

Base Language - Bug #4355

exception member of the CATCH block can be referenced after the block catching it

10/15/2019 10:05 AM - Constantin Asofiei

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Constantin Asofiei	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:			
billable:	No	case_num:	
vendor_id:	GCD	version:	
Description			
Related issues:			
Related to Base Language - Feature #3751: implement support for OO 4GL and st...			Closed
Related to Base Language - Feature #4373: finish core OO 4GL support			New

History

#1 - 10/15/2019 10:07 AM - Constantin Asofiei

4GL allows a case like:

```
do:
  catch ex as progress.lang.apperror:
    end.
end.

message ex.
```

Currently, the FWD conversion looks like:

```
doBlock(TransactionType.SUB, "blockLabel0", onPhrase0, new Block(
  (Init) () ->
  {
    catchError(BaseError.class, (ex) ->
    {
      // the block code here
    });
  },
  (Body) () ->
  {
    message("e");
    undoThrow("blockLabel0", e);
  });
```

See how the ex member is hidden in the lambda expression. We need to expose this to the parent block.

The good news is that the scope of the ex var is at the parent block, so we can just define a member there.

#2 - 10/15/2019 10:07 AM - Constantin Asofiei

- Related to Feature #3751: implement support for OO 4GL and structured error handling added

#3 - 10/30/2019 09:56 AM - Greg Shah

- Related to Feature #4373: finish core OO 4GL support added

#4 - 05/12/2020 01:18 PM - Greg Shah

- Status changed from New to Test

- % Done changed from 0 to 100

- Assignee set to Constantin Asofiei

This is fixed in 4231b.

Constantin: Can you link us to the discussion of your fix (and the revision of 4231b in which the fix was added)?

#5 - 05/12/2020 01:40 PM - Constantin Asofiei

Greg Shah wrote:

Constantin: Can you link us to the discussion of your fix (and the revision of 4231b in which the fix was added)?

Fixed in rev 4231b rev 11517, "The exception var in a CATCH statement must be defined too, with a `_top` suffix, to be referenced from the top-level block."

This was part of #4401-41 work, this comment:

```
repeat:
  catch errImport as Progress.Lang.Error :
    message string(errImport).
  end.
end.

delete object errImport.
```

We need to define `errImport` at the lambda AND at the top-level block. The names must be different, and at the lambda we need to do `topVarDef.assign(lambdaRef)`.

#6 - 06/20/2020 11:53 AM - Greg Shah

- Status changed from Test to Closed

Task branch 4231b has been merged to trunk as revision 11347.