

Database - Bug #2090

correct sorting of DMO properties during schema conversion

03/13/2013 02:14 AM - Eric Faulhaber

Status:	Closed	Start date:	05/23/2013
Priority:	Normal	Due date:	05/24/2013
Assignee:	Constantin Asofiei	% Done:	100%
Category:		Estimated time:	8.00 hours
Target version:	Runtime Support for Server Features	case_num:	
billable:	No		
vendor_id:	GCD		
Description			

History

#1 - 05/01/2013 07:40 PM - Eric Faulhaber

When using the DataModelWorker originally developed during the Majic conversion project with a newer project, we hit an error during schema conversion that our DMO property sorting algorithm in that class did not follow the required collections sorting contract. This broke schema conversion. To resolve this, I updated DataModelWorker.PropertyComparator.compare to sort deterministically. This fixed the immediate issue, but there are two potential problems with this fix:

- I don't think the new sorting (or the old, for that matter) is compatible with the way Progress orders fields. This is potentially important when displaying or updating entire buffers (leaving the frame management and field placement to the 4GL), rather than specifying particular fields in a form.
- It did not pass conversion regression testing with Majic. That is, DMO interfaces and implementation classes had their methods emitted in a different order with the fix than before. This also caused differences in business logic (code related to frames if I recall).

#2 - 05/02/2013 02:56 AM - Constantin Asofiei

- File src.diff added

Although the runtime passed, I've double-checked the generated sources, and it does affect frame definitions/display in these programs:

```
diff -r generated/20130430a/src/aero/timco/majic/dan/Dec.java generated/20130502a/src/aero/timco/majic/dan/Dec.java
diff -r generated/20130430a/src/aero/timco/majic/train2/RstDspFrm.java generated/20130502a/src/aero/timco/majic/train2/RstDspFrm.java
diff -r generated/20130430a/src/aero/timco/majic/ui/dan/DecFrame0.java generated/20130502a/src/aero/timco/majic/ui/dan/DecFrame0.java
diff -r generated/20130430a/src/aero/timco/majic/ui/train2/RstDspFrmTrigger2Fout.java generated/20130502a/src/aero/timco/majic/ui/train2/RstDspFrmTrigger2Fout.java
```

In Dec.java, the problem is that in a definition like:

```
def temp-table tt like emp field other as int.
```

the "other" field is added first instead of last, when "display tt." is called.

In RstDspFrm.java, the problem is again related to a temp-table defined with explicit fields this time, but the order doesn't look right neither in the original or new frame definition, when the temp-table is used in a "display tt." statement.

So, we are not out of the woods yet with this one. The full src diff is attached.

#3 - 05/02/2013 12:21 PM - Eric Faulhaber

- Estimated time set to 8.00
- Due date set to 05/06/2013
- Assignee set to Constantin Asofiei
- Target version set to Milestone 7

#4 - 05/22/2013 10:59 AM - Constantin Asofiei

The remaining problem here is that this fix changes the widget order for some "display temp-buffer." cases in MAJIC. We need to confirm this is a real problem and not a false alarm.

#5 - 05/22/2013 12:50 PM - Greg Shah

How much time is estimated to be needed to check this?

#6 - 05/23/2013 08:03 AM - Constantin Asofiei

- File ecf_test20130223a.zip added

Attached the DataModelWorker changes.

#7 - 05/23/2013 08:15 AM - Constantin Asofiei

I can confirm this is a real problem. This is the case in dan/Dec.java:

```
def temp-table tt like book
    field ex as log.

form tt.
view.
```

- without the update, the "ex" field is put last.
- with the update, the "ex" field is put first.

I think this should be the case from train2/RstDspFrm.java:

```
def temp-table tt
    field bb like book.book-title
    field f1 as int
    field f2 as log
    field a1 as dec.

form tt.
view.
```

- without the update, "bb" field is first
- with the update, "bb" field is last

Eric, I think I know where the problem is. In the generated .p.schema file, the fields defined with "like" have an "order" annotation. The fields defined explicitly do not have an "order" annotation. For the temp-table case, I think 4GL sets a new "order" value for each field, depending on its position at the temp-table definition. Idea is, we should not inherit the "order" from the target field when "like" is used, but instead create new "order" annotations based on the field's position.

The 8-hour estimation for solving this should be enough.

#8 - 05/23/2013 08:27 AM - Greg Shah

Good work.

Please go ahead and work this. Right now, the code as we have it checked into bazaar can't properly convert the customer app code. That is a real problem that we need to resolve ASAP.

#9 - 05/23/2013 12:54 PM - Constantin Asofiei

- Status changed from New to WIP

- Start date set to 05/23/2013

- Due date changed from 05/06/2013 to 05/24/2013

#10 - 05/23/2013 04:17 PM - Constantin Asofiei

- File *ca_upd20130523a.zip* added

Please see the attached fix. At this time, I think the DataModelWorker solved the symptom, and not the cause. Idea is, for temp-tables, there were cases when the "order" annotation was missing (for new fields) or inherited from the persistent table (for "like" fields or "like table" clause). The fix adds explicit "order" annotation to all temp-table fields, calculated using their natural order, from the temp-table definition. With this change, I think the DataModelWorker change can be discarded, as the "order" annotation needs to be mandatory.

#11 - 05/24/2013 01:46 AM - Constantin Asofiei

There are two issues with this update:

1. I left behind the ORDER nodes
2. I expected for the parser, in a define temp-table like source-table. case, to preserve the order of the source-table's fields, which is not the case. I think I need to move this solution at the parser, when LIKE and FIELD clauses are processed.

#12 - 05/24/2013 06:10 AM - Constantin Asofiei

- File *ca_upd20130524d.zip* added

This fix has passed MAJIC conversion testing; there are diffs, but they are expected. I think this fixes the DMO-related part of this note in #2104, comment 31:

I have also seen that DMO impl files and frame definitions cannot be easily compared between 2 different systems (e.g. devsrv01 vs staging), because of ordering issues in those files. Within a single system (e.g. just devsrv01), the only consistent different is the create alias. I prefer to fix all of these issues if possible, but the create alias is the most important.

Now I'm running runtime conversion testing with it. After this, I will put the server project under conversion.

#13 - 05/24/2013 04:21 PM - Constantin Asofiei

ca_upd20130524d.zip has passed runtime regression testing. I'll run full server project conversion next.

#14 - 05/25/2013 03:22 AM - Constantin Asofiei

The server project compiled converted fully (no schema generation errors). Lots of changes, I'm running conversion again with revision 10334 + the fix, to isolate the changes produced by this fix and be able to compare with the M4 generated sources.

#15 - 05/25/2013 09:01 AM - Constantin Asofiei

Beside renumbering of the TempRecord DMOs (not sure why, but I think this should not be a problem), changes related to this bug are in only 4 files. I've checked them all and the generated code is OK now (previous generated code had the fields in incorrect order).

Please take a look and if the fix looks good to you, I'll release it.

#16 - 05/25/2013 09:12 AM - Constantin Asofiei

- Status changed from WIP to Review

#17 - 05/27/2013 04:49 PM - Eric Faulhaber

The changes look good to me; please commit. Nice work!

#18 - 05/28/2013 02:55 AM - Constantin Asofiei

Committed to bazaar revision 10354.

#19 - 05/28/2013 10:59 AM - Eric Faulhaber

- Status changed from Review to Closed

- % Done changed from 0 to 100

#20 - 11/16/2016 11:43 AM - Greg Shah

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

Files

src.diff	588 KB	05/02/2013	Constantin Asofiei
ecf_test20130223a.zip	6.81 KB	05/23/2013	Constantin Asofiei
ca_upd20130523a.zip	5.34 KB	05/23/2013	Constantin Asofiei
ca_upd20130524d.zip	50.4 KB	05/24/2013	Constantin Asofiei