

## Database - Bug #5953

### Exceptions raised by UDFs with PRESELECT

01/06/2022 10:46 AM - Igor Skornyakov

<b>Status:</b>	New	<b>Start date:</b>	01/06/2022
<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>case_num:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			

### History

#### #2 - 01/06/2022 10:48 AM - Igor Skornyakov

It seems that we have a serious problem with PRESELECT if UDF raises an error. Consider the following code:

```
outer2:
do on stop undo, leave outer2:
  REPEAT PRESELECT EACH udf-tests NO-LOCK WHERE string(fdate, fmt) = udf-tests.fstr-result AND udf-tests.test-n
ame = 'dtostring':
  err-flag3 = false.
  message 'in preselect loop:'.
  err-flag = true.
  inner:
  do on stop undo, leave inner:
    FIND NEXT udf-tests NO-ERROR.
    err-flag = no.
    if not AVAILABLE udf-tests then do:
      message "no more records".
      leave outer2.
    end.
    message "After find next:" udf-tests.id ERROR-STATUS:ERROR ERROR-STATUS:NUM-MESSAGES AVAILABLE udf-t
ests.
    if not AVAILABLE udf-tests then leave outer2.
  end.
  message "After find: error =" ERROR-STATUS:ERROR
    "num-messages =" ERROR-STATUS:NUM-MESSAGES
    "type =" ERROR-STATUS:TYPE
    "available =" AVAILABLE udf-tests
    "err-flag = " err-flag.
  if ERROR-STATUS:ERROR then leave outer2.
END.
end.
message 'done' err-flag3.
```

It is converted to

```
OnPhrase[] onPhrase4 = new OnPhrase[]
{
  new OnPhrase(Condition.STOP, Action.LEAVE, "outer2")
};

doBlock(TransactionType.SUB, "outer2", onPhrase4, new Block((Body) () ->
{
  PreselectQuery query2 = new PreselectQuery();

  repeat("loopLabel2", new Block((Init) () ->
  {
    RecordBuffer.openScope(udf-tests);
    query2.initialize(udf-tests, "toString(udf-tests.fdate, upper(udf-tests.fmt)) = upper(udf-tests.fst
```

```

rResult) and upper(udftests.testName) = 'DTOSTRING'", null, "udftests.id asc", LockType.NONE);
    },
    (Body) () ->
    {
        errFlag3.assign(false);
        message("in preselect loop:");
        errFlag.assign(true);

        OnPhrase[] onPhrase5 = new OnPhrase[]
        {
            new OnPhrase(Condition.STOP, Action.LEAVE, "inner")
        };

        doBlock(TransactionType.SUB, "inner", onPhrase5, new Block((Body) () ->
        {
            silent(() -> query2.next());
            errFlag.assign(false);

            if (!_not(udftests.available()))
            {
                message("no more records");
                leave("outer2");
            }

            message(new Object[]
            {
                "After find next:",
                (integer) new FieldReference(udftests, "id").getValue(),
                ErrorManager.isError(),
                ErrorManager.numErrors(),
                udftests.available()
            });

            if (!_not(udftests.available()))
            {
                leave("outer2");
            }
        }
        ));

        message(new Object[]
        {
            "After find: error =",
            ErrorManager.isError(),
            "num-messages =",
            ErrorManager.numErrors(),
            "type =",
            ErrorManager.getResourceType(),
            "available =",
            udftests.available(),
            "err-flag = ",
            errFlag
        });

        if ((ErrorManager.isError()).booleanValue())
        {
            leave("outer2");
        }
    });
});

message(new Object[]
{
    "done",
    errFlag3
});

```

With 4GL the STOP is raised on

```

REPEAT PRESELECT EACH udftests NO-LOCK WHERE string(fdate, fmt) = udftests.fstr-result AND udftests.test-name = 'dtostring':

```

In particular, we do not see in preselect loop: message.

With FWD we get the SQLException only on the first query2.next() execution. **after** the in preselect loop: message is shown. What is much worse, is that the StopConditionException which is thrown after the SQLException is caught and converted to the UdfException does not cause the exit from the outer2 block, we just leave only the inner block instead.