## Database - Feature #2258

## retrofit BUFFER-COPY and BUFFER-COMPARE language statement runtime support to use legacy field names

03/13/2014 01:41 PM - Eric Faulhaber

| | | | |
|---|---|---|---|
| **Status:** | Closed | **Start date:** | |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Vadim Nebogatov | **% Done:** | 100% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | Cleanup and Stablization for Server Features | | |
| **billable:** | No | **vendor_id:** | GCD |

**Description**

### History

**#1 - 03/13/2014 02:00 PM - Eric Faulhaber**

*- Target version set to Milestone 11*

When runtime support for the handle-based methods BUFFER-COPY() and BUFFER-COMPARE() originally was implemented, we did not have legacy field name information available at runtime. So, the initial implementation of these features was based on DMO property names (the converted forms of the legacy field names). This was fine for the internals (and in fact, the core implementation still uses the DMO property names to do its work), but wherever inputs or outputs are exposed to converted code, we have to use the legacy field names.

We since have added legacy field name information using Java annotations, exposed through the TableMapper class. As part of Milestone 5, I rewrote the runtime support for the converted, BUFFER-COPY() and BUFFER-COMPARE(), handle-based methods to integrate this information.

We now have to do a similar retrofit for the language statement versions of these operations. Again, the internal implementation should change as little as possible, still using DMO properties to do the actual work, but wherever field names are exposed in the API as inputs (e.g., include/exclude lists) or outputs (e.g., lists of fields which did not match for BUFFER-COMPARE), we have to use the legacy names.

Please use the recent changes to RecordBuffer, TableMapper, and the new FieldInfo class as a guide and re-use as much as possible. I'm regression testing those changes now, and I will post the bzr revision under which they are committed, once they have passed testing.

**#2 - 03/17/2014 04:37 PM - Eric Faulhaber**

*- Assignee set to Stanislav Lomany*

I have checked in my pending changes under bzr rev. 10493.

**#3 - 03/29/2014 07:38 AM - Eric Faulhaber**

*- Assignee changed from Stanislav Lomany to Vadim Nebogatov*

**#4 - 04/05/2014 06:42 AM - Vadim Nebogatov**

During testing I have found one bug not related directly with issue.

RecordBuffer, line 4032:

```
if (!propsMap.containsKey(prop1))
```

should be

```
if (!propsMap.containsKey(new DatumAccess(prop1, null)))
```

As result buffer-compare gives wrong result both for method and statement variants.

Test case

```
def var cresult as char.
def var lresult as logical.

def temp-table tt1
  field f1 as int
  field f2 as int
  field f3 as int.
def temp-table tt2
  field f1 as int
  field f2 as int
  field f3 as int.

def buffer btt1 for tt1.
def buffer btt2 for tt2.

def var h1 as handle.
def var h2 as handle.

h1 = buffer tt1:handle.
h2 = buffer tt2:handle.

create btt1.
assign btt1.f1 = 1
       btt1.f2 = 2
       btt1.f3 = 3.

create btt2.
assign btt2.f1 = 1
       btt2.f2 = 42
       btt2.f3 = 16.

buffer-compare btt1 except f3 to btt2  save cresult.
message "Result 1:" string(cresult).

h1:FIND-FIRST().
h2:FIND-FIRST().

lresult = h1:BUFFER-COMPARE(h2, "casesensitive", "f3").
message "Result 2:" string(lresult).
```

Result:

```
Result 1: 
Result 2: yes
```

expected:

```
Result 1: f2
Result 2: no
```

Should I fix it in separate update or along with this issue?

**#5 - 04/05/2014 08:33 AM - Eric Faulhaber**

Please include the fix with your update for this issue.

**#6 - 04/07/2014 02:15 PM - Vadim Nebogatov**

One more not related bug found with compare with BINARY both for statement (method works properly):

```
def var c-result as char.
def var l-result as logical.

def temp-table tt1
  field f1 as character
  field f2 as character
  field f3 as character
  field f4 as character.
def temp-table tt2
  field f1 as character
  field f2 as character
  field f3 as character.

def buffer btt1 for tt1.
def buffer btt2 for tt2.

def var h1 as handle.
def var h2 as handle.

h1 = buffer tt1:handle.
h2 = buffer tt2:handle.

create btt1.
assign btt1.f1 = "aa"
       btt1.f2 = "bb"
       btt1.f3 = "cc"
       btt1.f4 = "dd".

create btt2.
assign btt2.f1 = "aA"
       btt2.f2 = "bb"
       btt2.f3 = "cc".

h1:FIND-FIRST().
h2:FIND-FIRST().

buffer-compare btt1 except f3 f4 to btt2 binary save c-result.
message "Result binary statement:" string(c-result).
```

Results:

```
Result binary statement:
```

Expected:

```
Result binary statement: f1
```

The reason is that conversion generates wrong method in BINARY case:

```
            RecordBuffer.compare(btt1, new String[]
            {
               "f3",
               "f4"
            }, true, btt2, true, cResult);
            message(new Object[]
            {
               "Result case-sensitive statement:",
               valueOf(cResult)
            });
            RecordBuffer.compare(btt1, new String[]
            {
               "f3",
               "f4"
            }, true, btt2, cResult);
            message(new Object[]
            {
               "Result binary statement:",
               valueOf(cResult)
            });
```

Should be the same as in CASE-SENSITIVE case.

**#7 - 04/08/2014 06:27 AM - Vadim Nebogatov**

Fixed with replacement in database_access.rules:

```
      <!-- handle case-sensitive option within buffer compare statement -->
      <rule>type == prog.kw_case_sen and parent.type == prog.kw_buf_comp
         <action>createJavaAst(java.bool_true, "", closestPeerId)</action>
      </rule>
```

replaced with

```
      <!-- handle case-sensitive and binary options within buffer compare statement -->
      <rule>(type == prog.kw_case_sen or
             type == prog.kw_binary) and
            parent.type == prog.kw_buf_comp
         <action>createJavaAst(java.bool_true, "", closestPeerId)</action>
      </rule>
```

**#8 - 04/09/2014 05:48 PM - Vadim Nebogatov**

*- File vmn_upd20140409a.zip added*

*- File vmn_upd20140409b.zip added*

I have attached update vmn_upd20140409a.zip and tests used vmn_upd20140409b.zip for review.

**#9 - 04/10/2014 12:46 AM - Eric Faulhaber**

Code review 20140409a:

The update looks good. Please regression test.

**#10 - 04/10/2014 12:50 AM - Eric Faulhaber**

*- % Done changed from 0 to 80*

*- Status changed from New to Test*

**#11 - 04/10/2014 03:26 PM - Vadim Nebogatov**

Regression 20140410_105536 completed. 3 tests failed:

1. tc_job_002 FAILED failure in step 40: 'Unexpected EOF in actual at page # 1.'
2. tc_dc_slot_029 FAILED failure in step 8: 'timeout before the specific screen buffer became available (Mismatched data at line 4, column 9. Expected '' (0x0000 at relative Y 4, relative X 9) and found '1' (0x0031 at relative Y 4, relative X 9).)'
3. tc_job_clock_002 FAILED failure in step 20: 'timeout before the specific screen buffer became available (Mismatched data at line 5, column 18. Expected '' (0x0000 at relative Y 5, relative X 18) and found ' ┌' (0x250C at relative Y 5, relative X 18).)'

Will start regression one more time.

**#12 - 04/11/2014 02:43 PM - Vadim Nebogatov**

Regression 20140411_102011 completed. 3 tests failed:

1. tc_job_002 FAILED failure in step 40: 'Unexpected EOF in actual at page # 1.'
2. gso_441 FAILED failure in step 67: 'timeout before the specific screen buffer became available (Mismatched data at line 0, column 1. Expected 'M' (0x004D at relative Y 0, relative X 1) and found '' (0x0000 at relative Y 0, relative X 1).)'
3. tc_job_clock_002 FAILED failure in step 20: 'timeout before the specific screen buffer became available (Mismatched data at line 5, column 18. Expected '' (0x0000 at relative Y 5, relative X 18) and found ' ┌' (0x250C at relative Y 5, relative X 18).)'

tc_job_clock_002 failed again, Will start regression one more time.

**#13 - 04/13/2014 01:47 PM - Vadim Nebogatov**

Regression 20140413_095429 completed. 6 tests failed, including

tc_job_002
tc_job_clock_002

Will start regression one more time.

**#14 - 04/13/2014 02:49 PM - Eric Faulhaber**

Since tc_job_clock_002 has failed each time, have you checked the server.log to see if there is a real regression? Have you tried running the test manually?

**#15 - 04/13/2014 05:22 PM - Vadim Nebogatov**

tc_job_clock_002 test fails each time because of timeout error:

failure in step 20: 'timeout before the specific screen buffer became available (Mismatched data at line 5, column 18. Expected '' (0x0000 at relative Y 5, relative X 18) and found ' ┌' (0x250C at relative Y 5, relative X 18).)'

Is it possible to execute regression of this test separately?

Server.log contains

```
java.lang.RuntimeException: com.goldencode.p2j.cfg.ConfigurationException: Error loading p2j.cfg.xml configura
tion
...
Caused by: com.goldencode.p2j.cfg.ConfigurationException: Cannot find resource cfg/p2j.cfg.xml
        at com.goldencode.p2j.cfg.Configuration.getInstance(Configuration.java:492)
        at com.goldencode.p2j.cfg.Configuration.getSchemaConfig(Configuration.java:228)
        at com.goldencode.p2j.persist.ConversionPool.initialize(ConversionPool.java:80)
        at com.goldencode.p2j.main.StandardServer.bootstrap(StandardServer.java:723)
        at com.goldencode.p2j.main.ServerDriver.start(ServerDriver.java:428)
        at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:423)
        at com.goldencode.p2j.main.ServerDriver.process(ServerDriver.java:155)
        at com.goldencode.p2j.main.ServerDriver.main(ServerDriver.java:772)
```

but all other tests excepting 6 are passed.

Also I noticed the following error on console, not sure if it happened previous times:

```
 Non-testing errors (CONNECT_FAILURE) occured during test execution:
     [exec] com.jcraft.jsch.JSchException: channel is not opened.
     [exec]         at com.jcraft.jsch.Channel.sendChannelOpen(Channel.java:670)
     [exec]         at com.jcraft.jsch.Channel.connect(Channel.java:151)
     [exec]         at com.goldencode.harness.transport.SSH2Transport.connect(SSH2Transport.java:79)
     [exec]         at com.goldencode.harness.transport.TransportManager.connect(TransportManager.java:69)
     [exec]         at com.goldencode.harness.Driver.<init>(Driver.java:83)
     [exec]         at com.goldencode.harness.TestSet.run(TestSet.java:257)
     [exec]         at java.lang.Thread.run(Thread.java:744)
     [exec]
     [exec] Running test plan... ** CTRL-C part has failed!
     [exec] ** Running main part...
     [exec]
     [exec] Running test plan... ESC M : R is 8, tm is 8, bm is 19
     [exec] ESC M : R is 8, tm is 8, bm is 19
     [exec] ESC M : R is 0, tm is 0, bm is 3
     ...
```

### #16 - 04/16/2014 03:26 AM - Vadim Nebogatov

Regression 20140415_154604 completed. 3 tests failed:

1. tc_job_002 FAILED failure in step 40: 'Unexpected EOF in actual at page # 1.'
2. gso_17 FAILED failure in step 79: 'Mismatched data at absolute row 13, relative row -1, page # -1, page size -1, last page false, column index 3. Expected '3' (0x33) and found '1' (0x31).'
3. gso_430 FAILED failure in step 9: 'timeout before the specific screen buffer became available (Mismatched data at line 18, column 0. Expected '┌' (0x250C at relative Y 18, relative X 0) and found '' (0x0000 at relative Y 18, relative X 0).)'

So, regression seems passed.

### #17 - 04/18/2014 06:15 PM - Vadim Nebogatov

Update vmn_upd20140409a.zip passed regression and distributed. Bzr revision is 10511.

### #18 - 04/19/2014 02:32 PM - Eric Faulhaber

*- % Done changed from 80 to 100*

*- Status changed from Test to Closed*

### #19 - 11/16/2016 12:07 PM - Greg Shah

*- Target version changed from Milestone 11 to Cleanup and Stablization for Server Features*

## Files

| | | | |
|---|---|---|---|
| vmn_upd20140409a.zip | 118 KB | 04/09/2014 | Vadim Nebogatov |
| vmn_upd20140409b.zip | 2.07 KB | 04/09/2014 | Vadim Nebogatov |