

Database - Bug #2344

invalid scoping for dynamic buffers

07/24/2014 07:45 AM - Stanislav Lomany

Status:	Closed	Start date:	07/24/2014
Priority:	Normal	Due date:	
Assignee:	Stanislav Lomany	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	Cleanup and Stablization for Server Features	case_num:	
billable:	No		
vendor_id:	GCD		
Description			

History

#1 - 07/24/2014 08:14 AM - Stanislav Lomany

Dynamic buffers for permanent tables should have global scope.

```
static DataModelObject createDynamicBufferForPermTable(Class dmoIface,
                                                       String bufferName,
                                                       Database database)
{
    ...
    int blockDepth = TransactionManager.getNestingLevel() - 1;
    DataModelObject buffer = (DataModelObject) RecordBuffer.define(bufClass,
                                                                    database.getName(),
                                                                    uniqueBufferName,
                                                                    bufferName,
                                                                    blockDepth); // scoped to current block in a
}

llBuffers
{
    ...
    RecordBuffer.openScopeAt(0, buffer); // global scope
    return buffer;
}
```

#2 - 07/24/2014 08:23 AM - Stanislav Lomany

```
private static Temporary createDynamicBufferForTempTable(Class dmoIface,
                                                         String bufferName,
                                                         RecordBuffer master,
                                                         TempTable tempTable,
                                                         int blockDepth)
{
```

```

...
Temporary temporary;
if (master != null)
{
    temporary = TemporaryBuffer.define((Temporary) master.getDMOProxy(),
                                       uniqueBufferName,
                                       bufferName,
                                       blockDepth);
}
else
{
    temporary = TemporaryBuffer.define(bufClass,
                                       uniqueBufferName,
                                       bufferName,
                                       blockDepth == 0,
                                       undoable); // SPECIFY blockDepth?
}
TemporaryBuffer temporaryBuffer = ((TemporaryBuffer) RecordBuffer.get(temporary));
temporaryBuffer.setDynamic();
temporaryBuffer.setTempTableRef(tempTable);
TemporaryBuffer.openScopeAt(blockDepth, temporary);

if (tempTable._dynamic())
{
    DynamicTablesHelper.getInstance().registerBufferForDynamicTable(temporaryBuffer);

    // Add the target buffer to the set of all buffers at top block. This is only used by
    // dynamically defined temp-table buffers because they must be ready available in
    // the current block depth without waiting for an open scope for moving from from
    // prepend set (yet, the buffer is still contained in pending buffers).
    bufferManager.registerBuffer(temporaryBuffer); // WORKAROUND FOR SOME SPECIFIC CASE
}

return temporary;
}

```

#3 - 07/25/2014 06:20 AM - Stanislav Lomany

- File svl_upd20140725a.zip added

Fixes my testcases, #2329 and parameter_code_search (I guess #2331 is about that?). Note that I've altered

```
** 224 ECF 20140703      Use current block depth in createDynamicBufferForPermTable instead of
**                      -1. This ensures buffer is registered properly with buffer manager.
```

and

```
** 097 OM 20140417      Dynamic temp-table buffers are directly available at the moment they
**                      are declared instead of waiting for next block to open.
```

But I'm not sure what to tests run in order to be sure that I've not regressed these solutions.

#4 - 07/25/2014 04:06 PM - Eric Faulhaber

Stanislav Lomany wrote:

Dynamic buffers for permanent tables should have global scope.

Are you sure about this? How did you confirm this?

#5 - 07/25/2014 04:25 PM - Eric Faulhaber

Pending a reply to my last question...

Stanislav Lomany wrote:

Fixes my testcases, #2329 and parameter_code_search (I guess #2331 is about that?).

Early on in #2331 (note 1), the department_code_create API exhibited similar/related behavior, which I addressed with my 224 change in RecordBuffer.

Please list your test cases -- I'm working on an issue related to buffer scope (static and dynamic) at the moment, so these are of particular interest.

But I'm not sure what to tests run in order to be sure that I've not regressed these solutions.

Make sure the department_code_create API is not regressed (I don't see how it would be from this change). There is still an issue with the output of

that API (the primary reason for #2331), but Vadim is working on that problem.

With respect to the RecordBuffer change you are proposing, I can see why it fixes the failures, but setting the blockDepth to 0 (global scope) seemed overly aggressive to me. Hence, the question in the last note.

#6 - 07/25/2014 04:26 PM - Stanislav Lomany

Dynamic buffers for permanent tables should have global scope.

Are you sure about this? How did you confirm this?

E.g. I've created a buffer in some non-top-level external procedure, and it will be available in the top-level external procedure.

#7 - 07/25/2014 04:27 PM - Greg Shah

Please check your testcases into testcases/uast/dynamic_buffer/.

#8 - 07/25/2014 04:47 PM - Stanislav Lomany

I use

```
uast/dynamic_tables/dbuf-double-pending.p
uast/dynamic_tables/dbuf-scope1.p
```

I checked them in that directory because all dbuf-* testcases reside there.

#9 - 07/25/2014 04:53 PM - Greg Shah

Good.

#10 - 07/25/2014 05:49 PM - Stanislav Lomany

Make sure the department_code_create API is not regressed

Am I missing some files or do you mean 4) Department CREATE.xml?

#11 - 07/25/2014 05:52 PM - Eric Faulhaber

Stanislav Lomany wrote:

Make sure the department_code_create API is not regressed

Am I missing some files or do you mean 4) Department CREATE.xml?

That's the right one, my mistake earlier.

#12 - 07/25/2014 05:53 PM - Eric Faulhaber

Are you getting this after running dbuf-scope1.p?

```
warning - pcondb for db p2j_test has lkctr -1 (395)
```

#13 - 07/25/2014 05:57 PM - Stanislav Lomany

Yes. I ignored it:

Cause An internal schema lock that is released was never taken out.

Resolution Upgrade to OpenEdge 10.1C02.

Workaround The message can be suppressed with the -nolkctr startup parameter within the client session.

#14 - 07/25/2014 06:09 PM - Stanislav Lomany

4) Department CREATE output:

```
<error source="sPApiUpd_ValidateAIRRecord adeptau0.p" subType="Invalid Data Value" type="Validation Failure">Department Type "J" is invalid.</error>
```

#15 - 07/26/2014 02:43 PM - Eric Faulhaber

Code review 0725a:

Looks good. Please note that Vadim has a change to RecordBuffer for #2331 (vmn_upd20140723a.zip) which is currently under regression test. Once that has passed testing and is checked in, please merge your update with that version and regression test.

Vadim: once your update passes regression testing, please check it in and post the bzd revision to #2331 -- Stanislav is a watcher on that issue.

#16 - 07/26/2014 02:52 PM - Eric Faulhaber

Stanislav Lomany wrote:

4) Department CREATE output:

[...]

You have to run "1) Department Type CREATE.xml" as a prerequisite, which will create the department type "J" to which the error message refers.

See <app_name>_api_xml_tests_explanation_20140617.doc in #2266.

#17 - 07/30/2014 10:04 AM - Stanislav Lomany

- Status changed from WIP to Review

Regression-tested. Committed to bzd rev 10586.

#18 - 07/30/2014 10:46 AM - Eric Faulhaber

- % Done changed from 0 to 100

- Status changed from Review to Closed

#19 - 11/16/2016 12:06 PM - Greg Shah

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

Files

svl_upd20140725a.zip	122 KB	07/25/2014	Stanislav Lomany
----------------------	--------	------------	------------------