

## Base Language - Bug #6446

### temp-table:create-like adds add before-table, too

05/27/2022 04:39 AM - Constantin Asofiei

<b>Status:</b>	Test	<b>Start date:</b>	
<b>Priority:</b>	High	<b>Due date:</b>	
<b>Assignee:</b>	Ovidiu Maxiniuc	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>case_num:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			

### History

#### #2 - 05/27/2022 04:41 AM - Constantin Asofiei

This test shows that a create-like adds the before-table, too, if the source has it.

```
def temp-table ttl before-table bttl field fl as int index ix1 is unique fl.
def dataset ds1 for ttl.
```

```
temp-table ttl:tracking-changes = true.
create ttl.
ttl.fl = 100.
```

```
find first bttl.
message bttl.fl.
temp-table ttl:tracking-changes = false.
```

```
def var htt as handle.
create temp-table htt.
htt:create-like(temp-table ttl:handle).
htt:temp-table-prepare("ttl").
htt:copy-temp-table(temp-table ttl:handle, no, yes).
```

```
def var hb as handle.
hb = htt:before-table:default-buffer-handle.
hb:find-first().
message hb valid-handle(hb) hb::fl hb:name.
```

This patch solves my issue for the test:

```
### Eclipse Workspace Patch 1.0
#P p2j6129a
Index: src/com/goldencode/p2j/persist/TempTableBuilder.java
=====
--- src/com/goldencode/p2j/persist/TempTableBuilder.java      (revision 3633)
+++ src/com/goldencode/p2j/persist/TempTableBuilder.java      (working copy)
@@ -231,6 +231,8 @@
 
     /** Flag indicating the delete was explicitly called for a BY-REFERENCE handle. */
     private boolean postponedDelete = false;
+
+    private boolean implicitBeforeTable;
 
     static
     {
@@ -2384,7 +2386,7 @@
     */
     public logical tempTablePrepare(String name)
```

```

    {
-     return tempTablePrepareImpl(name, null);
+     return implicitBeforeTable ? tempTablePrepare(name, true) : tempTablePrepareImpl(name, null);
    }

    /**
@@ -2400,7 +2402,7 @@
    public logical tempTablePrepare(character name)
    {
        String tableName = name.toJavaType();
-     return tempTablePrepareImpl(tableName, null);
+     return implicitBeforeTable ? tempTablePrepare(name, true) : tempTablePrepareImpl(tableName, null);
    }

    /**
@@ -3138,6 +3140,10 @@
        return new logical(false);
    }

+     // check before-table
+     BufferImpl buf = (BufferImpl) recordBuffer.getDMOProxy();
+     implicitBeforeTable = (buf.isAfterBuffer() && buf.peerBuffer != null);
+
    addAllFields(getExistingFields(recordBuffer));

    if (handleIndex)

```

What I couldn't test in OE is the temp-table-prepare("tt1", no). - to check if the before-table from the source can be 'discarded' during prepare. This syntax is not accepted by my OE install.

### #3 - 05/31/2022 09:13 AM - Constantin Asofiei

This is committed in 6129a/13897. Ovidiu, if there are no concerns, I'll close the task.

### #4 - 05/31/2022 09:54 AM - Ovidiu Maxiniuc

- % Done changed from 0 to 100

- Status changed from New to WIP

The patch is good. You can close the task.

**#5 - 06/07/2022 07:37 AM - Constantin Asofiei**

- *Status changed from WIP to Test*