

Database - Bug #7788

NPE when assigning query to browse

09/12/2023 03:47 AM - Alexandru Donica

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

History

#1 - 09/12/2023 04:42 AM - Alexandru Donica

This code works in 4GL, but after conversion it throws a NullPointerException.

```
DEFINE TEMP-TABLE test-table FIELD someId AS INTEGER .

DEFINE BROWSE notes-browse WITH SIZE 10 BY 10.

DEFINE FRAME fMain notes-browse WITH SIZE 10 BY 10.

DEF VAR new-query-handle AS HANDLE NO-UNDO.
DEFINE VAR q AS HANDLE NO-UNDO.
CREATE QUERY q.
notes-browse:QUERY = q.

notes-browse:QUERY:QUERY-CLOSE ().

CREATE QUERY new-query-handle.
new-query-handle:SET-BUFFERS (BUFFER test-table:HANDLE).
new-query-handle:QUERY-PREPARE ("FOR EACH test-table BY test-table.someId").

notes-browse:QUERY = new-query-handle.

notes-browse:QUERY:QUERY-OPEN ().
```

```
23/09/12 11:38:18.593+0300 | SEVERE | com.goldencode.p2j.util.TransactionManager | ThreadName:Conversation [0
000001A:bogus], Session:00000026, Thread:00000027, User:bogus | Abnormal end; original error: java.lang.NullPo
interException
    at com.goldencode.p2j.persist.PreselectQuery.execute(PreselectQuery.java:5348)
    at com.goldencode.p2j.persist.PreselectQuery.reposition(PreselectQuery.java:3250)
    at com.goldencode.p2j.persist.QueryWrapper.reposition(QueryWrapper.java:3126)
    at com.goldencode.p2j.ui.BrowseWidget.setQuery(BrowseWidget.java:2681)
    at com.goldencode.p2j.ui.BrowseWidget.setQueryAsHandle(BrowseWidget.java:2744)
    at com.goldencode.dataset.Start.lambda$execute$0(Start.java:46)
    at com.goldencode.p2j.util.Block.body(Block.java:636)
    at com.goldencode.p2j.util.BlockManager.processBody(BlockManager.java:8867)
    at com.goldencode.p2j.util.BlockManager.topLevelBlock(BlockManager.java:8503)
    at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:576)
    at com.goldencode.p2j.util.BlockManager.externalProcedure(BlockManager.java:549)
    at com.goldencode.dataset.Start.execute(Start.java:36)
```

#2 - 09/12/2023 07:26 AM - Ovidiu Maxiniuc

Nice catch!

It seems like the query from new-query-handle is not properly initialized. I did not run the testcase but it's strange that in case of the first query assigned to browse, the initialize() seems to have been called. But q OTOH, has not been completely defined yet (no buffers set, no prepared statement).

#3 - 09/12/2023 08:18 AM - Alexandru Lungu

That part is not quite relevant for the test-case. The following is also problematic:

```
DEFINE TEMP-TABLE test-table FIELD someId AS INTEGER .
DEFINE BROWSE notes-browse WITH SIZE 10 BY 10.

// without empty query initialization

DEF VAR new-query-handle AS HANDLE NO-UNDO.
CREATE QUERY new-query-handle.
new-query-handle:SET-BUFFERS (BUFFER test-table:HANDLE).
new-query-handle:QUERY-PREPARE ("FOR EACH test-table BY test-table.someId").
notes-browse:QUERY = new-query-handle. // at this point, reposition(1) is called, even if the query is not opened yet
notes-browse:QUERY:QUERY-OPEN ().
```

There is a deferQueryReposition flag, especially to avoid the eager reposition. However, this is set to:

```
// find out if we need deferred reposition for the query was not opened
deferQueryReposition = this.query == null &&
                        _dynamic()          &&
                        !((QueryWrapper) resource).isOpen().booleanValue();
```

There are two issues here:

- The browse is not dynamic, so _dynamic returns false. I guess we can remove the _dynamic check here?
- The existing query is null, **but** in the [#7788-1](#) note, the current query is not null. Yet, the defer is required.

From my POV, !((QueryWrapper) resource).isOpen().booleanValue() is the only condition that is relevant for deferQueryReposition.

#4 - 09/12/2023 10:32 AM - Ovidiu Maxiniuc

Alexandru Lungu wrote:

There are two issues here:

- The browse is not dynamic, so `_dynamic` returns false. I guess we can remove the `_dynamic` check here?
- The existing query is null, **but** in the [#7788-1](#) note, the current query is not null. Yet, the defer is required.

From my POV, `!((QueryWrapper) resource).isOpen().booleanValue()` is the only condition that is relevant for `deferQueryReposition`.

Yes, in the light of the new testcase you are right.

However, I think the condition might be more complex than that. It was added as part of #4124 implementation, based on specific testcase(s) constructed to fit the needs of that task. This was the only way since we did not have a proper specification sheet (except for the OE manual which is not always correct). So the implementation tried to cover the exception cases known/observed at that time.

What I want to say is probably there is some need for those conditions. Maybe they were tested individually and the actual operator is `||` instead of `&&`? We need further tests here.

#5 - 09/12/2023 11:10 AM - Alexandru Lungu

Maybe they were tested individually and the actual operator is `||` instead of `&&`?

This was exactly what I thought about first time.

We need further tests here.

I agree. I wanted to share my findings here on the `deferQueryReposition` conditional, just to know what the focus should be.

#6 - 09/12/2023 07:25 PM - Stanislav Lomany

Let me put my thoughts. Is it a browse issue? Yes and no.

First, if you try this code:

```
QueryWrapper.createQuery(newQueryHandle);
newQueryHandle.unwrapBufferCollection().setBuffers(new handle(buffer(testTable)));
newQueryHandle.unwrapQuery().prepare("FOR EACH test-table BY test-table.someId");
newQueryHandle.unwrapQuery().reposition(1); // NOT queryReposition(1)
```

It'll abend with the NPE above. This happens because unlike (all?) other queries, PreselectQuery doesn't check if a query is closed. Although I think no 4GL code can be converted into this, FWD framework shouldn't abend, especially because of a visually perfectly legit code.

Next, the browse. It silently calls reposition(1) for closed queries. I don't think that should happen.

And when we come to that, we really draw our attention to the deferQueryReposition flag. Sorry, I think I was the author of it and I don't know what is it for, I mean why we're preserving it as a class field. There potentially could be 4GL quirks, but I did testing and I haven't found anything as well there're no quirks mentioned in the comments. In BrowseWidget.setQuery you just check if the query is opened and reposition(1) now or call addOpenedListener otherwise and that's it. deferReposition parameter in BrowseWidget.setQuery is also not needed in this case.

#7 - 10/23/2023 10:34 AM - Alexandru Donica

I have returned to this issue. I tried modifying this piece of code:

```
deferQueryReposition = this.query == null &&
                        _dynamic()           &&
                        !((QueryWrapper) resource).isOpen().booleanValue();
```

into this:

```
deferQueryReposition = !((QueryWrapper) resource).isOpen().booleanValue();
```

I deleted the first 2 conditions, because:

- I did not see why the deferQueryReposition had to depend on the old query (being null). It seemed to me that it should be about the new query.
- In my test case, the browse is static, and deferQueryReposition should be true, so I tried removing the _dynamic() check.
- If the new query is not yet opened, the deferQueryReposition should be true regardless of the browse type or the previous query

These were my arguments (based on my initial thoughts after reading the code and these comments a few times) for my change. I also removed the deferReposition parameter in the setQuery(P2JQuery, boolean) method and only used the data member deferQueryReposition, and, when unbinding the old query, I no longer reset this data member.

The code no longer threw a NPE for my test.

Furthermore, I will create tests in 4GL with:

- combinations of static/dynamic browse with static/dynamic queries
- combinations of calls to set-browse-query(q1) and set-browse-query(q2) and open-query(q1) and open-query(q2) in different orders with different values to try to find bugs
- multiple openings of queries, if browse repositions or not, and other tests if I think of any.

I wrote this as a small update, and to see if any watcher familiar with this topic has anything to add/say to/about my reasonings, changes, and test ideas.

#8 - 10/23/2023 10:57 AM - Alexandru Donica

- Assignee set to Alexandru Donica