IRD to Composer Migration

IRD Variable Handling

12/19/2025

IRD Variable Handling

In IRD 8.0.1, additional variable scopes have been introduced that do not exist within Orchestration. These new scopes for variables are not supported because of the significant ramifications on how variables can be accessed and/or defined with Orchestration Server's ability to operate in a fully distributed, multi-threaded environment.

Orchestration only supports LOCAL scoped variables as these map directly to variables that can be created within SCXML. For the remaining sets of more global scopes exposed by IRD, the following table provides the current recommended approach. In general, the use cases for these are seen to be more business application related and as such should be stored within a centralized database that can manage concurrent access and transactional-based locking, or by other similar means that can provide global access control to data.

Variable Scope	Description	Migration Notes
SCRIPT	A SCRIPT variable is created when the strategy is preloaded and is destroyed when the strategy is released from URS's memory. Every run of the strategy can change the value of the same SCRIPT variable. Use SCRIPT variables with caution. (One possible use is as counters over all interactions processed by the same strategy.) Values are shared within a single strategy.	If the SCRIPT variable is only used for rea purposes, then this could be populated w the provision definition and provided wh script is invoked. However, this variable then need to be passed as a parameter t subroutines or other workflows. If the va is not read only, then a DB block or othe mechanism would need to be used.
USER	A USER variable is created when the tenant is active and is destroyed when the tenant is released from URS's memory. Everything running within the tenant can change the value of the same USER variable. Values are shared within the current tenant.	Since a single ORS node may provide su for multiple tenants, it is recommended t this variable be stored and operated upo a centralized DB accessed via the DB blo
GLOBAL	A GLOBAL variable is created when the particular instance of URS runs and is destroyed when the URS stops running or is released from memory. Everything running on this instance of URS can change the value of the same GLOBAL variable. Values are shared within the entire URS instance.	Since Sessions may be swapped betwee Orchestration nodes, this is not really a r concept within Orchestration because se are not sticky. Any session can be execut a multitude of nodes through its life cycl There is currently no other recommenda this other than possibly a centralized DB via DB blocks
INTERACTION	An INTERACTION variable is created when a particular interaction is active and is destroyed when the interaction ends. The value of an INTERACTION variable for one interaction has no effect on the value of the same INTERACTION variable for another interaction. Values are shared for the current interaction only	Since this is related to the actual interac udata can be used to accomplish this tas would ensure that the data is shared acr Orchestration Nodes as well as any othe component whenever an interaction is o on.