

Database - Bug #11913

Conversion of an ASSIGN trigger with only an OLD value generates Java that does not compile

09/28/2026 06:12 AM - Alexandru Lungu

Status:	Test	Start date:	
Priority:	Normal	Due date:	
Assignee:	Alexandru Lungu	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		reviewer:	Ovidiu Maxiniuc
billable:	No	production:	No
vendor_id:	GCD	env_name:	
case_num:		topics:	
version_reported:			
version_resolved:			
Description			

History

#1 - 09/28/2026 06:21 AM - Alexandru Lungu

- Status changed from New to Review
- Assignee set to Alexandru Lungu
- % Done changed from 0 to 100
- reviewer Ovidiu Maxiniuc added

Symptom

A schema ASSIGN trigger that declares an OLD value but no NEW value, and that also references the table buffer, converts to an assign(...) method whose parameter list contains the old-value initialization statement. The generated class does not compile.

Seen in testcases (data/trg/order-order-num-a.p, data/trg/orderline-order-num-a.p):

```
[javac] src/com/goldencode/testcases/data/trg/OrderOrderNumA.java:39: error: <identifier> expected
[javac]     public void assign(OrderOrderNumA.this.oldVal = TypeFactory.initInput((integer) _oldVal), final O
rder_.Buf _order_, final BaseDataType _oldVal)
```

Reproduction

```
trigger procedure for assign of order.orderNum old oldVal like order.orderNum.

if support.db.DatabaseTriggers:fireAssign('order', 'orderNum', string(order.orderNum), string(oldVal))
then
    return no-apply.
```

A trigger that declares both NEW and OLD and does not read the buffer converts correctly (e.g. data/trg/order-customer-num-a.p):

```
public void assign(final Undoable _newVal, final BaseDataType _oldVal)
{
    OrderCustomerNumA.this.oldVal = TypeFactory.initInput((integer) _oldVal);
    OrderCustomerNumA.this.newVal = TypeFactory.initInputOutput((integer) _newVal);
    ...
}
```

Root cause

rules/convert/variable_definitions.rules attaches the promoted OLD/NEW value initialization to the method block, and finds that block as the **first child** of the method_def:

```
<action>meth_block_id = getAst(lastid).getAncestor(-1, java.method_def).getChildAt(0).id</action>
```

When the trigger references its table, a buffer parameter (final Order_.Buf _order_) is emitted as the first child of the method_def, ahead of the block, so the initialization is grafted into the parameter list. The line was introduced with the trigger-parameter rework of 2024-12-18 (rev 15623.1.5) and is present in trunk.

Fix

Look the block up by type, as other rules do (e.g. output_parameters.rules, control_flow.rules):

```
<action>meth_block_id = getAst(lastid).getAncestor(-1, java.method_def).getImmediateChild(java.block, null).id</action>
```

Fix is in 11913a/rev. 16772. Please review.

#3 - 09/28/2026 08:10 AM - Ovidiu Maxiniuc

- Status changed from Review to Internal Test

That's a nice catch!

Indeed, the AST node for attaching the code for initialisation of the old value is not the very first child of METHOD_DEF (which in this case is the LPARENS for parameter declaration) but the BLOCK/BOGUS where the record buffer for target buffer is also instantiated.

A single code style issue: the current line is 138 chars long, well beyond the 110 character limit.

#4 - 09/29/2026 04:09 AM - Alexandru Lungu

- Status changed from Internal Test to Review

I completed testing and found just one issue; the session triggers require another form of identifying the target body; because in their case there was no proper .getAncestor(-1, java.method_def).getImmediateChild(java.block, null) resolution. Check 11913a/rev. 16773.

#5 - 09/29/2026 05:50 AM - Ovidiu Maxiniuc

- *Status changed from Review to Internal Test*

Another nice catch!. The fix makes sense.

#6 - 09/29/2026 05:56 AM - Alexandru Lungu

- *Status changed from Internal Test to Merge Pending*

The fix is regression tested across several examples of schema triggers and session triggers in different combinations. I will proceed with merging.

#7 - 09/29/2026 05:57 AM - Alexandru Lungu

- *Status changed from Merge Pending to Test*

Branch 11913a was merged into trunk as rev. 16772 and archived.