

Database - Bug #6744

rollback when there are catch/finally blocks opening their own transactions

09/09/2022 12:08 PM - Constantin Asofiei

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			

History

#2 - 09/09/2022 12:13 PM - Constantin Asofiei

I'm seeing an abend with a customer app where a savepoint is no longer valid (when trying to rollback-/commit-). I've been trying to duplicate it, but I haven't been lucky - it seems to happen only when a legacy exception is being thrown and there are catch/finally blocks which calls code which opens another transaction; for example, this:

```
block-level on error undo, throw.
```

```
def temp-table ttl field f1 as int.
```

```
procedure proc1.  
  do:  
    create ttl.  
    ttl.f1 = 11.  
  
    undo, throw new Progress.Lang.AppError("problem!").  
  end.  
end.
```

```
procedure proc2.  
  do transaction:  
    create ttl.  
    ttl.f1 = 22.  
    undo, leave.  
  end.  
end.
```

```
procedure proc0.  
  run proc1.  
  finally:  
    run proc2.  
  end.  
end.
```

```
do transaction:  
  do transaction:  
    create ttl.  
    ttl.f1 = 33.  
  end.  
  run proc0.  
  catch ex as progress.lang.error:  
    message ex.  
  end.  
end.
```

```
for each ttl:  
  message ttl.f1.  
end.
```

it finds a record in FWD (the '33' record), but only when the catch ex as `progress.lang.error: code exists`.

It may be also related to the fact that we have in BlockManager code like this:

```
catch (LegacyErrorException lex)
{
    throw lex;
}
```

which just re-throws the exception without doing any rollback for the current block. I've been trying to add code to track how savepoints are created, released, rolledback and committed, but I don't understand really how these are supposed to work. For example, in this testcases:

- there is a savepoint S1 created before validation
- during validation, `lazilySetSavepoints` is being called, it creates a new savepoint S2 at a block, but the savepoint S1 at the validation gets rolledback/committed. It doesn't look write to me to keep savepoint S2 at the block.