

## Database - Feature #2044

### implement support for a DELETE statement with a validation expression

02/24/2013 03:17 PM - Greg Shah

<b>Status:</b>	Closed	<b>Start date:</b>	02/24/2013
<b>Priority:</b>	Normal	<b>Due date:</b>	07/02/2013
<b>Assignee:</b>		<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Runtime Support for Server Features	<b>vendor_id:</b>	GCD
<b>billable:</b>	No		
<b>Description</b>			
<b>Subtasks:</b>			
Feature # 2045: conversion support for a DELETE statement with a validation expression			<b>Closed</b>
Feature # 2046: runtime support for DELETE statement with a validation expression			<b>Closed</b>

#### History

##### #1 - 02/24/2013 03:31 PM - Greg Shah

- File *delete-with-validation.p* added

- File *ges\_upd20130224a.zip* added

Add support to the conversion and runtime for a DELETE statement with a validation expression. We already support DELETE but need to emit a validation inner class (with proper promotion of all var refs etc...) and emit a new instance of that validator into the `buffer.delete()` method call. For now the runtime will be stubbed to accept the validation expression and conditionally delete, but later one we will need to test the behavior here and confirm that it is correct.

See the attached testcase for the syntax that must be supported. Attached is also the set of modifications made to add the conversion support, the runtime stubs and an untested implementation (which may be completely wrong).

##### #2 - 02/24/2013 04:04 PM - Eric Faulhaber

Code review 20130224a:

Changes look OK to me. Note that `database_access.rules` needs a header entry for your changes.

##### #3 - 02/24/2013 04:08 PM - Greg Shah

I fixed the header and have re-attached the update. This is going into conversion regression testing now.

##### #4 - 02/24/2013 04:08 PM - Greg Shah

- File *ges\_upd20130224b.zip* added

And now the update...

##### #5 - 02/24/2013 05:08 PM - Greg Shah

It passed conversion testing and is committed to bzz as revision 10198.

##### #6 - 07/25/2013 05:27 PM - Vadim Nebogatov

I have tested both in P2J and 4GL and also with NO-ERROR and some other examples like

```
create person.  
person.last-name = "Bogus1".  
person.ssn = "11".  
delete person validate(last-name eq "BOGUS", "1 We only delete when the last name is bogus!") no-error.  
delete person validate(last-name eq "BOGUS", "2 We only delete when the last name is bogus!").
```

It seems it works correctly now excepting some minor differences like

in 4GL:

```
2 We only delete when the last name is bogus!
```

in P2J

```
** 2 We only delete when the last name is bogus! (-1)
```

What else should be implemented, checked or tested in this issue?

#### **#7 - 07/25/2013 05:57 PM - Eric Faulhaber**

Interesting. I have not previously seen a Progress-style error which does not have the leading asterisks and trailing error number.

Please propose a change which adjusts for these minor differences with minimal or (if possible) no change to the converted code. Other than that, if the core logic is working, there is nothing more to do with this issue.

#### **#8 - 07/25/2013 06:17 PM - Vadim Nebogatov**

- File 56.png added

- File 23.png added

I have attached screenshots

**#9 - 07/25/2013 08:22 PM - Eric Faulhaber**

Oh, I believed you, I just found it interesting how inconsistent Progress is.

**#10 - 07/26/2013 06:17 AM - Vadim Nebogatov**

Eric Faulhaber wrote:

Please propose a change which adjusts for these minor differences with minimal or (if possible) no change to the converted code. Other than that, if the core logic is working, there is nothing more to do with this issue.

It seems I did not see in 4GL messages enumerated with (-1). I suggest

1) use

```
ErrorMessage.recordOrThrowError(-1, validateMessage().toStringMessage(), false);
```

in BufferValidator.validate() in order to suppress prefix in this case

2) correct ErrorMessage.buildErrorText() :

```
public static String buildErrorText(int num, String text, boolean prefix)
{
    if (num < 0)
    {
        if (prefix)
        {
            text = "*** " + text;
        }
        return text;
    }
    else
    {
        String spec = (prefix ? "*** %s. (%d)" : "%s. (%d)");
        return String.format(spec, text, num);
    }
}
```

## #11 - 07/26/2013 07:21 AM - Greg Shah

2 things:

1. Are you sure that failing a validation in the 4GL is considered an "ERROR" in the 4GL sense? What do the ERROR-STATUS attributes/methods return in the NO-ERROR case? Does an ERROR condition get raised in the non-NO-ERROR case? I suspect the 4GL doesn't throw an ERROR and does not update the ERROR-STATUS handle, because it is just drawing message text. This is the same way we do it when a UI validation fails. There is no ERROR condition raised in that case.

2. Don't change buildErrorText() before we really know if this is ERROR behavior. In addition, we often use a negative number in an ErrorManager call when we can't duplicate the 4GL error but we know there should be one there. The proposed result would not be correct for other callers of buildErrorText(), so the proposed change would not be accepted.

## #12 - 07/26/2013 10:52 AM - Vadim Nebogatov

### Non-Error case

```
create person.  
person.last-name = "Bogus1".  
person.ssn = "11".
```

```
message "ERROR-STATUS:ERROR" ERROR-STATUS:ERROR.  
message "ERROR-STATUS:NUM-MESSAGES" ERROR-STATUS:NUM-MESSAGES.  
message "ERROR-STATUS:GET-NUMBER[1]" ERROR-STATUS:GET-NUMBER[1].
```

```
delete person validate(last-name eq "BOGUS", "1 We only delete when the last name is  
bogus!") no-error.
```

```
message "ERROR-STATUS:ERROR" ERROR-STATUS:ERROR.  
message "ERROR-STATUS:NUM-MESSAGES" ERROR-STATUS:NUM-MESSAGES.  
message "ERROR-STATUS:GET-MESSAGE[1]" ERROR-STATUS:GET-MESSAGE[1].  
message "ERROR-STATUS:GET-NUMBER[1]" ERROR-STATUS:GET-NUMBER[1].
```

### Result (error status is changed):

```
ERROR-STATUS:ERROR no  
ERROR-STATUS:NUM-MESSAGES 0  
ERROR-STATUS:GET-NUMBER[1] 0  
ERROR-STATUS:ERROR yes  
ERROR-STATUS:NUM-MESSAGES 1  
ERROR-STATUS:GET-MESSAGE[1] 1 We only delete when the last name isbogus!  
ERROR-STATUS:GET-NUMBER[1] 0
```

### Error case

```
create person.  
person.last-name = "Bogus1".  
person.ssn = "11".
```

```
message "ERROR-STATUS:ERROR" ERROR-STATUS:ERROR.  
message "ERROR-STATUS:NUM-MESSAGES" ERROR-STATUS:NUM-MESSAGES.  
message "ERROR-STATUS:GET-NUMBER[1]" ERROR-STATUS:GET-NUMBER[1].
```

```
delete person validate(last-name eq "BOGUS", "1 We only delete when the last name is  
bogus!").
```

```
message "ERROR-STATUS:ERROR" ERROR-STATUS:ERROR.
```

Result (execution is stopped after first validation error):

```
ERROR-STATUS:ERROR no  
ERROR-STATUS:NUM-MESSAGES 0  
ERROR-STATUS:GET-NUMBER[1] 0  
1 We only delete when the last name isbogus!
```

**#13 - 07/26/2013 03:41 PM - Vadim Nebogatov**

What about adding one more parameter to buildErrorText and recordOrThrowError:

boolean withNumber

and do not print number in message if withNumber is false? But use number, including -1, for da.addRecord(new ErrorEntry(num, errmsg)).

Or you prefer to leave message for BufferValidator.validate() like now, with number?

**#14 - 07/29/2013 09:33 AM - Greg Shah**

Before I can answer, I have one more question. In both cases (NO-ERROR and without NO-ERROR), what is returned from error-status:get-number(1)?

**#15 - 07/29/2013 11:48 AM - Vadim Nebogatov**

Greg Shah wrote:

Before I can answer, I have one more question. In both cases (NO-ERROR and without NO-ERROR), what is returned from error-status:get-number(1)?

Everywhere ERROR-STATUS:GET-NUMBER[1] 0. I have corrected examples and outputs to note 12.

**#16 - 07/29/2013 11:58 AM - Greg Shah**

I don't see the value in making changes to builtErrorText(). If I understand correctly, we just need to bypass the call to builtErrorText() in this case.

It seems to me that the solution should be this:

1. Pass an error number of 0 instead of -1.
2. Add a boolean asMsg parameter to recordOrThrowError(). As usual, just add this to the main "worker" version and pass false from the current methods so that we can minimize the change throughout P2J.
3. If asMsg is true, bypass the call to builtErrorText() and just use the given text as the message text.

If I am correct, then you can duplicate the exact behavior with that approach.

**#17 - 07/29/2013 03:36 PM - Vadim Nebogatov**

- File `vmn_upd20130729b.zip` added

- File `vmn_upd20130729a.zip` added

I have attached update `vmn_upd20130729a.zip/vmn_upd20130729b.zip`

- 1) Added a boolean `asMsg` parameter to `recordOrThrowError()`.
- 2) based on 4GL behavior: show `validateMessage` if error is happened on `validateExpression` calculation.

Example (person table is empty):

```
delete person validate(last-name eq "BOGUS", "We only delete when the last name is bogus!").
```

4GL:

```
** No Person record is available. (91)
We only delete when the last name is bogus!
```

Tested also situations when `validateMessage` also fails on calculation (for example, also database-driven) .

Example (person table is empty):

```
delete person validate(last-name eq "BOGUS", last-name + "BOGUS").
```

4GL:

```
** No Person record is available. (91)
** No Person record is available. (91)
```

**#18 - 07/29/2013 04:21 PM - Vadim Nebogatov**

Do you plan to support Alternate Buffer Pool (since 10.2B)?

**#19 - 07/30/2013 02:47 PM - Vadim Nebogatov**

I have tested DELETE with validation when record is locked. In this case 4GL simply ignores this record.

One user executes

```
delete from person.  
create person.  
person.last-name = "Bogus".  
person.ssn = "1".  
FIND FIRST person EXCLUSIVE-LOCK NO-WAIT NO-ERROR.  
message "find".  
FIND FIRST person EXCLUSIVE-LOCK NO-WAIT NO-ERROR.  
message "find".  
FIND FIRST person EXCLUSIVE-LOCK NO-WAIT NO-ERROR.  
message "find".
```

and stops when message "find" appears.

Second user executes

```
FIND FIRST person EXCLUSIVE-LOCK NO-WAIT NO-ERROR.  
delete person validate(last-name eq "BOGUS", "We only delete when the last name is  
bogus!").  
message "deleted".
```

and receives result

```
** No Person record is available. (91)  
We only delete when the last name is bogus!
```

**#20 - 07/30/2013 02:58 PM - Greg Shah**

Be careful with your conclusions here. This result can be heavily affected by the 4GL caching/flushing behavior AND by the "dirty database" support. Eric will have to respond in some detail to give thoughts about your results.

**#21 - 07/31/2013 07:27 AM - Eric Faulhaber**

Vadim Nebogatov wrote:

Do you plan to support Alternate Buffer Pool (since 10.2B)?

Not to my knowledge. There are no BUFFER-POOL entries in the .df file of the current project.

How would this affect your implementation of this feature?

**#22 - 07/31/2013 08:21 AM - Eric Faulhaber**

Code review 20130729a:

Code looks fine. However, I would suggest, instead of changing handle and ControlFlowOps, leave them as is, and add back an implementation of `ErrorManager.recordOrThrowError(int[] nums, String[] texts, boolean prefix)`, which simply calls `recordOrThrowError(nums, texts, prefix, false)`. Also add an `@throws EndConditionException` javadoc entry to `BufferValidator.validate()` to describe the special case in which `EndConditionException` is thrown.

After these changes, please regression test the update and check it in if it passes.

**#23 - 07/31/2013 08:38 AM - Eric Faulhaber**

Vadim Nebogatov wrote:

I have tested DELETE with validation when record is locked. In this case 4GL simply ignores this record.

I'm not sure what you mean by "ignores this record". The second user could not obtain the record, because it was requested with a lock with the NO-WAIT, NO-ERROR options. Since it was locked by the first user, the second user's person buffer is simply empty when the delete statement is called (hence the 91 error). Please explain what this test is meant to determine.

**#24 - 07/31/2013 12:32 PM - Vadim Nebogatov**

Eric Faulhaber wrote:

Vadim Nebogatov wrote:

I have tested DELETE with validation when record is locked. In this case 4GL simply ignores this record.

I'm not sure what you mean by "ignores this record". The second user could not obtain the record, because it was requested with a lock with the



NO-WAIT, NO-ERROR options. Since it was locked by the first user, the second user's person buffer is simply empty when the delete statement is called (hence the 91 error). Please explain what this test is meant to determine.

You are correct. Sometimes I forget that delete (and with validation too) works with concrete record in contrast to delete from. Please ignore my phrase. It seems my latest test is senseless.

**#25 - 07/31/2013 01:36 PM - Vadim Nebogatov**

- File *vmn\_upd20130731a.zip* added

Update *vmn\_upd20130731a.zip* replaces update *vmn\_upd20130729a.zip* with fixed comments in note 22. Updated to revision 10369. Regression started.

**#26 - 08/01/2013 03:54 PM - Vadim Nebogatov**

Regression 20130801\_101909 completed. There are 6 FAILED tests in *ts\_tests*.

- 1. *tc\_job\_002* FAILED failure in step 40: 'Unexpected EOF in actual at page # 1.'
- 2. *tc\_dc\_slot\_024* 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 'r' (0x250C at relative Y 5, relative X 18).)'
- 4. *tc\_job\_clock\_004* FAILED failure in step 3: 'Timeout while waiting for event semaphore to be posted.'
- 5. *tc\_job\_matlcron\_001* FAILED failure in step 40: 'timeout before the specific screen buffer became available (Mismatched data at line 21, column 0. Expected 'R' (0x0052 at relative Y 21, relative X 0) and found ' ' (0x0000 at relative Y 21, relative X 0).)'
- 6. *tc\_job\_matlcron\_003* FAILED failure in step 40: 'timeout before the specific screen buffer became available (Mismatched data at line 21, column 0. Expected 'R' (0x0052 at relative Y 21, relative X 0) and found ' ' (0x0000 at relative Y 21, relative X 0).)'

I will execute one more time when *devsrv01* will be available again

**#27 - 08/02/2013 10:27 AM - Vadim Nebogatov**

Regression 20130802\_053132 completed. There are 4 FAILED and 3 FAILED\_DEPENDENCY tests in *ts\_tests*.

```
1. tc_job_002          FAILED          failure in step 40: 'Unexpected EOF in actual at page # 1.'
2. tc_job_clock_001   FAILED          failure in step 9: '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 'r' (0x250C at relative Y 5, relative X 18).)'
3. tc_job_clock_003   FAILED_DEPENDENCY dependency chain: tc_job_clock_001
4. tc_job_clock_004   FAILED          failure in step 10: '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 'r' (0x250C at relative Y 5, relative X 18).)'
5. tc_job_clock_005   FAILED          failure in step 23: '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 'r' (0x250C at relative Y 5, relative X 18).)'
6. tc_job_clock_006   FAILED_DEPENDENCY dependency chain: tc_job_clock_001 --> tc_job_clock_003
7. tc_dc_slot_023     FAILED_DEPENDENCY dependency chain: tc_job_clock_001 --> tc_job_clock_003 --> tc_job_clock_006
```

May I commit update?

**#28 - 08/02/2013 12:39 PM - Eric Faulhaber**

I think it is ok to commit.

**#29 - 08/07/2013 05:59 PM - Vadim Nebogatov**

Eric Faulhaber wrote:

Vadim Nebogatov wrote:

Do you plan to support Alternate Buffer Pool (since 10.2B)?

Not to my knowledge. There are no BUFFER-POOL entries in the .df file of the current project.

How would this affect your implementation of this feature?

Sorry, I missed your reply, there were no email notifications.

I am not sure if it affects implementation. I asked just because I saw the following in DELETE statement reference:

```
If you define an alternate buffer for a table, you can delete a record from that buffer by using the name of the buffer with the DELETE statement
```

**#30 - 08/13/2013 12:49 PM - Eric Faulhaber**

vmn\_upd20130731a.zip update passed regression testing and was committed to bzz rev 30372.

**#31 - 08/13/2013 12:50 PM - Eric Faulhaber**

- Status changed from New to Closed

**#32 - 11/16/2016 11:43 AM - Greg Shah**

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

**Files**

---

delete-with-validation.p	139 Bytes	02/24/2013	Greg Shah
ges_upd20130224a.zip	80.5 KB	02/24/2013	Greg Shah
ges_upd20130224b.zip	80.5 KB	02/24/2013	Greg Shah
56.png	70 KB	07/25/2013	Vadim Nebogatov
23.png	35.2 KB	07/25/2013	Vadim Nebogatov
vmn_upd20130729a.zip	46.3 KB	07/29/2013	Vadim Nebogatov
vmn_upd20130729b.zip	1009 Bytes	07/29/2013	Vadim Nebogatov
vmn_upd20130731a.zip	16.2 KB	07/31/2013	Vadim Nebogatov