

Database - Feature #1662

runtime support for table parameters

10/22/2012 12:30 AM - Eric Faulhaber

Status:	Closed	Start date:	07/17/2013
Priority:	Normal	Due date:	07/22/2013
Assignee:	Stanislav Lomany	% Done:	100%
Category:		Estimated time:	32.00 hours
Target version:	Runtime Support for Server Features	version:	
billable:	No		
vendor_id:	GCD		
Description			

History

#1 - 10/22/2012 12:30 AM - Eric Faulhaber

Needs research: is it just a handle or is there some special DB processing needed at the transition point?

#2 - 10/31/2012 04:02 PM - Greg Shah

- Target version set to Milestone 7

#3 - 04/26/2013 09:25 AM - Eric Faulhaber

- Subject changed from support temp-table handles as parameters to runtime support for table parameters

- Due date set to 07/18/2013

- Assignee set to Stanislav Lomany

- Start date set to 07/15/2013

- Estimated time changed from 24.00 to 32.00

#4 - 05/01/2013 06:16 PM - Eric Faulhaber

- Due date changed from 07/18/2013 to 07/22/2013

- Start date changed from 07/15/2013 to 07/17/2013

#5 - 12/05/2013 03:55 PM - Stanislav Lomany

Does it include TABLE-HANDLE parameters?

#6 - 12/05/2013 04:02 PM - Eric Faulhaber

Yes.

#7 - 12/11/2013 07:40 AM - Stanislav Lomany

Eric, I'm still investigating the question in order to provide exact list of required changes, but at this point it is obvious that:

for TABLE parameters:

- Append on the calling side doesn't work (but conversion part is OK).
- Constraints violation error on table append doesn't handled properly.

TABLE-HANDLE parameters are NOT supported at runtime.

- Passing a handle for static tables shouldn't be a problem.
- For dynamic tables we have to implement runtime part, i.e. create a copy of the passed table and copy data if necessary from/to the source table.
- Problems inherited from the TABLE parameter issues.

#8 - 12/12/2013 08:18 AM - Stanislav Lomany

For M7 we have to implement table-handles for static tables, append on both sides. I guess fixing constraints violation also should be done. I'm trying to figure out if we should implement table-handle for dynamic tables, it is the most resource-demanding part.

#9 - 12/12/2013 02:25 PM - Stanislav Lomany

I've found cases where dynamic tables are passed, so all problems noted above should be fixed. And I need to finish the extensive set of testcases. Should I work on the issues after the testcases will be finished?

#10 - 12/12/2013 02:32 PM - Eric Faulhaber

Yes.

#11 - 12/18/2013 04:56 AM - Stanislav Lomany

- File *testcases_external_proc.zip* added

Attached testcases for external procedures. The plan is:

1. Implement/fix all issues for static temp tables (e.g. add support for static temp table handles).
2. Implement all features for dynamic tables.
3. Port testcases for internal procedures and functions.
4. Check behavior for internal procedures and functions.

#12 - 12/30/2013 11:32 AM - Stanislav Lomany

Eric, TemporaryBuffer.copyAllRows which is supposed to perform record copy upon calling a procedure which uses TABLE/TABLE-HANDLE parameters doesn't work for extents. Basically, at this point we cannot determine original field order when it comes to extents. E.g.

```
def temp-table tt1 field f1 as integer extent 3
                  field f2 as integer extent 3
                  field f3 as char.
```

and

```
def temp-table tt1 field f1 as integer extent 3
                  field f3 as char
                  field f2 as integer extent 3.
```

are indistinguishable in terms of field order. We can rely on the getters/setters order in DMO interface/implementation class, but it seems too implicit to me and, say, adding field index option to LegacyField annotation seems more appropriate.

Also, I haven't look into the customer code to check if temp tables with extents are used, but considering the amount of TABLE/TABLE-HANDLE usage it may not be easy to give a definitive answer.

Any ideas?

#13 - 01/01/2014 11:33 PM - Eric Faulhaber

We already have an annotation like this for [#2134](#), but currently it is only for extent fields which have been expanded with custom hints. You can adapt this according to your needs. If you take part of the code from that issue and integrate it here, please notify Vadim accordingly.

In general, I am OK with your approach, BUT, we need to have the primary functionality for this issue tested and checked in before you leave for vacation. If this piece (which is somewhat of an edge case, but possibly in use) has to wait until you get back, that is preferable to delaying the primary functionality.

#14 - 01/04/2014 05:04 PM - Stanislav Lomany

Eric, please consider OUTPUT TABLE-HANDLE parameters, like this:

```
define output parameter table-handle th.  
...  
th = temp-table some-table:handle.
```

When this procedure ends we should copy data from th to the new or existing table on the calling side. Note that unlike the TABLE parameter case we don't know what table will be selected as the output table. The way that seems most appropriate to me is to register a top-level finalizable that will perform copy when the called procedure ends. However the problem occurs if a static temp table is selected for output. Its scope can be closing at the same level so the buffer scope can be closed before our finalizable has copied the data. The ideas that I have:

1. Create prioritized finalizables.
2. Track all static table scoped to the same external procedure.
3. Technically finalizables are stored in the list so if I move `TemporaryBuffer.createDynamicTable` call (kind of associate for table handles) before `openScope` calls it will work.

Any ideas?

#15 - 01/05/2014 01:48 PM - Eric Faulhaber

Stanislav Lomany wrote:

[...]

The ideas that I have:

1. Create prioritized finalizables.
2. Track all static table scoped to the same external procedure.
3. Technically finalizables are stored in the list so if I move `TemporaryBuffer.createDynamicTable` call (kind of associate for table handles) before `openScope` calls it will work.

Any ideas?

I guess I like option 3, as it is least invasive (does not require additional data structures to be maintained). However the javadoc for `TransactionManager`'s finalizable registration methods does not make explicit the contract about the order in which notifications are performed with respect to the order in which objects are registered. Since you are now relying on behavior which is otherwise encapsulated in the TM, please update the javadoc accordingly, so this contract is explicit and can be relied upon by clients of TM.

#16 - 01/06/2014 02:13 PM - Stanislav Lomany

- File *svl_upd20130106a.zip* added

TABLE/TABLE-HANDLE parameters work with the following exceptions:

1. extents copying doesn't work;
2. thparm-invalid-table and thparm-table-mismatch testcases do not pass;
3. I haven't checked it for internal procedures and functions, but I assume it will work for the most part.

#17 - 01/07/2014 02:43 AM - Stanislav Lomany

- File *svl_upd20140107a.zip* added

Merged with the latest P2J.

#18 - 01/07/2014 01:01 PM - Eric Faulhaber

- % Done changed from 0 to 80

- Status changed from New to Test

#19 - 01/08/2014 01:46 PM - Eric Faulhaber

- File *svl_upd20140108a.zip* added

Code review 20140107a:

- I was a bit confused by the deferred copy error processing in TemporaryBuffer, with the corresponding change in TM. Please explain the reason this is needed.
- In StaticTempTable, many of the newly implemented methods are documented to return true on success, but you are displaying an error, then returning new logical() (unknown value). Should these be returning false?
- Other methods in StaticTempTable which are documented not to work on static temp tables throw UnsupportedOperationException. Is this because they are not expected to be reached for static tables, or is there a Progress-style error that should be raised here instead?
- You have overridden StaticTempTable.equals(Object), but you have not added a corresponding, overridden hashCode method. It may be that this class is not used in hash tables, but please follow the best practice of implementing a compatible hashCode method, just in case.

The attached update has an updated P2JIndex class, merged with the latest version in bsr. I do not plan to re-run regression testing, since the changes in P2JIndex were minor and did not conflict. I have committed this update to bsr (rev 10431) because apparently it works with your test cases and has passed regression testing, but please address the above questions/issues.

#20 - 01/25/2014 08:06 PM - Eric Faulhaber

- Status changed from Test to Closed

- % Done changed from 80 to 100

#21 - 01/28/2014 01:38 PM - Stanislav Lomany

- I was a bit confused by the deferred copy error processing in TemporaryBuffer, with the corresponding change in TM. Please explain the reason this is needed.

This code is intended to avoid unnecessary logging if a copy error occurs. If logging is OK then I can just throw an exception which will be caught in TM.processFinalizables.

#22 - 02/03/2014 09:49 AM - Stanislav Lomany

Issue noted in note 4 in [#2207](#) also affects table parameters. The only difference is that the records can't be partially copied, so the copy order doesn't matter. Testcase:

test.p:

```
def temp-table tt
  field f1 as integer
  field f2 as integer
  index idx1 is unique f1.
```

```
create tt.
tt.f1 = 55.
tt.f2 = 55.
```

```
run test2.p (output table tt append) no-error.
```

```
form tt with frame f1 10 down.
for each tt:
  display tt with frame f1.
  down with frame f1.
end.
```

test2.p:

```
def temp-table tt
  field f1 as integer
  field f2 as integer.
```

```
def output parameter table for tt.
```

```
create tt.
tt.f1 = 1.
tt.f2 = 3.
```

```
create tt.
tt.f1 = 1.
tt.f2 = 4.
```

#23 - 02/03/2014 09:51 AM - Stanislav Lomany

Also, NO-ERROR option is not properly handled.

#24 - 11/16/2016 11:42 AM - Greg Shah

- Target version changed from Milestone 7 to Runtime Support for Server Features

Files

testcases_external_proc.zip	13 KB	12/18/2013	Stanislav Lomany
svl_upd20130106a.zip	233 KB	01/06/2014	Stanislav Lomany
svl_upd20140107a.zip	233 KB	01/07/2014	Stanislav Lomany
svl_upd20140108a.zip	234 KB	01/08/2014	Eric Faulhaber