Base Language - Bug #7301

pre-scan of a 'DEFINE BUFFER btt1 FOR btt1' from within a method

04/25/2023 07:06 AM - Constantin Asofiei

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Constantin Asofiei	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#2 - 04/25/2023 07:14 AM - Constantin Asofiei

In pre-scan, no scope is pushed for top-level blocks (as we can't track them). This case fails to parse in pre-scan:

```
class oo.PBuf:
   def temp-table tt1 field f1 as int.

   def private buffer btt1 for tt1.

   method public void m1():
      find first btt1.
   end.

   method public void m2():
      define buffer btt1  for btt1.

      find first btt1.
   end.
end.
```

because 'btt1' is the same as the target and source at the DEFINE BUFFER from m2() method.

The solution seems to be to allow 'same buffer' in SchemaDictionary.findTableFromNode, for pre-scan case:

```
--- old/6129c/src/com/goldencode/p2j/schema/SchemaDictionary.java
+++ new/6129c/src/com/goldencode/p2j/schema/SchemaDictionary.java
@@ -541,6 +541,9 @@
 **     CA  20230110          Let 'getImmediateChild' work with a set, if an array is used for the types.
 ** 101 CA  20230321          A TEMP-TABLE's signature must include index information (this includes the index
 **                           name, options, field position and type).
+** 102 CA  20230425          In pre-scan mode, a "DEFINE btt1 FOR btt1" from within a method fails lookup, as
+**                           there is no scope pushed for top-level blocks in pre-scan;  fixed this case by
+**                           allowing 'the same' buffer to be returned in 'findTableFromNode'.
 */

 /*
@@ -2801,6 +2804,12 @@
          fromNode = findEntry(fromTable, type, scopes.size() - 1, USER_GLOBAL_SCOPE, false);
       }

+      if (SymbolResolver.isPreScan() && fromNode == toNode)
+      {
+         // in pre-scan mode, there is no scope added for top-level blocks.
+         return fromNode;
+      }
+
       // Make sure the node we found is not our "to" node. This can happen if the "to" node is
       // named the same as the "from" node, and was not qualified to differentiate it when it was
```

```
       // added. Also ensure that the "from" node is not a buffer;  we need the backing table.
```

#3 - 04/25/2023 09:51 AM - Constantin Asofiei

The previous patch is not correct, it regresses this case:

```
   method public void m3(buffer btt1 for btt1):
   end.
```

the fix which works seems to be:

```
=== modified file 'src/com/goldencode/p2j/uast/progress.g'
--- old/src/com/goldencode/p2j/uast/progress.g  2023-03-31 19:04:00 +0000
+++ new/src/com/goldencode/p2j/uast/progress.g  2023-04-25 13:41:22 +0000
@@ -7551,6 +7551,12 @@
    {
       String name = symbol.getText();

+      if (SymbolResolver.isPreScan() && name.equalsIgnoreCase(recordName))
+      {
+         // in pre-scan mode, leave only the 'default' buffer and do not allow override with the same name.
+         return true;
+      }
+
       // the target scope is determined by whether we are currently
       // executing in an internal procedure, function or trigger in which
       // case bufferScope is false, otherwise it is true and the
```

I'm trying now a complete parsing of the customer's app and see where it goes.

#4 - 04/25/2023 02:37 PM - Constantin Asofiei

- Assignee set to Constantin Asofiei

Created task branch 7301a from trunk rev 14551.

The patch in previous note is in rev 14552.

#5 - 04/26/2023 03:56 PM - Greg Shah

Code Review Task Branch 7301a Revision 14552

It seems that this logic may be broken if the recordName is a qualified name. Otherwise I have no objections.

#6 - 05/01/2023 05:11 AM - Constantin Asofiei

Greg Shah wrote:

> It seems that this logic may be broken if the recordName is a qualified name. Otherwise I have no objections.

In 4GL, you can't DEFINE BUFFER b1 FOR b1. where b1 is DEFINE BUFFER b1 FOR p2j_test.book.. This hide/redefine works only for temp-table buffers, which can't be qualified AFAIK. I've added a comment related to this and committed the change to 7199c rev 14554

7301a was dead-archived.

#7 - 06/05/2023 10:30 AM - Constantin Asofiei

7199c was merged to trunk rev 14610 and archived

#8 - 06/05/2023 10:32 AM - Greg Shah

- Status changed from New to Closed