

Conversion Tools - Bug #3114

changing SharedVariableManager to call TransactionManager.register[Global]()

05/17/2016 04:29 PM - Greg Shah

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

#1 - 05/17/2016 04:35 PM - Greg Shah

It seems like we could change SharedVariableManager to call TransactionManager.register[Global]() when a new [global] shared variable is created and drop those register calls from the business logic.

Eric has found a case where we are registering duplicate references in the global undoables list. If you RUN an external procedure that has a def new global shared var and you do that RUN in a loop, then the same global shared variable will be added to the global scope of the undoables list multiple times (once for every loop iteration). He is putting some protection logic into the TM to not add the reference if it already exists. But it seems like the SVM could hide the registration completely AND it always knows if something is created (then we register) versus imported (no registration).

The only issue I see is that it will be called during the constructor for an external procedure, before the external proc scope opens. So we would have to implement a deferred registration. Other than that, I don't see much of a problem.

Constantin: Am I missing anything here?

#2 - 05/18/2016 10:57 AM - Constantin Asofiei

Greg, you are correct that SVM should do the register, and not emit explicit TM.register calls. And yes, we need to do this in a deferred registration; also, the GLOBAL vars need to be added in a deferred way, too, otherwise we will leak global vars, if the instantiation of the external procedure fails for some other reason (i.e. another shared var error after a global was already registered).

Otherwise, I don't see anything else of concern.