

Base Language - Bug #4424

Additional def var needed to receive table handle when keyword FOR included in def parameter

11/22/2019 02:08 PM - Roger Borrello

Status: Closed	Start date:
Priority: Normal	Due date:
Assignee: Greg Shah	% Done: 100%
Category:	Estimated time: 0.00 hour
Target version:	
billable: No	case_num:
vendor_id: GCD	version:
Description	
Related issues:	
Related to Database - Feature #3816: table and table handle parameter options Closed	

History

#1 - 11/22/2019 02:58 PM - Roger Borrello

General

We found usual syntax in code where the keyword for was included as a qualifier to the def input parameter... statement, which cause an Expression execution error.

20191231 Update: Fixed in 4207a Revisions 11346-11347

Update: Being worked by GES

P2J Branch

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

Testcase

Checked into testcase_repo:

uast/table-handle_parameters/tablehandle_parameter_with_for.p

```
procedure input_parameter_with_for:
  // Use these 2 lines to demonstrate the behavior of the "for" being handled.
  // Note that order matters. If you reverse the "define variable" and "define input parameter"
  // lines, it will result in the below error:
  // ** Unknown Field or Variable name - gh. (201)
  // Only unsubscripted INPUT HANDLE program variables allowed in TABLE-HANDLE FOR phrase. (9080)
  define variable gh as handle no-undo.
  define input parameter table-handle for gh.

  // Use the below line to demonstrate typical behavior (without the "for")

  //define input parameter table-handle gh.

  message gh:name.
end procedure.

define variable tt as handle no-undo.
create temp-table tt.

define variable tth as handle no-undo.

tt:add-new-field("num","integer").

tt:temp-table-prepare("ordx").
```

```
tth = tt:handle.
```

```
run input_parameter_with_for (table-handle tth).
```

Artifacts

Execution Log

```
[java] -----  
[java] Core Code Conversion  
[java] -----  
[java]  
[java] Optional rule set [customer_specific_conversion] not found.  
[java] ./abl/tablehandle_parameter_with_for.p  
[java] EXPRESSION EXECUTION ERROR:  
[java] -----  
[java] ref = this.getImmediateChild(prog.kw_tab_hand, null).getImmediateChild(prog.expression, null).getI  
mmEDIATEChild(prog.var_handle, null)  
[java] ^ {  
java.lang.NullPointerException }  
[java] -----  
[java] Elapsed job time: 00:00:00.759  
[java] ERROR:  
[java] com.goldencode.p2j.pattern.TreeWalkException: ERROR! Active Rule:  
[java] -----  
[java] RULE REPORT  
[java] -----  
[java] Rule Type : WALK  
[java] Source AST: [ define ] BLOCK/PROCEDURE/BLOCK/STATEMENT/DEFINE_PARAMETER/ @5:4 {17179869212}  
[java] Copy AST : [ define ] BLOCK/PROCEDURE/BLOCK/STATEMENT/DEFINE_PARAMETER/ @5:4 {17179869212}  
[java] Condition : ref = this.getImmediateChild(prog.kw_tab_hand, null).getImmediateChild(prog.expressio  
n, null).getImmediateChild(prog.var_handle, null)  
[java] Loop : false  
[java] --- END RULE REPORT ---  
[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 @1:95 [DEFINE_PARAM  
ETER id=17179869212]  
[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 @1:95  
[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.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: java.lang.NullPointerException  
[java] at com.goldencode.expr.CE9850.execute (Unknown Source)  
[java] at com.goldencode.expr.Expression.execute (Expression.java:391)  
[java] ... 21 more
```

AST Snippet

```

<ast col="0" id="17179869211" line="0" text="statement" type="STATEMENT">
  <ast col="4" id="17179869212" line="5" text="define" type="DEFINE_PARAMETER">
    <annotation datatype="java.lang.String" key="name" value="gh"/>
    <annotation datatype="java.lang.Long" key="type" value="380"/>
    <annotation datatype="java.lang.Long" key="parmtype" value="691"/>
    <annotation datatype="java.lang.Boolean" key="vardef" value="true"/>
    <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
    <annotation datatype="java.lang.String" key="javaname" value="gh_1"/>
    <annotation datatype="java.lang.String" key="classname" value="handle"/>
    <annotation datatype="java.lang.Long" key="procrefid" value="17179869186"/>
    <ast col="11" id="17179869214" line="5" text="input" type="KW_INPUT"/>
    <ast col="27" id="17179869218" line="5" text="table-handle" type="KW_TAB_HAND">
      <ast col="40" id="17179869220" line="5" text="for" type="KW_FOR"/>
      <ast col="44" id="17179869222" line="5" text="gh" type="SYMBOL"/>
    </ast>
  </ast>
</ast>

```

#2 - 11/22/2019 03:08 PM - Greg Shah

The solution is to add code in annotations that will detect this case. When found, it should:

- delete the FOR keyword and fixup the tree as if it was never there
- delete the extra handle variable

If I understand correctly, the code will naturally convert after that.

#4 - 11/26/2019 09:37 AM - Roger Borrello

Note that reversing the lines in the testcase alters the behavior in Progress.

```

define variable gh as handle no-undo.
define input parameter table-handle for gh.

```

works fine. But reversing the lines:

```
define input parameter table-handle for gh.  
define variable gh as handle no-undo.
```

results in:

```
** Unknown Field or Variable name - gh. (201)  
Only unsubscripted INPUT HANDLE program variables allowed in TABLE-HANDLE FOR phrase. (9080)
```

#5 - 11/27/2019 09:57 AM - Greg Shah

- Status changed from New to Test
- % Done changed from 0 to 100

The fix for this is in 4207a revision 11344.

#6 - 11/27/2019 09:58 AM - Greg Shah

- Assignee set to Greg Shah

#7 - 11/27/2019 09:59 AM - Greg Shah

- Related to Feature #3816: table and table handle parameter options added

#8 - 11/27/2019 10:11 AM - Greg Shah

This fix enables the TABLE-HANDLE FOR <existing_handle_var> use case. We already have support in conversion (to see it, search on `is_table_handle_for`) but it did not work. This was due to the fact that in `progress.g` we only matched a SYMBOL and not a possible expr. This extra matching is only in the case where the FOR is present (undocumented 4GL feature).

#9 - 11/27/2019 10:18 AM - Roger Borrello

Converted code fails to compile

```
compile:  
[javac] Compiling 36 source files to /home/rfb/projects/VirtualBox-VMs/shared/projects/testcases/uast/uast  
/buildarea/build/classes  
[javac] /home/rfb/projects/VirtualBox-VMs/shared/projects/testcases/uast/uast/buildarea/src/com/goldencode  
/testcases/abl/TablehandleParameterWithFor.java:38: error: annotation @LegacyParameter is missing a default va  
lue for the element 'type'  
[javac]         @LegacyParameter(mode = "INPUT")  
[javac]         ^  
[javac] /home/rfb/projects/VirtualBox-VMs/shared/projects/testcases/uast/uast/buildarea/src/com/goldencode  
/testcases/abl/TablehandleParameterWithFor.java:45: error: cannot find symbol  
[javac]         handle gh = new handle(_gh.getTableHandle());  
[javac]                             ^  
[javac] symbol:   variable _gh  
[javac] location: class TablehandleParameterWithFor
```

```
[javac] /home/rfb/projects/VirtualBox-VMs/shared/projects/testcases/uast/uast/buildarea/src/com/goldencode
/testcases/abl/TablehandleParameterWithFor.java:49: error: cannot find symbol
[javac]         TemporaryBuffer.createDynamicTable(_gh, gh, true, false);
[javac]                                     ^
[javac]     symbol:   variable _gh
[javac]     location: class TablehandleParameterWithFor
[javac] 3 errors
```

#10 - 11/27/2019 10:45 AM - Constantin Asofiei

Roger, take these files from 3809e:

```
rules/annotations/collect_names.rules
rules/convert/method_definitions.rules
```

They are built from trunk, so is safe to apply.

#11 - 11/27/2019 11:15 AM - Roger Borrello

Constantin Asofiei wrote:

Roger, take these files from 3809e:

[...]

They are built from trunk, so is safe to apply.

Same errors occurred after applying those files. What were you expecting to be fixed?

I notice that `_null` is passed in as the parameter, hinting that the emitter didn't know what the parameter was.

```
@LegacySignature(type = Type.PROCEDURE, name = "input_parameter_with_for", parameters =
{
    @LegacyParameter(mode = "INPUT")
})
/**
 * Use these 2 lines to demonstrate the behavior of the "for" being handled.
 */
public void inputParameterWithFor(final TableParameter _null)
{
    handle gh = new handle(_gh.getTableHandle());

    internalProcedure(new Block((Body) () ->
    {
        TemporaryBuffer.createDynamicTable(_gh, gh, true, false);
        message(gh.unwrap().name());
    }));

    // Use the below line to demonstrate typical behavior (without the "for")
```

```
// define input parameter table-handle gh.
```

```
}
```

#12 - 11/27/2019 11:16 AM - Constantin Asofiei

They fix another issue, I thought it was related to this. So looks like there's another case to fix. Please post the AST for define input parameter table-handle for gh.

#13 - 11/27/2019 11:20 AM - Roger Borrello

Constantin Asofiei wrote:

They fix another issue, I thought it was related to this. So looks like there's another case to fix. Please post the AST for define input parameter table-handle for gh.

```
<ast col="0" id="210453397537" line="0" text="statement" type="STATEMENT">
  <ast col="4" id="210453397538" line="8" text="define" type="DEFINE_PARAMETER">
    <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
    <annotation datatype="java.lang.Long" key="procrefid" value="210453397506"/>
    <ast col="11" id="210453397540" line="8" text="input" type="KW_INPUT"/>
    <ast col="27" id="210453397544" line="8" text="table-handle" type="KW_TAB_HAND">
      <ast col="40" id="210453397546" line="8" text="for" type="KW_FOR"/>
      <ast col="0" id="210453397548" line="0" text="expression" type="EXPRESSION">
        <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
        <ast col="44" id="210453397549" line="8" text="gh" type="VAR_HANDLE">
          <annotation datatype="java.lang.Long" key="oldtype" value="2782"/>
          <annotation datatype="java.lang.Long" key="refid" value="210453397524"/>
          <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
        </ast>
      </ast>
    </ast>
  </ast>
</ast>
</ast>
```

#14 - 11/27/2019 11:23 AM - Roger Borrello

Roger Borrello wrote:

Constantin Asofiei wrote:

They fix another issue, I thought it was related to this. So looks like there's another case to fix. Please post the AST for define input parameter table-handle for gh.

[...]

Note that in this testcase, the gh is used in both a define variable and define input parameter table-handle for gh:

```
define variable gh as handle no-undo.  
define input parameter table-handle for gh.
```

#15 - 11/27/2019 11:24 AM - Constantin Asofiei

Roger Borrello wrote:

Note that in this testcase, the gh is used in both a define variable and define input parameter table-handle for gh:

Is it the same 'pointer'? If you change one, does the other change too, in the same program?

#16 - 11/27/2019 11:26 AM - Roger Borrello

Constantin Asofiei wrote:

Roger Borrello wrote:

Note that in this testcase, the gh is used in both a define variable and define input parameter table-handle for gh:

Is it the same 'pointer'? If you change one, does the other change too, in the same program?

Those 2 lines can be replaced with define input parameter table-handle gh. and you will get proper behavior. Does that help?

#17 - 11/27/2019 11:27 AM - Constantin Asofiei

Roger Borrello wrote:

Those 2 lines can be replaced with define input parameter table-handle gh. and you will get proper behavior. Does that help?

Yes, it should. Please post the AST and the LegacySignature annotation.

#18 - 11/27/2019 11:30 AM - Roger Borrello

Constantin Asofiei wrote:

Roger Borrello wrote:

Those 2 lines can be replaced with define input parameter table-handle gh. and you will get proper behavior. Does that help?

Yes, it should. Please post the AST and the LegacySignature annotation.

```
<ast col="0" id="210453397529" line="0" text="statement" type="STATEMENT">
  <ast col="4" id="210453397530" line="11" text="define" type="DEFINE_PARAMETER">
    <annotation datatype="java.lang.String" key="name" value="gh"/>
    <annotation datatype="java.lang.Long" key="type" value="380"/>
    <annotation datatype="java.lang.Long" key="parmttype" value="691"/>
    <annotation datatype="java.lang.Boolean" key="vardef" value="true"/>
    <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
    <annotation datatype="java.lang.String" key="javaname" value="gh"/>
    <annotation datatype="java.lang.String" key="classname" value="handle"/>
    <annotation datatype="java.lang.Long" key="procrefid" value="210453397506"/>
    <ast col="11" id="210453397532" line="11" text="input" type="KW_INPUT"/>
    <ast col="27" id="210453397536" line="11" text="table-handle" type="KW_TAB_HAND">
      <ast col="40" id="210453397538" line="11" text="gh" type="SYMBOL"/>
    </ast>
  </ast>
</ast>
```

```
@LegacySignature(type = Type.PROCEDURE, name = "input_parameter_with_for", parameters =
{
  @LegacyParameter(name = "gh", type = "HANDLE", mode = "INPUT")
})
/**
 * Use these 2 lines to demonstrate the behavior of the "for" being handled.
 */
public void inputParameterWithFor(final TableParameter _gh)
{
  handle gh = new handle();

  internalProcedure(new Block((Body) () ->
  {
    TemporaryBuffer.createDynamicTable(_gh, gh, true, false);
    message(gh.unwrap().name());
  }));
}
```


#19 - 11/27/2019 11:39 AM - Constantin Asofiei

Roger, there is no vardef annotation in the case in [#4424-13](#) (the AST is from the full conversion, right?) Add `-Drules.tracing=true` to your conversion script and check a `vardef-by-rule`, to find who sets this annotation.

#20 - 11/27/2019 11:49 AM - Roger Borrello

Constantin Asofiei wrote:

Roger, there is no vardef annotation in the case in [#4424-13](#) (the AST is from the full conversion, right?) Add `-Drules.tracing=true` to your conversion script and check a `vardef-by-rule`, to find who sets this annotation.

This didn't work: `java -classpath p2j/build/lib/p2j.jar com.goldencode.p2j.convert.ConversionDriver -Drules.tracing=true f2+m0+cb ./table-handle_parameters/tablehandle_parameter_with_for.p`

Is there a `-D` debug level I should use, or does `rule.tracing=true` go someplace else?

#21 - 11/27/2019 11:50 AM - Constantin Asofiei

Roger Borrello wrote:

... or does `rule.tracing=true` go someplace else?

Before the main class, like this:

```
java -Drules.tracing=true -classpath p2j/build/lib/p2j.jar com.goldencode.p2j.convert.ConversionDriver f2+m0+cb
./table-handle_parameters/tablehandle_parameter_with_for.p
```

#22 - 11/27/2019 12:05 PM - Roger Borrello

Constantin Asofiei wrote:

Roger Borrello wrote:

... or does `rule.tracing=true` go someplace else?

Before the main class, like this:

```
java -Drules.tracing=true -classpath p2j/build/lib/p2j.jar com.goldencode.p2j.convert.ConversionDriver f2+m0+cb
./table-handle_parameters/tablehandle_parameter_with_for.p
```

Thank you for that hint, I'll be using it!

Here's what's in the parameter section for the *good* version:

```
<ast col="4" id="468151435290" line="11" text="define" type="DEFINE_PARAMETER">
  <annotation datatype="java.lang.String" key="name" value="gh"/>
  <annotation datatype="java.lang.Long" key="type" value="380"/>
  <annotation datatype="java.lang.Long" key="parmtype" value="691"/>
  <annotation datatype="java.lang.Boolean" key="vardef" value="true"/>
  <annotation datatype="java.lang.String" key="vardef-byrule" value="fixups/post_parse_fixups.xml:41
2"/>
  <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
  <annotation datatype="java.lang.String" key="support_level-byrule" value="gaps/gap_analysis_markin
g.xml:430"/>
  <annotation datatype="java.lang.String" key="javaname" value="gh"/>
  <annotation datatype="java.lang.String" key="javaname-byrule" value="annotations/variable_definiti
ons.rules:191"/>
  <annotation datatype="java.lang.String" key="classname" value="handle"/>
  <annotation datatype="java.lang.String" key="classname-byrule" value="annotations/variable_definit
ions.rules:332"/>
  <annotation datatype="java.lang.Long" key="procrefid" value="468151435266"/>
  <annotation datatype="java.lang.String" key="procrefid-byrule" value="annotations/procedures.rules
:181"/>
  <ast col="11" id="468151435292" line="11" text="input" type="KW_INPUT"/>
  <ast col="27" id="468151435296" line="11" text="table-handle" type="KW_TAB_HAND">
    <ast col="40" id="468151435298" line="11" text="gh" type="SYMBOL"/>
  </ast>
</ast>
```

And here what there for the *bad* version:

```
<ast col="4" id="468151435298" line="8" text="define" type="DEFINE_PARAMETER">
  <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
  <annotation datatype="java.lang.String" key="support_level-byrule" value="gaps/gap_analysis_markin
g.xml:430"/>
  <annotation datatype="java.lang.Long" key="procrefid" value="468151435266"/>
  <annotation datatype="java.lang.String" key="procrefid-byrule" value="annotations/procedures.rules
:181"/>
  <ast col="11" id="468151435300" line="8" text="input" type="KW_INPUT"/>
  <ast col="27" id="468151435304" line="8" text="table-handle" type="KW_TAB_HAND">
    <ast col="40" id="468151435306" line="8" text="for" type="KW_FOR"/>
    <ast col="0" id="468151435308" line="0" text="expression" type="EXPRESSION">
      <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
      <annotation datatype="java.lang.String" key="support_level-byrule" value="gaps/gap_analysis_ma
rking.xml:430"/>
    <ast col="44" id="468151435309" line="8" text="gh" type="VAR_HANDLE">
      <annotation datatype="java.lang.Long" key="oldtype" value="2782"/>
      <annotation datatype="java.lang.Long" key="refid" value="468151435284"/>
      <annotation datatype="java.lang.String" key="refid-byrule" value="fixups/post_parse_fixups.x
ml:480"/>
    <annotation datatype="java.lang.Long" key="support_level" value="16400"/>
    <annotation datatype="java.lang.String" key="support_level-byrule" value="gaps/gap_analysis_
marking.xml:430"/>
  </ast>
</ast>
</ast>
</ast>
```

#23 - 11/27/2019 12:07 PM - Constantin Asofiei

Roger Borrello wrote:

Here's what's in the parameter section for the *good* version:

The vardef annotation is set by fixups/post_parse_fixups.xml:412; this checks for a tempidx annotation (after parsing) - I think you are missing this annotation, so this is a parser problem. Check the ASTs with the good and bad versions, only after the parse (without post_parse_fixups) - I don't know how to block FWD to not execute post_parse_fixups@, I usually attach the debugger (or run from Eclipse), breakpoint right after parsing has finished, and kill the process.

#24 - 11/27/2019 12:17 PM - Roger Borrello

Constantin Asofiei wrote:

Roger Borrello wrote:

Here's what's in the parameter section for the *good* version:

The vardef annotation is set by fixups/post_parse_fixups.xml:412; this checks for a tempidx annotation (after parsing) - I think you are missing this annotation, so this is a parser problem. Check the ASTs with the good and bad versions, only after the parse (without post_parse_fixups) - I don't know how to block FWD to not execute post_parse_fixups@, I usually attach the debugger (or run from Eclipse), breakpoint right after parsing has finished, and kill the process.

I am trying to dig into [#4427](#) at the same time. Are you able grab 4207a to help? It would be greatly appreciated.

#25 - 11/27/2019 01:24 PM - Constantin Asofiei

Roger Borrello wrote:

I am trying to dig into [#4427](#) at the same time. Are you able grab 4207a to help? It would be greatly appreciated.

Sorry, I was debugging another issue and I need to sign off for today. I'll look tomorrow, if you don't get to it today.

#26 - 11/28/2019 01:28 PM - Constantin Asofiei

I've been trying to understand how TABLE-HANDLE and TABLE-HANDLE FOR behave - looks like they are not equivalent. The FOR option links the local var to the parameter (as a 'pointer') - if you change the parameter, the variable is changed, too. There may be more differences.

#27 - 11/28/2019 02:20 PM - Constantin Asofiei

Stanislav, I see that you've added testcases related to PARAMETER TABLE-HANDLE [FOR] (like th-for-regular.p). Was this supposed to be working in FWD at some point?

#28 - 11/28/2019 02:40 PM - Stanislav Lomany

Sorry, I don't remember anything about that.

#29 - 11/28/2019 05:05 PM - Constantin Asofiei

Greg, I've fixed this in the conversion rules (it was not a parser problem in the end). But before checking what it works in 4GL, I added also DATASET-HANDLE FOR and DATASET/TABLE-HANDLE FOR in functions (because it made sense, to be symmetric...) - this is not supported by 4GL, should I remove these?

#30 - 11/28/2019 05:22 PM - Constantin Asofiei

Anyway, the changes are in 4207a rev 11346. Greg, please review.

#31 - 11/28/2019 10:31 PM - Roger Borrello

Constantin Asofiei wrote:

Anyway, the changes are in 4207a rev 11346. Greg, please review.

Sorry for the lack of response. Today was Thanksgiving in the US. The customer code used this unorthodox method to specify a buffer handle as an input parameter.

You have to be careful to work your 3809e changes into these changes. I see some conflicts with your H022 updates.

#32 - 11/29/2019 06:09 AM - Constantin Asofiei

Roger Borrello wrote:

Sorry for the lack of response. Today was Thanksgiving in the US.

Happy Thanksgiving! :)

You have to be careful to work your 3809e changes into these changes. I see some conflicts with your H022 updates.

The changes are built on top of 4207a, not 3809e. If we need other changes from 3809e, we will pull it from there (i.e. the files from [#4424-10](#), if they will be needed at some point).

Constantin Asofiei wrote:

Roger Borrello wrote:

Sorry for the lack of response. Today was Thanksgiving in the US.

Happy Thanksgiving! :)

You have to be careful to work your 3809e changes into these changes. I see some conflicts with your H022 updates.

The changes are built on top of 4207a, not 3809e. If we need other changes from 3809e, we will pull it from there (i.e. the files from [#4424-10](#), if they will be needed at some point).

No problem. I just wanted to raise awareness that there are a couple of threads containing some of those files.

I am OOO today, as well, but wanted to take a quick look at this update. I did a full build, and came across another issue and have added a new testcase: `uast/table-handle_parameters/tablehandle_parameter_with_table.p`

This is another procedure with an input parameter that is a TABLE instead of TABLE-HANDLE. The result is:

```
[java] -----
[java] Code Conversion Annotations
[java] -----
[java]
[java] Optional rule set [customer_specific_annotations_prep] not found.
[java] ./abl/tablehandle_parameter_with_table.p
[java] Elapsed job time: 00:00:01.362
[java] EXPRESSION EXECUTION ERROR:
[java] -----
[java]   throwException("Missing 'refid' annotation")
[java]   ^ { Missing 'refid' annotation [DEFINE_PARAMETER id <244813135899> 4:4] }
[java] -----
[java] EXPRESSION EXECUTION ERROR:
[java] -----
[java]   currlist.add(execLib("get_classname", copy))
[java]   ^ { Expression execution error @1:1 }
[java] -----
[java] ERROR:
[java] com.goldencode.p2j.pattern.TreeWalkException: ERROR! Active Rule:
[java] -----
[java]           RULE REPORT
[java] -----
[java] Rule Type :    WALK
[java] Source AST: [ define ] BLOCK/PROCEDURE/BLOCK/STATEMENT/DEFINE_PARAMETER/ @4:4 {244813135899}
[java] Copy AST  : [ define ] BLOCK/PROCEDURE/BLOCK/STATEMENT/DEFINE_PARAMETER/ @4:4 {244813135899}
[java] Condition :   throwException("Missing 'refid' annotation")
[java] Loop      :   false
[java] --- END RULE REPORT ---
[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:562)
[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 @1:14 [DEFINE_PARAM
ETER id=244813135899]
[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 @1:14
[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.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.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.NamedFunction.execute (NamedFunction.java:450)
[java]     at com.goldencode.p2j.pattern.AstSymbolResolver.execute (AstSymbolResolver.java:712)
[java]     at com.goldencode.p2j.pattern.CommonAstSupport$Library.execLib (CommonAstSupport.java:1534)
[java]     at com.goldencode.expr.CE9131.execute (Unknown Source)
[java]     at com.goldencode.expr.Expression.execute (Expression.java:391)
[java]     ... 16 more
[java] Caused by: com.goldencode.p2j.pattern.CommonAstSupport$UserGeneratedException: Missing 'refid' annotation [DEFINE_PARAMETER id <244813135899> 4:4]
[java]     at com.goldencode.p2j.pattern.CommonAstSupport$Library.throwException (CommonAstSupport.java:2817)
[java]     at com.goldencode.p2j.pattern.CommonAstSupport$Library.throwException (CommonAstSupport.java:2802)
[java]     at com.goldencode.expr.CE9134.execute (Unknown Source)
[java]     at com.goldencode.expr.Expression.execute (Expression.java:391)
[java]     ... 28 more

```

#34 - 11/29/2019 10:43 AM - Constantin Asofiei

Roger Borrello wrote:

I am OOO today, as well, but wanted to take a quick look at this update. I did a full build, and came across another issue and have added a new testcase: uast/table-handle_parameters/tablehandle_parameter_with_table.p

This is another procedure with an input parameter that is a TABLE instead of TABLE-HANDLE. The result is:

Thanks, it looks like a regression from my changes. I'll take a look.

#35 - 12/02/2019 08:13 AM - Constantin Asofiei

This is fixed in 11347. Please test again.

#36 - 12/02/2019 09:01 AM - Greg Shah

Code Review Task Branch 4207a Revisions 11346-11347

Overall, the changes look good.

One question: in scope_promotion, what is the idea behind removing the promotion of the handle var being referenced? It seems like that var may in fact need to be promoted.

#37 - 12/02/2019 09:05 AM - Constantin Asofiei

Greg Shah wrote:

Code Review Task Branch 4207a Revisions 11346-11347

Overall, the changes look good.

One question: in scope_promotion, what is the idea behind removing the promotion of the handle var being referenced? It seems like that var may in fact need to be promoted.

That code was promoting the DEFINE PARAMETER - this must never happen, as it breaks recursivity. Also, that parmrefid annotation (at the referenced DEF VAR) was removed completely, I don't understand what it was trying to do - I think it was an early attempt at adding support for TABLE-HANDLE parameters.

#38 - 12/02/2019 09:05 AM - Greg Shah

OK, that works for me.

#39 - 12/02/2019 09:07 AM - Constantin Asofiei

And another reason why parmrefid was also incorrect: a var can be referenced by multiple parameters, not just one.

#40 - 12/02/2019 09:16 AM - Roger Borrello

Constantin Asofiei wrote:

This is fixed in 11347. Please test again.

Both testcases (tablehandle_parameter_with_for.p and tablehandle_parameter_with_table.p) convert and compile (as well as customer code).

Great job!

#41 - 03/03/2020 02:39 PM - Roger Borrello

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

#42 - 03/04/2020 10:28 AM - Greg Shah

- *Status changed from Test to Closed*