Database - Feature #2618

Feature # 2619 (Closed): misc database features needed for a GUI app

add some buffer/query/temp-table attributes

07/25/2015 08:44 AM - Greg Shah

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Ovidiu Maxiniuc	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	GUI Support for a Complex ADM2 App		
billable:	No	**vendor_id:**	GCD

Description

History

#1 - 07/25/2015 08:47 AM - Greg Shah

Add support (conversion and runtime) for the following:

BUFFER:NUM_FIELDS
QUERY:SKIP-DELETED-RECORD
TEMP-TABLE:UNDO

#2 - 07/25/2015 08:48 AM - Greg Shah

- Parent task set to #2619

#3 - 10/09/2015 12:10 PM - Ovidiu Maxiniuc

- Status changed from New to WIP

- Assignee set to Ovidiu Maxiniuc

Current status:

- BUFFER:NUM_FIELDS. Looking over the P2J code, it seems to be fully implemented.
- QUERY:SKIP-DELETED-RECORD. Conversion is implemented (at least in 2272b/10988). Because the doc is almost non-existent, I did some testcase to understand the impact of this attribute of the query. Unfortunately, I was not able too see any difference. If a record is deleted, the query will never iterate it, regardless of this attribute value. One version of this testcase is committed as uast/query/skip-deleted-record.p.
- TEMP-TABLE:UNDO. Conversion is implemented (at least in 2272b/10988). Also documentation is scarce, I am testing this attribute at this moment. I will update this note.

#4 - 10/22/2015 03:17 PM - Ovidiu Maxiniuc

I continued investigations on SKIP-DELETED-RECORD but without success. I checked the GUI sources and there are only two occurrences. I will try to isolate a testcase starting from those files.
I understand that this should be similar to SET DELETED statement from dbase/foxpro that works with DELETE/RECALL. This attribute would be meaningful if a RECALL statement would be available in Progress, but AFAIK, there is none. The only way to 'recall' deleted records is to rollback a transaction. I will attempt tomorrow to duplicate/isolate a such construct on windev01 a query with a transaction, too.

The UNDO attribute seems to be wrongly placed. In fact, I believe that we should better distinct between methods and statements of a Progress TEMP-TABLE handle. They are put together in OE docs. However, there are two kind of attributes and methods: for *design* of the dynamic

temp-table, that should unwrap to the builder class, and *runtime* of a temp-table. If for the first kind unwrapping to TempTable looks fine, the second kind should be unwrapped to Temporary or, probably better, to TemporaryBuffer.
For example, converting:

```
h-tt56 = buffer tt56:handle.
MESSAGE h-tt56:has-records.
```

we get something like:

```
Tt561_1.Buf tt56 = TemporaryBuffer.define(Tt561_1.Buf.class, "tt56", "tt56", false);
[...]
hTt56.assign(tt56);
message(new Object[] { hTt56.unwrapTempTable().hasRecords() });
```

It's visible that hTt56 will wrap a Temporary object, but when it is checked for records (or UNDO status), the handle tried to unwrap a TempTable and the call fails.
OTOH, some link must be maintained between the two types of resources.

#5 - 10/23/2015 08:28 AM - Ovidiu Maxiniuc

Further investigations I realized in the second half of my previous note I was wrong. Please ignore that part.

#6 - 10/23/2015 04:01 PM - Ovidiu Maxiniuc

The implementation of UNDO attribute is done for dynamic temp-tables.
However, I discovered that the same attribute of static temp-tables are also alterable. I committed some testcases that demonstrate this (uast/undo/changing-undo-for-temp-tables.p, uast/undo/undo-attribute-dynamic-temp-table.p, uast/undo/undo-attribute-static-temp-table.p).
There is a little more work here, because the NO-UNDO flag is converted statically into the buffer definition. Most likely, nobody will alter the attribute, but I would like to do the full implementation of this attribute.

I am a little concerned about some exceptions that are printed to console, when the external procedure is finished. It looks like we already encountered something similar:

```
[10/23/2015 22:54:04 EEST] (com.goldencode.p2j.persist.Persistence$Context:WARNING) [00000002:0000000C:bogus--
>local/_temp/primary] rolling back orphaned transaction
[10/23/2015 22:54:04 EEST] (com.goldencode.p2j.persist.RecordBuffer:WARNING) Error rolling back transaction
com.goldencode.p2j.persist.PersistenceException: [00000002:0000000C:bogus-->local/_temp/primary] error persist
ing object
        at com.goldencode.p2j.persist.Persistence.handleException(Persistence.java:4075)
        at com.goldencode.p2j.persist.Persistence.save(Persistence.java:2768)
        at com.goldencode.p2j.persist.RecordBuffer.flush(RecordBuffer.java:5468)
        at com.goldencode.p2j.persist.RecordBuffer.setCurrentRecord(RecordBuffer.java:9421)
        at com.goldencode.p2j.persist.RecordBuffer.finishedImpl(RecordBuffer.java:8746)
        at com.goldencode.p2j.persist.RecordBuffer.finished(RecordBuffer.java:5176)
        at com.goldencode.p2j.util.TransactionManager.processFinalizables(TransactionManager.java:4990)
        at com.goldencode.p2j.util.TransactionManager.popScope(TransactionManager.java:2295)
        at com.goldencode.p2j.main.StandardServer.invoke(StandardServer.java:1394)
        at com.goldencode.p2j.main.StandardServer.standardEntry(StandardServer.java:427)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
```

```
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:497)
        at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
        at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:705)
        at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
        at com.goldencode.p2j.net.Conversation.run(Conversation.java:163)
        at java.lang.Thread.run(Thread.java:745)
Caused by: org.hibernate.NonUniqueObjectException: a different object with the same identifier value was alrea
dy associated with the session: [com.goldencode.p2j.persist.dynamic._temp.impl.DynamicRecord1Impl#1]
        at org.hibernate.event.internal.AbstractSaveEventListener.performSave(AbstractSaveEventListener.java:1
80)
        at org.hibernate.event.internal.AbstractSaveEventListener.saveWithGeneratedId(AbstractSaveEventListene
r.java:136)
        at org.hibernate.event.internal.DefaultSaveOrUpdateEventListener.saveWithGeneratedOrRequestedId(Defaul
tSaveOrUpdateEventListener.java:204)
        at org.hibernate.event.internal.DefaultSaveEventListener.saveWithGeneratedOrRequestedId(DefaultSaveEve
ntListener.java:55)
        at org.hibernate.event.internal.DefaultSaveOrUpdateEventListener.entityIsTransient(DefaultSaveOrUpdate
EventListener.java:189)
        at org.hibernate.event.internal.DefaultSaveEventListener.performSaveOrUpdate(DefaultSaveEventListener.
java:49)
        at org.hibernate.event.internal.DefaultSaveOrUpdateEventListener.onSaveOrUpdate(DefaultSaveOrUpdateEve
ntListener.java:90)
        at org.hibernate.internal.SessionImpl.fireSave(SessionImpl.java:756)
        at org.hibernate.internal.SessionImpl.save(SessionImpl.java:748)
        at org.hibernate.internal.SessionImpl.save(SessionImpl.java:744)
        at com.goldencode.p2j.persist.Persistence.save(Persistence.java:2753)
        at com.goldencode.p2j.persist.RecordBuffer.flush(RecordBuffer.java:5468)
        at com.goldencode.p2j.persist.RecordBuffer.setCurrentRecord(RecordBuffer.java:9421)
        at com.goldencode.p2j.persist.RecordBuffer.finishedImpl(RecordBuffer.java:8746)
        at com.goldencode.p2j.persist.RecordBuffer.finished(RecordBuffer.java:5176)
        at com.goldencode.p2j.util.TransactionManager.processFinalizables(TransactionManager.java:4990)
        at com.goldencode.p2j.util.TransactionManager.popScope(TransactionManager.java:2295)
        at com.goldencode.p2j.main.StandardServer.invoke(StandardServer.java:1394)
        at com.goldencode.p2j.main.StandardServer.standardEntry(StandardServer.java:427)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:497)
        at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:76)
        at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:705)
        at com.goldencode.p2j.net.Conversation.block(Conversation.java:319)
        at com.goldencode.p2j.net.Conversation.run(Conversation.java:163)
        at java.lang.Thread.run(Thread.java:745)
```

#7 - 10/28/2015 03:41 PM - Ovidiu Maxiniuc

Task branch 2618a created from main trunk revno 10950. It contains the implementation of UNDO attribute and fixes for exception above. Committed revision 10951.

#8 - 10/29/2015 12:06 AM - Eric Faulhaber

Code review 2618a/10951:

This update looks good. Please clean up a few minor issues:

- either add a header entry to BufferManager or back out the whitespace-only change;
- typo in the comment at DynamicTablesHelper:1472;
- typo at RecordBuffer:907.

You can test this branch any time you're ready, but we will not merge to trunk until after 2677a, so please plan accordingly.

Nice work tracking down and fixing the NonUniqueObjectException.

#9 - 10/29/2015 11:19 AM - Ovidiu Maxiniuc

Guy,

From the initial list, the SKIP-DELETED-RECORD is the only attribute that lacks support in p2j. I was not able to understand how the mechanism work. As I wrote in a previous note, the documentation is very limited and I was not able to create a testcase on windev01 that demonstrates the influence of this attribute. In GUI sources, there are only two usages (adcdedi1.w and adcdedi2.w). In both occurrences, the value is set to true, which I realized it's the default value (from my investigations), so the statements are in fact no-ops.
Could you provide a simple example of usage of this attribute where its effect is visible, please?

#10 - 10/29/2015 12:02 PM - Guy M

Hi Ovidiu,

To be honest I didn't know about the existence of that attribute, nor what it did.

Looking at the code, a temp-table is built up (the available records), and then the selected records are removed from the available records temp-table and added into a selected records temp-table. Perhaps whoever originally wrote that code was worried that by removing records out of the original query, it would result in it not functioning correctly. I guess the code was tried with the attribute set, and it worked, so it was left in.

I think that rather than both of us wasting time investigating SKIP-DELETED-RECORD, we just amend the code to remove these statements. If you ever get another customer who actually uses SKIP-DELETED-RECORD (= FALSE), then you can investigate it then.

Hope that's OK.

Regards,
Guy

#11 - 10/29/2015 12:15 PM - Guy M

I've updated the bzr repo on lincon01 with versions of adcdedi1.w and adcdedi2.w with the references to SKIP-DELETED-RECORD removed.

#12 - 10/29/2015 04:28 PM - Eric Faulhaber

Thanks, Guy.

#13 - 10/29/2015 04:34 PM - Eric Faulhaber

Code review 2618a/10952:

Ovidiu, all looks good. To double-up on regression testing, I'm going to be making updates to 2618a for my work in #2692. Once that's ready, I'll get back with you about regression testing this branch.

#14 - 10/29/2015 04:36 PM - Eric Faulhaber

- % Done changed from 0 to 80

- Status changed from WIP to Test

#15 - 11/10/2015 04:49 PM - Eric Faulhaber

Code review 2618a/10953:

Changes look good. Due to the regression found in 10952, I separated my work for #2692 into a separate branch (2692b). So, 2618a now will contain only your changes. Please regression test. I'll try another run of unit tests with 2618a tonight and let you know the results.

#16 - 11/10/2015 04:54 PM - Eric Faulhaber

On second thought: Greg, would you prefer we hold off regression testing 2618a and 2692b until you've merged 2677a to trunk, then rebase those branches and test that? I don't think there should be much/any overlap with 2677a (which is why I suggested we test 2618a), but I understand it would be more painful to rebase 2677a than the other two branches.

#17 - 11/11/2015 08:29 AM - Greg Shah

There is no need to wait on testing. Get it done now and then merge into **2677a**. Do not merge to trunk.

#18 - 11/11/2015 11:19 AM - Eric Faulhaber

Common search and full search unit tests went OK with 2618a/10953. Running sysmaint and developer tests next...

#19 - 11/11/2015 12:00 PM - Eric Faulhaber

Sysmaint tests were OK.

#20 - 11/11/2015 02:17 PM - Eric Faulhaber

Developer tests were OK.

If normal regression testing goes well, please merge the changes to 2677a.

#21 - 12/01/2015 08:17 AM - Greg Shah

Can we close this?

#22 - 12/02/2015 08:35 AM - Ovidiu Maxiniuc

The update was merged into task branch 2677a as revision 11046. 2677a is already in trunk.
I believe we can close this issue.

#23 - 12/02/2015 09:03 AM - Greg Shah

- % Done changed from 80 to 100

- Status changed from Test to Closed

#24 - 11/16/2016 12:12 PM - Greg Shah

- Target version changed from Milestone 12 to GUI Support for a Complex ADM2 App