Base Language - Bug #8084

refactor the 'execute' method for legacy classes

11/27/2023 02:23 PM - Constantin Asofiei

Status:	Test	**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 #8124: rework conversion and runtime suppo...	**New**

History

#1 - 11/27/2023 02:28 PM - Constantin Asofiei

Currently, each legacy class has a synthetic 'execute' instance method which includes initialization like instance-level buffer scopes (for temp-tables), ON ROUTINE-/BLOCK-LEVEL statements@, etc.

We can get rid of the BlockManager.externalProcedure call in the converted code, and emulate this directly in ControlFlowOps.initializeLegacyObject, only once, for the entire initialization of this instance.

The overhead comes from the fact that currently, each super-class has its own BlockManager.externalProcedure Ideally, considering that the synthetic 'execute' method can't contain application converted code, and will contain just initialization state, we can cut-down on the state being tracked when executing this (I'm thinking about ProcedureManager stacks).

#2 - 11/28/2023 03:29 AM - Alexandru Lungu

Contantin, just a question here: I thought we open scopes in externalProcedure due to the fact that we are aware **only** at conversion which buffer scopes should be opened. How can we know at run-time which buffers to open? Do we use reflection and identify the global record buffers?

#3 - 11/28/2023 03:31 AM - Constantin Asofiei

What I meant is to emit all the code (like buffer scopes) directly in the Java method, and do not enclose it in BlockManager.externalProcedure. All legacy object instantiation is done via ObjectOps APIs, so we can call the push/pop scope code there. So this will perform only one scope processing per instance, not one for each super-class, as it is done now.

#4 - 11/28/2023 03:49 AM - Alexandru Lungu

Oh right, this makes sense - getting rid of the BlockManager API. This way we avoid doing ProcedureManager work.

#5 - 12/01/2023 06:00 AM - Constantin Asofiei

- Status changed from New to WIP

- Assignee set to Constantin Asofiei

Created task branch 8084a from trunk rev 14853

#7 - 12/13/2023 06:40 AM - Constantin Asofiei

- % Done changed from 0 to 100

- Status changed from WIP to Review

The changes for this and #8069 are in 8084a rev 14873. Please review.

#8 - 12/13/2023 07:10 PM - Greg Shah

Code Review Task Branch 8084a Revisions 14870 through 14873

The changes are good.

As an additional improvement, I wonder if we should change how BLOCK-LEVEL and ROUTINE-LEVEL statements work.

Instead of emitting onBlockLevel(Condition.ERROR, Action.THROW); or onRoutineLevel(Condition.ERROR, Action.THROW); into the execute() method, I think we should just emit a class-level annotation. It is really just a boolean flag anyway. This would be the case for both procedures and OO .cls files. The result would be much cleaner (code wise). I also wonder if it might be more efficient since some additional execute() methods will now be empty.

I also don't know why we emit the condition and action parameters since they cannot vary and really have no meaning. But if we get rid of these calls anyway, it would be even better.

#9 - 12/14/2023 09:32 AM - Constantin Asofiei

Greg Shah wrote:

> As an additional improvement, I wonder if we should change how BLOCK-LEVEL and ROUTINE-LEVEL statements work.

I agree, but this would take a little more work.

Can 8084a be committed as is?

#10 - 12/14/2023 09:47 AM - Greg Shah

- Status changed from Review to Internal Test

Yes, if you are complete with testing.

#11 - 12/14/2023 11:34 AM - Constantin Asofiei

Greg Shah wrote:

> Yes, if you are complete with testing.

I haven't converted all apps, but I think the conversion rules are good.

#12 - 12/14/2023 11:38 AM - Greg Shah

If you've done enough, you can merge.

#13 - 12/14/2023 11:56 AM - Constantin Asofiei

Branch 8084a was merged to trunk rev 14876 and archived.

#14 - 12/14/2023 12:18 PM - Greg Shah

- Status changed from Internal Test to Test

#15 - 12/14/2023 12:24 PM - Greg Shah

- Related to Feature #8124: rework conversion and runtime support for BLOCK-LEVEL and ROUTINE-LEVEL statements added