Runtime Infrastructure - Feature #12118

implement a 4GL-style memory analyzer view in FWD

11/04/2026 03:44 PM - Constantin Asofiei

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**reviewer:**	
vendor_id:	GCD	**production:**	No
case_num:		**env_name:**	
version_reported:		**topics:**	
version_resolved:			

Description

History

#1 - 11/04/2026 03:48 PM - Constantin Asofiei

The idea here is to build a view, which can be opened in the Web client via the toolbar menu or other shortcut, where we can show:

- high-level view of persistent programs, allocated objects, dynamic resources (temp-tables, datasets, etc) - we can use both 4GL-style chains from where to gather this info, or even from the FWD-specific storage of these
- expanded view of each type, where we can show who allocated this (INSTANTIATING-PROCEDURE, etc)

The point here is to provide easier insights into the runtime, at a 4GL-level view, and not rely only on the Java heap dumps to investigate these.

Also, it would be interesting to get a REST API and dump this, automatically, from all session/contexts, in some JSON format.

#2 - 11/04/2026 11:55 PM - Greg Shah

Why not track and expose this via JMX?

#3 - 11/05/2026 01:42 AM - Constantin Asofiei

Greg Shah wrote:

> Why not track and expose this via JMX?

The upside I see is ease of access, you have this in the browser, and don't need to get into another tool. JMX can give you total live counts for example, but doesn't allow you to easily dig into and get the details for i.e. a certain resource.

#4 - 11/05/2026 07:48 AM - Alexandru Lungu

> JMX can give you total live counts for example, but doesn't allow you to easily dig into and get the details for i.e. a certain resource.

JMX can give you whatever you live. QueryProfiler JMX is building a JSON with queries, the cache hit/miss, etc. Our JMX layer is limited to timers and counters, but, as mentioned, it can be extended to cover any kind of output.

#5 - 11/06/2026 06:35 AM - Greg Shah

To memorialize our in person discussion, the plan is to extend our JMX support with these new features AND to add a nice diagnosis UI in the web client to display these details. The JMX approach is consistent with the rest of this kind of instrumentation and it allows a customer that already uses JMX to see all this detail. We still gain a nice end-user ability to view this data but we don't lose the benefits of JMX.