

Database - Feature #4155

remove conversion dependency on metadata .df file

07/26/2019 11:45 AM - Eric Faulhaber

Status:	WIP	Start date:	
Priority:	Normal	Due date:	
Assignee:	Greg Shah	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		vendor_id:	GCD
billable:	No		
Description			
Related issues:			
Related to Base Language - Feature #3810: SECURITY-POLICY and other security ...			Closed

History

#1 - 07/26/2019 11:45 AM - Eric Faulhaber

- Related to Feature #3810: SECURITY-POLICY and other security features added

#2 - 07/26/2019 11:55 AM - Eric Faulhaber

FWD currently parses a metadata .df file exported from the data dictionary as part of early schema dictionary processing. We need to remove this dependency and replace it with a different source of this metadata information. Ideally, this new source would be transformed easily into the .dict AST format that we use as an input to the SchemaDictionary.

#3 - 06/10/2020 05:33 PM - Greg Shah

- Assignee set to Greg Shah
- Status changed from New to WIP
- Tracker changed from Bug to Feature

Our ultimate objective with schema metadata is to eliminate the conversion of these DMOs and instead rely upon metadata DMOs that are hard coded as part of FWD. This makes sense for multiple reasons:

- In our experience, the 4GL metadata is always upwards compatible, so there is no reason to simulate multiple different versions. We can have a single version that is very recent and it can support all applications.
- Having different metadata implementations in different conversions means:
 - More possible bugs.
 - Harder to diagnose issues because there is more complexity.
 - Extra inputs, configuration and points of failure for both conversion and runtime.
 - The runtime must have special code to deal with and hide the fact that these metadata DMOs are generated in customer packages. In other words, the runtime will be simpler when it can directly reference built-in metadata DMOs.

I see no advantage to the current approach.

Although I do not plan to shift to this model as part of this task, I don't want to create a separate repository of the metadata configuration. Since we plan to have dedicated DMOs for these metadata tables, then we can create them now and use the annotations to store all the other configuration needed to replace the standard.df. Later on, we can modify the runtime to use these new DMOs and at that time we would "cut out" the conversion part of the metadata processing. Until that time, the DMOs would only be used as a source of metadata.

As noted above, the easiest path here is to directly create .dict ASTs from these hard coded metadata DMOs (and their annotations). Then the schema loading will need some modification to avoid parsing the metadata and instead to read the generated .dict ASTs. From there (in the short term), the conversion process will run unchanged.