Database - Bug #2524

missing support for NEXT-SIBLING attribute for buffer handles

02/24/2015 02:44 PM - Ovidiu Maxiniuc

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Ovidiu Maxiniuc	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	Cleanup and Stablization for Server Features		
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#1 - 02/24/2015 03:11 PM - Ovidiu Maxiniuc

- File t2524a.p added

I noticed the following message printed in the app-server 'client' log after each executed request.

```
    **PREV-SIBLING is not a queryable attribute for BUFFER widget. (4052)
```

The next part of the discussion took place via mail with Eric, so I am dumping it here.

> That's interesting. Must not be a doubly-linked list (or possibly, the 4GL documentation is incomplete), because, while NEXT-SIBLING is supported for buffer handle objects, PREV-SIBLING is not.
> So, the SESSION:FIRST-BUFFER attribute evidently holds the head of the list. How do buffer handles get added to the list (and removed from it)? Is it automatic?

True, this is a one-direction chain, starting at SESSION:FIRST-BUFFER. There is no PREV-SIBLING attribute defined for buffer handles.
From my tests, the management of this chain is fully automated. In fact the NEXT-SIBLING is a read-only attribute, the P4GL programmer cannot alter it directly, just iterate the list and call methods on handles.

> On the face of it, this finding still doesn't explain the original error message, which was for PREV-SIBLING. However, if you look at the implementations of HandleChain, I guess the messages are reversed for getNextSibling and getPrevSibling.

You are right again, the messages were reversed. In fact the BufferImpl wrongly implements the support for next/prev sibling. The correct way is that hasNextSibling to return true and hasPrevSibling to return false.

I attached a testcase I used to test how P4GL works with this chain. With the above mentioned (small) changes the test procedure works, but the result is not the same, the list of buffers are not identical. Here are the output compared:

P4GL	P2J
1 crt: newBook deleting newBook buffer 2 crt: newBook2 deleting newBook2 buffer 3 crt: b3s deleting b3s buffer	1 crt: tt0 2 crt: sb 3 crt: book 4 crt: b3s deleting b3s buffer

Short analysis:

- p4gl keeps dynamic temp-tables buffers (newBook & newBook2) defined in procedures that are already finished, the defined buffers survive in this list. We don't. Those buffers are lost in P2J as the procedure where they were created ends.

- I was not able to force p4gl to add to this list static buffers. Instead P2j will add both static temp tables (tt0), permanent table buffers (book) and buffers defined for those buffers (sb).
- the only common buffers from the list are those initialized with CREATE BUFFER ... FOR TABLE syntax.

#2 - 02/24/2015 03:14 PM - Ovidiu Maxiniuc

- Status changed from New to WIP

#3 - 02/25/2015 03:53 PM - Ovidiu Maxiniuc

The reference pdf states that the SESSION:FIRST-BUFFER is the head of the linked list of *"handles for the dynamic buffers. The table may be either a temp-table or a connected database, **in that order**"*.
Indeed, from my tests, we need to add to the list only the dynamic buffers. The static buffers are not visible in this list. The second important thing is the confirmation that the order of the buffers is the order of the declarations, except for pushing the temp-tables in front.
There are some difficulties at this moment:

- the BufferImpl s register the HandleChain, but at that moment the dynamic attribute is not yet available because the object is not yet fully constructed. However, not the handles should be added to BUFFER list but the RecordBuffer s created. We should add it to list when the buffer is set to dynamic;
- the sorting mechanism must be implemented: use a delimitation pointer, insert temp-table buffers before and permanent ones after. This requires an iteration of the list. Alternatively, we could use two list and after the last temp-table to jump automatically to first permanent table buffer. Since this algorithm must be used only on buffers (I don't know if other resources requires special sorting) the insertion method should be protected and specialized in the extended class.

#4 - 03/03/2015 12:26 PM - Ovidiu Maxiniuc

- File om_upd20150303c.zip added

- File t2524a.p added

The attached update fixes the management of dynamic buffer handles.

#5 - 03/03/2015 03:29 PM - Eric Faulhaber

- Target version set to Milestone 11

Code review 0303c:

You added a call to release() at TemporaryBuffer:3411, which seems unrelated to the remaining changes. Please explain why that change was made. Note that unless other parts of the runtime have been retrofitted to involve HandleChain in runtime code used by the regression test environment, this one change to TemporaryBuffer is the only change I can see which would require regression testing this update. Everything else is about dynamic database.

Please group the public method HandleChain.interlink with the other public methods in the class.

Everything else looks good to me. Nice work.

Constantin: please review the HandleChain changes.

Ovidiu: please rename the testcase to something more descriptive (you can keep "2524" as part of the name if you want) and check it into the testcases project (and note that new name here).

#6 - 03/03/2015 05:28 PM - Eric Faulhaber

One more thing: please make the BufferImpl constructor you added protected instead of public. As an implementation class, I want to expose as little of it to the outside world as necessary.

#7 - 03/04/2015 05:17 AM - Ovidiu Maxiniuc

- Target version deleted (Milestone 11)

Eric Faulhaber wrote:

> Ovidiu: please rename the testcase to something more descriptive (you can keep "2524" as part of the name if you want) and check it into the testcases project (and note that new name here).

committed to testcases at following location: uast/buffer/buffer-handle-chain.p

#8 - 03/04/2015 05:37 AM - Constantin Asofiei

Ovidiu, about HandleChain changes: the buffer-specific code I think should reside in overridden methods in the BufferImpl class.

#9 - 03/04/2015 05:43 AM - Ovidiu Maxiniuc

Constantin Asofiei wrote:

> Ovidiu, about HandleChain changes: the buffer-specific code I think should reside in overridden methods in the BufferImpl class.

I know, that was my first thought, too (see my last phrase from note3 above). However, the data that they use is private to HandleChain. I am now thinking how to meet-at-the-midle implementation. If you have any idea that should work, please let me know.

#10 - 03/04/2015 05:45 AM - Ovidiu Maxiniuc

- Target version set to Milestone 11

#11 - 03/04/2015 06:45 AM - Constantin Asofiei

Ovidiu Maxiniuc wrote:

> Constantin Asofiei wrote:
>
> > Ovidiu, about HandleChain changes: the buffer-specific code I think should reside in overridden methods in the BufferImpl class.
>
> I know, that was my first thought, too (see my last phrase from note3 above). However, the data that they use is private to HandleChain. I am now thinking how to meet-at-the-midle implementation. If you have any idea that should work, please let me know.

Why not add protected workers in HandleChain, to access the private fields? I.e. get/setLast/FirstResource for WorkArea.first/lastResource . You already have setters for the next/PrevSibling, what is missing is a getter which returns the reference directly (not wrapped in a handle instance).

#12 - 03/12/2015 01:11 PM - Eric Faulhaber

- Assignee set to Ovidiu Maxiniuc

#13 - 03/13/2015 03:25 PM - Eric Faulhaber

Ovidiu, please merge the 0303c update up to the latest revision of P2J.

#14 - 03/13/2015 04:23 PM - Ovidiu Maxiniuc

- File om_upd20150313a.zip added

The attached update is based on 10810 revision from bzr.
The interlink method specialized in BufferImpl will handle special sorting of buffers in the resource chain.

Note: the RecordBuffer contains the update for getLDBName() (returns null for aliases to unconnected databases). Please let me know if I should extract this small change as a separate incremental update.

#15 - 03/13/2015 04:46 PM - Eric Faulhaber

Code review 0313a:

- The change to RecordBuffer.getLDBName is OK to leave, but it will necessitate runtime regression testing.
- There is still buffer-specific code in HandleChain (e.g., PERMANENT_TABLE).
- You removed the call to release() from TemporaryBuffer, but you never addressed my question in note 5 above.
- Should BufferImpl.hasNextSibling return different results by instance, or is it meant to return a class-wide true or false? This is a genuine question, not a request to change the implementation.

The rest looks fine. Once you've addressed the above, please post a final update for Constantin's review.

#16 - 03/13/2015 05:53 PM - Ovidiu Maxiniuc

Eric Faulhaber wrote:

> Code review 0313a:
>
> - The change to RecordBuffer.getLDBName is OK to leave, but it will necessitate runtime regression testing.

OK

> - There is still buffer-specific code in HandleChain (e.g., PERMANENT_TABLE).

I understand. I will move the related code to BufferImpl.

> - You removed the call to release() from TemporaryBuffer, but you never addressed my question in note 5 above.

The call to release() was an attempt to fix the issue from respective if block (there is a large comment). I keep it in my testing environment breakpointed, but it was never hit since I running the investigation. Since I don't have a proof of its utility I was not intending to promote the change to review, I manually remove it when creating the updates. Accidentally, it slipped into 0313a.

> - Should BufferImpl.hasNextSibling return different results by instance, or is it meant to return a class-wide true or false? This is a genuine question, not a request to change the implementation.

Only dynamic buffers are stored in the HandleChain list. Normally, the buffers should know when they are dynamically created. However, the p2j implementation use the same constructors and only at a later moment, the builder will declare them as dynamic and call interlink() to put them in the list before returning the new resource. So buffers are not stored in chain when the HandleChain constructor is called (hasNextSibling should be false) only after they have been declared as dynamic (hasNextSibling should now return true).

> The rest looks fine. Once you've addressed the above, please post a final update for Constantin's review.

OK, thanks.

#17 - 03/16/2015 11:12 AM - Ovidiu Maxiniuc

- File om_upd20150316a.zip added

Constantin, please review the attached update.

#18 - 03/16/2015 02:35 PM - Constantin Asofiei

Ovidiu Maxiniuc wrote:

> Constantin, please review the attached update.

Only issue I see is that you have this System.out.println("\tInterlinking BUFFER " + permanentTable); on line 2873 - please comment/remove it.

As I understand from the code, only dynamic buffers are chained, correct?

#19 - 03/16/2015 02:48 PM - Ovidiu Maxiniuc

Constantin Asofiei wrote:

> Only issue I see is that you have this System.out.println("\tInterlinking BUFFER " + permanentTable); on line 2873 - please comment/remove it.

It was only for debugging purposes. I will remove it.

> As I understand from the code, only dynamic buffers are chained, correct?

Exactly. See OE reference, page 1455:
***Note**: Only dynamic buffers created with the CREATE BUFFER statement are chained on the SESSION system handle.*

Thanks, I will enqueue the update for regression testing.

#20 - 03/18/2015 09:57 AM - Ovidiu Maxiniuc

- File om_upd20150316b.zip added

The first run of testing returned success for ctr+c and a few issues for main.
I'm giving it a second chance: restarting the testing from main-regression only.

#21 - 03/24/2015 08:54 AM - Ovidiu Maxiniuc

- File om_upd20150324a.zip added

The update passed the regression testing. It was merged with bzr repo' and then committed back as revision 10820. It was distributed to team by email.

#22 - 03/24/2015 10:21 AM - Eric Faulhaber

- % Done changed from 0 to 100

- Status changed from WIP to Closed

#23 - 11/16/2016 12:06 PM - Greg Shah

- Target version changed from Milestone 11 to Cleanup and Stablization for Server Features

Files

t2524a.p	3.29 KB	02/24/2015	Ovidiu Maxiniuc
om_upd20150303c.zip	164 KB	03/03/2015	Ovidiu Maxiniuc
t2524a.p	5.5 KB	03/03/2015	Ovidiu Maxiniuc
om_upd20150313a.zip	165 KB	03/13/2015	Ovidiu Maxiniuc
om_upd20150316a.zip	165 KB	03/16/2015	Ovidiu Maxiniuc
om_upd20150316b.zip	165 KB	03/18/2015	Ovidiu Maxiniuc
om_upd20150324a.zip	166 KB	03/24/2015	Ovidiu Maxiniuc