

Database - Feature #6479

pending DISCONNECT and static buffers

05/31/2022 03:12 PM - Greg Shah

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

History

#1 - 05/31/2022 03:13 PM - Greg Shah

From #6277-162, by Constantin:

But, the weird thing happens: for a .cls, if the database is connected when the legacy class was loaded (as this is done only once), accessing a property getter which is using buffers in this disconnected database will no longer raise a STOP condition (this is what FWD does, could not duplicate), and will actually lookup the record, even if NUM-DBS and LDBNAME reports no connected database!

FWD currently does this, while in the static property getter:

```
StopConditionException.<init>(String) line: 98
ErrorManager.postprocessOnAbend(String[]) line: 3445
ErrorManager.showErrorAndAbend(int[], String[]) line: 1362
RecordBuffer.getDatabase() line: 9605
RecordBuffer.initialize() line: 6876
```

It doesn't make any sense, somehow OE keeps the connection open in the static class state... and a DISCONNECT does not affect it.

An example is this:

```
class oo.Foo:
  def static buffer bbook for book.
  def public static property p1 as int get:
  do on error undo, leave:
    find first bbook no-lock where bbook.book-title = m1().
  end.
  message avail(bbook) num-dbs ldbname(1).
  return 12345.
end.

method public static char m1():
  return string(now).
end.
end.
```

if you do this:

```
connect c:\p2j_Test\p2j_test.
message oo.foo:p1. // database connected, record not found
disconnect "p2j_test".
message oo.foo:p1. // database disconnected (num-dbs = 0, ldbname(1) = ?), record not found
```

A question: why do we raise a STOP condition in RecordBuffer.initialize if the database is not connected?

#2 - 05/31/2022 03:14 PM - Greg Shah

From #6277-163, by Constantin:

Found the root cause: <https://knowledgebase.progress.com/articles/Article/P141997>

A DISCONNECT must be set to 'pending' if there are buffers defined as static class members, attached to that database... and as classes can't ever discard static buffers, the DISCONNECT will never execute. But NUM-DBS and LDBNAME function will be affected, will no longer report this 'disconnected but actually not' database.

#3 - 05/31/2022 03:23 PM - Constantin Asofiei

- File 6479_patch1.txt added

The only solution is to mark any buffer defined as a static class member or in a static method as 'static', at RecordBuffer.define. There is no way at runtime to find the correct method on the stack (as the signature is missing).

The current patch is attached.

I need help with ConnectionManager, the proper place to cleanup the NUM-DBS and LDBNAME, but keep the database connected behind the scene.

#5 - 06/01/2022 02:05 AM - Constantin Asofiei

The issue is not just about static buffers - we need a reference counter, as the pending disconnect is possible with persistent programs, too.

So, we have a program like this:

```
procedure proc0.  
    find first book no-lock.  
    message avail(book) num-dbs ldbname(1).  
end.
```

and another program like this:

```
connect c:\p2j_Test\p2j_Test.  
def var h as handle.  
run a.p persistent set h.  
// this 'binds' the buffer to the database  
run proc0 in h.  
// disconnect is marked as pending  
disconnect "p2j_test".  
// database still connected  
run proc0 in h.  
// database gets disconnected, as the buffer gets deleted  
delete object h.  
// this RUN fails because the a.p's RCODE-INFO:DATABASE-REFERENCES dbs are no longer connected  
run a.p persistent set h.
```

#6 - 06/01/2022 02:57 AM - Constantin Asofiei

And considering that static buffers are associated with a legacy class instance (which is the equivalent of a persistent program), and also the class instances which are the same as a persistent program, I need to remove the 'static' conversion flag for the buffer, and work only at runtime - use the reference counter to track buffers bound to a database.

Eric/Ovidiu: how can I properly get the persist.Database from a RecordBuffer's 'ldbOrAlias' parameter? The reference counter will be in a map, with Database as key. Or better said, is BufferManager.activeDatabases actually tracking buffers?

#7 - 06/01/2022 03:12 PM - Constantin Asofiei

The latest patch is this:

```
### Eclipse Workspace Patch 1.0
#P p2j6129a
Index: src/com/goldencode/p2j/persist/ConnectionManager.java
=====
--- src/com/goldencode/p2j/persist/ConnectionManager.java      (revision 3650)
+++ src/com/goldencode/p2j/persist/ConnectionManager.java      (working copy)
@@ -2861,13 +2861,18 @@
     */
     private void disconnectImmediately(Database database, boolean sessionCleanup)
     {
-         if (sessionCleanup)
+         String ldbName = ldbMap.get(database);
+         if (ldbName != null)
             {
-             String ldbName = ldbMap.get(database);
-             if (ldbName != null)
+             if (sessionCleanup)
                 {
                     removeConnected(ldbName, sessionCleanup);
                 }
+             else if (BufferManager.get().databaseInUse(database))
                 {
                     // do nothing
                     return;
                 }
             }

         if (!database.isLocal())
@@ -3645,6 +3650,12 @@
             throw new PersistenceException("DISCONNECT: Database " + ldbName + " is not open or unknown", 1009)
;
     }

+     if (BufferManager.get().databaseInUse(getDatabase(ldb)))
+     {
+         // do nothing
+         return;
+     }
+
     // remove the database from the connected map whether or not the disconnect can occur immediately
     Database database = removeConnected(ldb, false);

Index: src/com/goldencode/p2j/persist/BufferManager.java
=====
--- src/com/goldencode/p2j/persist/BufferManager.java          (revision 3650)
+++ src/com/goldencode/p2j/persist/BufferManager.java          (working copy)
@@ -2627,6 +2627,13 @@
     }
     }

+     boolean databaseInUse(Database database)
+     {
+         MutableInteger val = activeDatabases.get(database);
+         return val != null && val.get() > 0;
+     }
}
```

```
+  
  /**  
   * A buffer was initialized. If it was previously stored as an uninitialized, open buffer, move it into  
   * the active, open buffers list. Perform special handling for persistent procedure buffers.
```

I haven't done standalone testcases, and #6277 didn't show that this feature is used there.

In any case, this needs to be fixed.

#8 - 06/07/2022 03:20 PM - Ovidiu Maxiniuc

The idea is good. I only see one problem: the activeDatabases of BufferManager tracks databases which have at least an active RecordBuffer, regardless it is static or not. If only the static buffers must checked then we could add supplementary information in activeDatabases, to know whether the buffers from current set are declared static.

#9 - 06/08/2022 02:25 AM - Constantin Asofiei

My understanding is that a database can't be disconnected if there is at least a buffer attached to it - static or dynamic, doesn't matter. Although some more testing is required.

This is not a priority for #6277, the issue was found when investigating another problem, for which in the end the root cause was something else.

Files

| | | | |
|-----------------|---------|------------|--------------------|
| 6479_patch1.txt | 6.58 KB | 05/31/2022 | Constantin Asofiei |
|-----------------|---------|------------|--------------------|