM - Roger Borrello

Constantin Asofiei wrote:

> Roger Borrello wrote:
>
> > Thanks for looking. I had a much more complicated testcase originally, which has been greatly reduced:
>
> Same with this test: if you change mybuff to integer, it will parse and convert properly with 3809e. So, we might have a problem with some special processing for character widgets.

If you put both cases in the same testcase (one with character, one with integer) they both have an issue:

```
define variable mybuff as character extent 1 no-undo.
define variable mynums as integer extent 1 no-undo.
define variable mynum as integer no-undo.
define variable x as integer no-undo.

form mybuff[1] mynum with frame f1.
form mynums[1] mynum with frame f2.

do with frame f1:
  update mybuff[1] mynum
  editing:
    readkey.
    if input mybuff[input mynum] = "" then
      message "Blank".
    else
      message "Not blank".
  end.
end.
do with frame f2:
  update mynums[1] mynum
  editing:
    readkey.
    if input mynums[input mynum] = 0 then
      message "Zero".
    else
      message "Not zero".
  end.
end.
```

Converted:

```
         doBlock("blockLabel0", new Block((Body) () ->
         {
            FrameElement[] elementList0 = new FrameElement[]
            {
               new Element(subscript(mybuff, 1), f1Frame.widgetMybuffArray0()),
               new Element(mynum, f1Frame.widgetMynum())
            };

            updateEditing(f1Frame, elementList0, "editingLabel0", new Block((Body) () -> 
            {
               KeyReader.readKey();

               if (_isEqual(f1Frame.getMybuffArray((f2Frame.getMynum(f2Frame.widgetMynum())).subscript()), "")
)
               {
                  message("Blank");
               }
               else
               {
                  message("Not blank");
               }
            }));
         }));

         doBlock("blockLabel1", new Block((Body) () -> 
         {
            FrameElement[] elementList1 = new FrameElement[]
            {
               new Element(subscript(mynums, 1), f2Frame.widgetMynumsArray0()),
               new Element(mynum, f2Frame.widgetMynum())
            };

            updateEditing(f2Frame, elementList1, "editingLabel1", new Block((Body) () -> 
            {
               KeyReader.readKey();

               if (_isEqual(f2Frame.getMynumsArray((f2Frame.getMynum(f2Frame.widgetMynum())).subscript()), 0))
               {
                  message("Zero");
               }
               else
               {
                  message("Not zero");
               }
            }));
         }));
```

#22 - 12/24/2019 12:19 PM - Roger Borrello

I see that getScreenValue() is the method used in your testcase. The AST and Java look so much cleaner than in my testcase. How could VAR_CHAR be so much different?

I am on holiday until 31 December. If there's any notes you could leave me, with hints about what to look for, I would be very happy to squash this bug.

#23 - 12/24/2019 02:22 PM - Constantin Asofiei

Roger, I think I found the root cause. Try replacing the walk-rules with ascent-rules in input_builtin_prep.rules. Why will this work: we have a nested INPUT function case. But, in case of character widgets, we do some processing to avoid the getScreenValue(). This includes duplicating an AST, and marking the original one as hidden. If we do this in the walk-rules, when we duplicate the AST, we duplicate the unprocessed nested INPUT AST. So, doing this on the ascent, we ensure we are duplicating the fully-processed AST.

What made it tick for me was that the nested INPUT AST (inside the hidden AST) had annotations (like methodtxt), while the copy did not.

#24 - 12/31/2019 10:19 AM - Roger Borrello

Constantin Asofiei wrote:

> Roger, I think I found the root cause. Try replacing the walk-rules with ascent-rules in input_builtin_prep.rules. Why will this work: we have a nested INPUT function case. But, in case of character widgets, we do some processing to avoid the getScreenValue(). This includes duplicating an AST, and marking the original one as hidden. If we do this in the walk-rules, when we duplicate the AST, we duplicate the unprocessed nested INPUT AST. So, doing this on the ascent, we ensure we are duplicating the fully-processed AST.
>
> What made it tick for me was that the nested INPUT AST (inside the hidden AST) had annotations (like methodtxt), while the copy did not.

Well Happy New Year! That did the trick for my testcase!

```
define variable mybuff as character extent 1 no-undo.
define variable mynums as integer extent 1 no-undo.
define variable mynum as integer no-undo.
define variable x as integer no-undo.

form mybuff[1] mynum with frame f1.
form mynums[1] mynum with frame f2.

do with frame f1:
  update mybuff[1] mynum
  editing:
    readkey.
    if input mybuff[input mynum] = "" then
      message "Blank".
    else
      message "Not blank".
  end.
end.
do with frame f2:
  update mynums[1] mynum
  editing:
    readkey.
    if input mynums[input mynum] = 0 then
      message "Zero".
    else
      message "Not zero".
  end.
end.
```

Converts to:

```
package com.goldencode.testcases.input;

import com.goldencode.p2j.util.*;
import com.goldencode.testcases.ui.input.*;
import com.goldencode.p2j.ui.*;

import static com.goldencode.p2j.util.BlockManager.*;
import static com.goldencode.p2j.util.CompareOps.*;
import static com.goldencode.p2j.util.ArrayAssigner.*;
import static com.goldencode.p2j.ui.LogicalTerminal.*;

/**
 * Business logic (converted to Java from the 4GL source code
 * in input/extent_subscript_using_input.p).
 */
public class ExtentSubscriptUsingInput
{
   ExtentSubscriptUsingInputF1 f1Frame = GenericFrame.createFrame(ExtentSubscriptUsingInputF1.class, "f1");

   ExtentSubscriptUsingInputF2 f2Frame = GenericFrame.createFrame(ExtentSubscriptUsingInputF2.class, "f2");

   /**
    * External procedure (converted to Java from the 4GL source code
    * in input/extent_subscript_using_input.p).
    */
   public void execute()
   {
      character[] mybuff = TypeFactory.characterExtent(1);
      integer[] mynums = TypeFactory.integerExtent(1);
      integer mynum = TypeFactory.integer();

      externalProcedure(ExtentSubscriptUsingInput.this, new Block((Body) () -> 
      {
         f1Frame.openScope();
         f2Frame.openScope();

         doBlock("blockLabel0", new Block((Body) () -> 
         {
            FrameElement[] elementList0 = new FrameElement[]
            {
               new Element(subscript(mybuff, 1), f1Frame.widgetMybuffArray0()),
               new Element(mynum, f1Frame.widgetMynum())
            };

            updateEditing(f1Frame, elementList0, "editingLabel0", new Block((Body) () -> 
            {
               KeyReader.readKey();

               if (_isEqual(f1Frame.getMybuffArray((f2Frame.getMynum()).subscript()), ""))
               {
                  message("Blank");
               }
               else
               {
                  message("Not blank");
               }
            }));
         }));

         doBlock("blockLabel1", new Block((Body) () -> 
         {
            FrameElement[] elementList1 = new FrameElement[]
            {
               new Element(subscript(mynums, 1), f2Frame.widgetMynumsArray0()),
               new Element(mynum, f2Frame.widgetMynum())
            };

            updateEditing(f2Frame, elementList1, "editingLabel1", new Block((Body) () -> 
            {
               KeyReader.readKey();

               if (_isEqual(f2Frame.getMynumsArray((f2Frame.getMynum()).subscript()), 0))
               {
                  message("Zero");
               }
               else
               {
                  message("Not zero");
               }
            }));
         }));
      }));
   }
}
```

I will look at customer code, too.

#25 - 12/31/2019 10:39 AM - Roger Borrello

Committed rules/annotations/input_builtin_prep.rules to **4207a-11367**.

#26 - 12/31/2019 10:42 AM - Greg Shah

- Status changed from WIP to Test

#27 - 03/03/2020 02:43 PM - Roger Borrello

Task branch 4207a was merged to trunk as revision 11344.

#28 - 03/04/2020 10:29 AM - Greg Shah

- Status changed from Test to Closed