# Database - Bug #8703

## NO-UNDO option at DMO must be removed

04/29/2024 10:51 AM - Constantin Asofiei

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

**Description**

**Related issues:**

| | |
|---|---|
| Related to Database - Bug #8593: H2 performance improvement for table 'trunca... | **Internal Test** |

### History

**#2 - 04/29/2024 10:56 AM - Constantin Asofiei**

*- Related to Bug #8154: refactor scopeable stacks added*

**#3 - 04/29/2024 10:56 AM - Constantin Asofiei**

*- Related to deleted (Bug #8154: refactor scopeable stacks)*

**#4 - 04/29/2024 10:56 AM - Constantin Asofiei**

*- Related to Bug #8593: H2 performance improvement for table 'truncate by multiplex' added*

**#5 - 04/29/2024 10:59 AM - Constantin Asofiei**

Assume you have this test:

```
def temp-table tt1 no-undo field f1 as int index ix1 f1.
def temp-table tt2 field f1 as int index ix1 f1.
def var k as int.
def var c as int.

do transaction:

   do k = 1 to 10000:
      create tt1.
      tt1.f1 = k.
      release tt1.
      create tt2.
      tt2.f1 = k.
      release tt2.
   end.

//   for each tt1: delete tt1. end.
   for each tt2: delete tt2. end.
   undo, leave.
end.

c = 0.
for each tt1: c = c + 1. end.
message c.

c = 0.
for each tt2: c = c + 1. end.
message c.
```

and another test which just defines:

```
def temp-table tt1 field f1 as int index ix1 f1.
```

Depending on the order of processing these two programs in the conversion, the 'noUndo' option at the DMO annotation may or may not be set.

The problem is the Table.noUndo annotation must be removed - currently this option is emitted at the temp-table DMO definition, and 4GL does not 'discriminate' between no-undo and undo temp-table buffers when passing them as a BUFFER parameter (see #8700-8 and #8700-7).

The DmoMeta.noUndo option needs to be refactored so that it uses the actual temp-table instance/resource in FWD to check this state, and not the DMO annotation. We already emit this no-undo flag when defining the master/default buffer for the temp-table.

**#6 - 04/29/2024 02:32 PM - Ovidiu Maxiniuc**

Constantin Asofiei wrote:

> The DmoMeta.noUndo option needs to be refactored so that it uses the actual temp-table instance/resource in FWD to check this state, and not the DMO annotation. We already emit this no-undo flag when defining the master/default buffer for the temp-table.

This seems like a problem. The attribute is currently "baked" into the temp-table DMO. This means that, depending on the order of precessing the external procedures, only one value will prevail.

To fix this, the attribute must be removed from the DMO, and added locally to buffer definition for each procedure, like this:

```
   TT001_1_1.Buf tt001 = TemporaryBuffer.define(TT001_1_1.Buf.class, "tt001", "tt001", /*global:*/ false, 
/*undoable:*/ true);
```

Note that this method already exists. We only need to emit the right parameter. And test whether it works correctly.

**#7 - 04/29/2024 02:34 PM - Constantin Asofiei**

Ovidiu Maxiniuc wrote:

> Note that this method already exists. We only need to emit the right parameter. And test whether it works correctly.

Yes, this works, but runtime in DmoMeta is dependent on the Table.noUndo annotation, which is incorrect (thus the title 'remove the NO-UNDO option at DMO').

**#8 - 04/29/2024 03:16 PM - Ovidiu Maxiniuc**

Super cool if it works.

As I wrote in #8703-6 I agree that "the attribute must be removed from the DMO". Luckily, there are only two usages of this member:

- DMOSignatureHelper.java
- TempTableHelper.java

The problem is how do we acquire the information at that moment, since the both places receive more or less the same DmoMeta as parameter?