Bugs - Bug #1706

temp-table buffer scoping

10/29/2012 08:44 AM - Constantin Asofiei

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

Description

Related issues:

Related to Database - Feature #2047: field references in procedures without e...	**Closed**	**02/25/2013**

History

#1 - 10/29/2012 08:45 AM - Constantin Asofiei

Following are details about the temp-table buffer scoping task:

1. What is done:
Attached are the changes related to the temp-table scoping issue. At this time, I realized I missed the TemporaryBuffer.clear(...) case - this isn't handled by the conversion rules. The other case, the bulk delete with where clause which uses RecordBuffer.delete(dmo, where, ...), is handled properly. The TemporaryBuffer.clear() case should be solved similar to RecordBuffer.delete(dmo, where, ...) - TemporaryBuffer.clear(dmo) should be renamed to TemporaryBuffer.clearImpl(dmo) and another TemporaryBuffer.clean(boolean openScope, dmo) should be added, which will behave similar to the RecordBuffer.delete(boolean openScope, dmo, where, ...) version.

Attached are also tests related to buffer scoping, for perm and temp-tables. No differences were found between them.

2. Original description:
Testcases:

gso405-dummy-func*.p

Synopsis:

Buffer availability depends on the presence of statements which use temp buffer outside the repeat cycle. Output of gso405-dummy-func2.p: in 4GL - no, no. In P2J - no, yes.

Explanation:

The reason is simple: temp-tables have record buffers that are scoped in Progress, though we have not investigated it carefully. However, in P2J we always assign the record buffer scope to the external procedure. In the dummy 1 case, the extra procedure causes the scope of the buffer to be raised to the external proc and thus it works exactly the same as P2J. But in the dummy 2, it looks to me like the record buffer is scoped to the REPEAT block. Thus at the end of that block, the record is released and when you loop back around there is no record in the buffer, so it will always report "no" to avail().

I put together the buffer scoping in the early days of P2J. I have some notes at the top of annotations/record_scoping.rules:

<!-- temp-tables and work-tables are currently scoped to the external
procedure but this is NOT CORRECT -->

At some point I had a finding that led me to believe it was correct. Then later, I had reason to suspect it was not correct, but I didn't fix teh issue, I just put the note in the ruleset. :<

Until today we haven't known of a specific reason to fix this. This would require conversion changes. Whether that is easy or hard depends on if there are any differences in scoping between temp-tables and permanent tables. So we would need to test a range of cases to see. The way to do this is to use the 4GL code like this: "COMPILE testcase.p listing my_list.txt."

Then look inside my_list.txt at the bottom to see where the buffers were scoped.

Details on how buffer scoping works can be found in uast/package.html. Search for the section entitled "Record Scopes". It is not directly in the table of contents (it is in the Schema Names section). As far as which cases need to be checked, the best place to see this is in that section of the package.html, you will find a table that has a first column of "Progress Source Code" which is the testcase (currently written with permanent tables) and then a column of notes that describe the scoping behavior. You will also see that the scopes are color coded in the source code for easier viewing.

We need to know if we must fix this to fix GSO 405, If so, then you will work your way through each of those testcases (in the uast/package.html). You will change each one to use TTs instead of perm tables and then run a listing of it. Compare the results with what I have documented and make a list of the differences. If everything is the same, then we will be very lucky.

If this doesn't have to be fixed, we will probably postpone this work.

#2 - 11/01/2012 05:03 PM - Greg Shah

- File gso405_testcases.zip added

- File scope_test.4gl.zip added

#3 - 11/01/2012 05:11 PM - Greg Shah

- File deleted (gso405_testcases.zip)

#4 - 11/01/2012 05:11 PM - Greg Shah

- File deleted (scope_test.4gl.zip)

#5 - 11/01/2012 05:11 PM - Greg Shah

- File ca_test20091204a.zip added

#6 - 04/21/2016 08:37 AM - Paul E

I know this is a very old issue - is it still relevant?

How would we determine whether our code suffers from this problem?

#7 - 04/21/2016 08:51 AM - Greg Shah

It may not be an issue any more because we have made changes recently to temp-table scoping and buffer processing. But we have not considered it recently to determine if it is still an issue. I can say that we did not need to fix this for TIMCO and we don't know of any case that affects your apps.

We have hundreds of tasks in the system, meant to ensure that we capture known issues. Any such tasks that are known to affect your code are already assigned to milestones.

I don't have any easy way to authoritatively determine if this case can be hit in your code.

Files

ca_test20091204a.zip	142 KB	11/01/2012	Greg Shah