

## Database - Bug #3031

### FOR EACH can access shared transient records in case of broken transaction isolation

03/18/2016 04:20 PM - Ovidiu Maxiniuc

<b>Status:</b> New	<b>Start date:</b>
<b>Priority:</b> Normal	<b>Due date:</b>
<b>Assignee:</b>	<b>% Done:</b> 0%
<b>Category:</b>	<b>Estimated time:</b> 0.00 hour
<b>Target version:</b>	
<b>billable:</b> No	<b>case_num:</b>
<b>vendor_id:</b> GCD	
<b>Description</b>	
Related to Database - Feature #1851: database performance/scalability tuning <b>New</b>	

#### History

##### #1 - 03/18/2016 04:33 PM - Ovidiu Maxiniuc

In P4gl, if transaction isolation is broken, FOR EACH can access shared transient records that have fully matched their indexes. I have added the `uast/shared-dirty-records/leaking*` procedures and data definition file to testcases repository. They are quite well documented and also contain the expected result from running them on 4GL.

After initialization and population of the table:

- `leaking1.p` creates two records that will never be committed:
  - one (with `id = 'NEW-ID'`) that doesn't break the isolation by not matching the index;
  - and one (with `seq = 3`) that gets shared among contexts.
- the second procedure `leaking2.p` searched for both records and iterates all accessible records from table. We can notice that if the first procedure is not finished:
  - the `NEW-ID` record is isolated and i.e. not visible;
  - the `seq = 3` record is visible in both direct `FIND` statement but also in `FOR EACH` looping query.

At this moment, P2J can see the second record by direct `FIND` -ing it, but will fail to retrieve the record in the `FOR EACH` loop. This is because the `FOR EACH` loop does not check the dirty database at all when creating the list of records to be processed.