Database - Bug #6940

unqualified field name resolution when there is an explicit buffer defined in an internal procedure

11/17/2022 02:46 AM - Constantin Asofiei

Status:	Test	**Start date:**	
Priority:	High	**Due date:**	
Assignee:	Constantin Asofiei	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD		

Description

History

#2 - 11/17/2022 02:52 AM - Constantin Asofiei

- Assignee set to Constantin Asofiei

The problem is when a buffer is defined explicitly as a parameter - FWD doesn't resolve the unqualified field properly.

```
def temp-table tt1 field f1 as int.

procedure proc0.
   def parameter buffer btt1 for tt1.
   create btt1.
   assign f1 = 10.
end.

procedure proc1.
   def parameter buffer b for book.
   create b.
   assign isbn = "1234" book-id = 1234 book-title = "abc".
end.
```

will use 'book' and 'tt1' instead of 'b' and 'btt1' in the ASSIGN statement.

#3 - 11/17/2022 08:01 AM - Constantin Asofiei

Greg, the problem is that for a DEF PARAM BUFFER ... FOR some-buf statement, some-buf gets promoted because def_buf_stmt will promote a

buffer for any call other than DEF BUFFER ... FOR some-buf. statement. So, when CREATE is executed, it will promote btt1 in our example, but as tt1 was promoted, too, f1 is resolved from tt1.

This patch looks like it solves the problem:

```
### Eclipse Workspace Patch 1.0
#P p2j6129a
Index: src/com/goldencode/p2j/uast/progress.g
===================================================================
--- workspace.java.open.client/p2j6129a/src/com/goldencode/p2j/uast/progress.g    (revision 3910)
+++ workspace.java.open.client/p2j6129a/src/com/goldencode/p2j/uast/progress.g    (working copy)
@@ -14048,12 +14048,9 @@
  * Separated from the main rule to properly build the AST without complexity.
  * This is the only reason such a simple rule is implemented separately.
  *
- * @param    promote
- *           true to force promotion of the referenced table.
- *
  * @return   The name of the first record referenced.
  */
-simple_for_record_spec [boolean promote] returns [String rtext = null]
+simple_for_record_spec returns [String rtext = null]
    :
       {
          NameNode nothing = null;
@ -14069,7 +14066,7 @@
          :
          KW_TEMP_TAB { forceTemp = true; forcePersistent = false; }
       )?
-      nothing = r:record[promote, false, true]
+      nothing = r:record[false, false, true]


{
          forceTemp = old1;
@ -14132,7 +14129,6 @@
 def_buf_stmt [boolean define_stmt, boolean optionalSym, boolean importedShared, Aast am, Aast st]
    :
{
-         boolean promote = !define_stmt;
          String rtext = null;
       }
       (
@ -14148,7 +14144,7 @@
          )
-         ( options { generateAmbigWarnings = false; } : rtext=simple_for_record_spec[promote] )?
+         ( options { generateAmbigWarnings = false; } : rtext=simple_for_record_spec )?
          (
             options { generateAmbigWarnings = false; }
             :
```

I just disabled the 'promote' code for any kind of statement, be it a standalone buffer definition or a buffer parameter definition. Do you recall why this 'promote' was kept in the first place?

#4 - 11/17/2022 08:02 AM - Constantin Asofiei

The patch is built on top of 6129b/14324.

#5 - 11/17/2022 06:39 PM - Greg Shah

FYI, I'm digging back through the history on this one. I'll pick it back up in the morning.

#6 - 11/18/2022 12:56 PM - Constantin Asofiei

I think the intent was to promote the parameter buffer, and not the 'source' for the parameter buffer. For example, this is a compile error in OE:

```
procedure proc1.
   def buffer bb for book.
   message isbn.
end.
```

while this is will compile OK, as 'bb' is already 'found':

```
procedure proc1.
   def parameter buffer bb for book.
   message isbn.
end.
```

Adding promote only for parameter buffers seems to work:

```
### Eclipse Workspace Patch 1.0
#P p2j6129a
Index: src/com/goldencode/p2j/uast/progress.g
===================================================================
--- workspace.java.open.client/p2j6129a/src/com/goldencode/p2j/uast/progress.g    (revision 3910)
+++ workspace.java.open.client/p2j6129a/src/com/goldencode/p2j/uast/progress.g    (working copy)
@@ -14048,12 +14048,9 @@
   * Separated from the main rule to properly build the AST without complexity.
   * This is the only reason such a simple rule is implemented separately.
   *
- * @param    promote
- *           true to force promotion of the referenced table.
- *
   * @return   The name of the first record referenced.
   */
-simple_for_record_spec [boolean promote] returns [String rtext = null]
+simple_for_record_spec returns [String rtext = null]
     :
        {
           NameNode nothing = null;
@@ -14069,7 +14066,7 @@
           KW_TEMP_TAB { forceTemp = true; forcePersistent = false; }
        )?
-      nothing = r:record[promote, false, true]
+      nothing = r:record[false, false, true]
```

```
{
          forceTemp = old1;
@ -14132,7 +14129,6 @@
 def_buf_stmt [boolean define_stmt, boolean optionalSym, boolean importedShared, Aast am, Aast st]
     :
{
-         boolean promote = !define_stmt;
          String rtext = null;
       }
       (
@ -14148,7 +14144,7 @@
```

```
      | { optionalSym }?
                // empty alternative
           )
-          ( options { generateAmbigWarnings = false; } : rtext=simple_for_record_spec[promote] )?
+          ( options { generateAmbigWarnings = false; } : rtext=simple_for_record_spec )?
           (
              options { generateAmbigWarnings = false; }
              :
@ -14178,7 +14174,7 @@


// imported (non-new) shared buffers are the equivalent of a FIND or CREATE, we must
             // promote these buffers to ensure proper name disambiguation
-            if (importedShared)
+            if (importedShared || !define_stmt)
{
                // not sure about the noProp flag here
                sym.promoteTableName(#b.getText(), true, false);
```

#7 - 11/18/2022 01:14 PM - Greg Shah

The change was introduced in trunk revision 4785:

```
** 128 ECF 20060818 CHG   @28686 Added boolean parameter to record rule. If
**                              true, record's table is promoted, else not.
```

I'm still trying to relate this back to the original task/bug it was meant to fix. The JPRM number 28686 does not appear in Redmine. I see some possible tasks from that same time period which could maybe be related: #346 and #347.

#8 - 11/19/2022 08:16 AM - Constantin Asofiei

There is no change in the application used for task #346. In the customer's app, all changes look 'expected', the parameter buffer is used instead of the implicit one.

I've committed this change to 6129b/14327.

#9 - 11/21/2022 07:38 AM - Greg Shah

In the end there was no information in email, testcases/uast/, Redmine or bzr that could lead be back to the original reason for the change in rev 4785.

The whole "promote" concept with the SchemaDictionary is just a hack/workaround to simulate the buffer scoping at parse time which we calculate much later (in annotations). It doesn't match 1 for 1 with the proper buffer scoping, so this is really a nasty area where we have built patch on patch on patch. Since we have no proper testcases for checking this (yet), we can only go by th elogic of the change and the affect it has on the converted code.

#10 - 11/21/2022 07:40 AM - Greg Shah

- % Done changed from 0 to 100

- Status changed from New to Test

Code Review Task Branch 6129b Revision 14327

The concept of the change is sound. Combined with the fact that it did not cause a regression/change in the converted code for the original ChUI application that was fixed by the work in revision 4785, I think this change is reasonable and seems correct.

We can watch out for any impact on other applications, but I think this is OK.