Database - Bug #6524

BUFFER-COPY does not create a record in the before-table

06/16/2022 01:16 PM - Constantin Asofiei

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

History

#1 - 06/16/2022 01:17 PM - Constantin Asofiei

BUFFER-COPY (statement and method) should create the record in the before-table:

```
def temp-table tt1 before-table btt1 field f1 as int.
def dataset ds1 for tt1.
def temp-table tt2 like tt1.

create tt2.
tt2.f1 = 123.
create tt1.

temp-table tt1:tracking-changes = true.
buffer-copy tt2 to tt1.
temp-table tt1:tracking-changes = false.

find first btt1.
message btt1.f1 row-state(btt1).
```

Ovidiu, please take a look ASAP.

#3 - 06/16/2022 01:17 PM - Constantin Asofiei

- Priority changed from Normal to High

#4 - 06/16/2022 03:59 PM - Ovidiu Maxiniuc

Constantin,

Normally, RecordBuffer.copy() checks whether the involved buffers have the conditions for copying the before images (line 4010, where boolean hasBefore local variable is set) and acts accordingly.

However, in this case, the operation is not possible. I modified your testcase and added a few lines, before enabling the tracking-changes attribute:

```
MESSAGE "tt1 table:"  TEMP-TABLE tt1:BEFORE-TABLE.
MESSAGE "tt1 buffer:" BUFFER tt1:BEFORE-BUFFER BUFFER tt1:BEFORE-ROWID.
MESSAGE "tt2 table:"  TEMP-TABLE tt2:BEFORE-TABLE.
MESSAGE "tt2 buffer:" BUFFER tt2:BEFORE-BUFFER BUFFER tt2:BEFORE-ROWID.
```

The output in 4GL is something like:

```
tt1 table: 7910
tt1 buffer: 7911 ?
tt2 table: ?
tt2 buffer: ? ?
```

What does it means? It means that the tt2 does not have a before-table, nor a before-buffer. So we have a problem.
Sincerely, I do not know where the create/modify operations are stored for tt2 in this case. More than that, how were these saved when the tracking-changes was NOT enabled at that moment for that table? Even stranger, you cannot do that without having the table (tt2) in a dataset or the 12355 error will be raised.

#5 - 06/16/2022 04:10 PM - Constantin Asofiei

Ovidiu, why are you talking about tt2 before-table? This is not about tt2, is about tt1 - this table has TRACKING-CHANGES set, so any change made in the after-table must be recorded in the before-table.

There may be cases where a before-record from the source table is copied to the dest table, but I don't know how to duplicate them. How confident are you that BUFFER-COPY (which uses explicit buffers!) copies the BEFORE-TABLE buffer, too?

#6 - 06/16/2022 04:49 PM - Ovidiu Maxiniuc

Constantin Asofiei wrote:

> Ovidiu, why are you talking about tt2 before-table? This is not about tt2, is about tt1 - this table has TRACKING-CHANGES set, so any change made in the after-table must be recorded in the before-table.

Yes, I am talking about tt2's BEFORE-TABLE as being the source in buffer-copy operation for tt1's BEFORE-TABLE (which is btt1).

> There may be cases where a before-record from the source table is copied to the dest table, but I don't know how to duplicate them. How confident are you that BUFFER-COPY (which uses explicit buffers!) copies the BEFORE-TABLE buffer, too?

Yes, if both the source and the destination have before-tables, they should be copied in FWD, too. I will be back soon with more info. But in the testcase you found, I do not know where OE takes the data which is copied. It seems that it keeps it in the background, the before-table is created and populated but not exposed to ABL programmer.

#7 - 06/16/2022 05:25 PM - Greg Shah

It seems to me the 4GL may not care if there is a before-table on the source temp-table (tt2). The before-table on the target temp-table (tt1) is enough because it is just there to track changes to that target temp-table. The changes come from tt1 instead of a 2nd buffer-copy from before-tt2 to before-tt1.

#8 - 06/16/2022 05:29 PM - Ovidiu Maxiniuc

Greg Shah wrote:

> It seems to me the 4GL may not care if there is a before-table on the source temp-table (tt2). The before-table on the target temp-table (tt1) is enough because it is just there to track changes to that target temp-table. The changes come from tt1 instead of a 2nd buffer-copy from before-tt2 to before-tt1.

You mean that by simply copying data from the tt2 (which does not have a before-table) to tt1, the before-table of tt1 is populated by the operations the buffer-copy implies? I will try to test that.

#9 - 06/16/2022 06:25 PM - Greg Shah

Yes. It seems to me this is not really about the buffer-copy itself. It is about the edits to tt1 which have to be tracked.

#10 - 06/16/2022 08:50 PM - Ovidiu Maxiniuc

- Status changed from New to WIP

- % Done changed from 0 to 100

It seems that the buffer copy operation was not very well understood at the moment it was implemented. The effort was to copy the buffer and the pair before at once. The above testcase and my changes to it proved it is much simpler, the before-buffer simply obeys the natural flow of data from the after-buffer.

Committed revision 13943.

#11 - 06/17/2022 06:11 AM - Constantin Asofiei

- Status changed from WIP to Test

Ovidiu, thank you, this solves the problem I was seeing.