

## Database - Feature #1651

### add support for dynamic database features

10/21/2012 05:22 PM - Eric Faulhaber

<b>Status:</b>	Closed	<b>Start date:</b>	04/02/2013
<b>Priority:</b>	Normal	<b>Due date:</b>	07/16/2013
<b>Assignee:</b>		<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Runtime Support for Server Features	<b>version:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			
<b>Subtasks:</b>			
Feature # 1652: add conversion and runtime support for dynamically prepared temp-tables			<b>Closed</b>
Feature # 1955: add conversion support for CREATE-TEMP-TABLE statement			<b>Closed</b>
Feature # 2122: add runtime support for dynamically prepared temp-tables			<b>Closed</b>
Feature # 2128: generate and load DMO interface and implementation classes for dynamic ...			<b>Closed</b>
Feature # 1653: add conversion and runtime support for dynamically prepared queries			<b>Closed</b>
Feature # 1954: add conversion support for CREATE QUERY statement			<b>Closed</b>
Feature # 2120: add runtime support for dynamically prepared queries			<b>Closed</b>
Feature # 2127: complete runtime support for dynamically prepared queries			<b>Closed</b>
Feature # 1654: add conversion and runtime support for dynamic buffer creation and cleanup			<b>Closed</b>
Feature # 1953: add conversion support for CREATE BUFFER statement			<b>Closed</b>
Feature # 2125: add runtime support for dynamic buffer creation and cleanup			<b>Closed</b>
Feature # 1655: add conversion and runtime support for certain dynamic database attribu...			<b>Closed</b>
<b>Related issues:</b>			
Related to Database - Feature #2235: fix deployment issues related to dynamic...			<b>Closed</b>

### History

#### #1 - 10/21/2012 05:24 PM - Eric Faulhaber

Requires binary TRPL as a pre-requisite, since conversion will have to occur at runtime, on the fly.

#### #2 - 10/31/2012 04:06 PM - Greg Shah

- Target version set to Milestone 7

#### #3 - 01/04/2013 01:44 PM - Eric Faulhaber

We need to determine whether dynamic database features are scope-agnostic in terms of resolving database references; i.e., do we need to keep track of which buffer names should be promoted based on the sequence of open buffer scopes at the time a dynamic feature is invoked? This is done by the SchemaDictionary at conversion time for static buffer references, to enable disambiguation of buffer name references. Are dynamic database features also aware of this scoping? We will need test cases.

If we do find we need to account for buffer name promotion at runtime, note that all per-context buffer scope information is maintained by the BufferManager, so we could get it from there on demand, if necessary (with some changes to that class).

#### #4 - 10/08/2013 08:34 AM - Eric Faulhaber

Note that any use of conversion features at runtime will require the use of existing or new conversion rule-sets and configuration information. Since we don't expect the entire conversion project to be available on a runtime installation, any such resources will need to be added to either the p2j.jar

or, if customer-specific, to an application jar file. This is not critical for an early working version, but it should not be precluded by our design and implementation.

**#5 - 10/08/2013 08:46 AM - Greg Shah**

To clarify:

1. We are not going to use a binary implementation of TRPL for our first implementation.
2. We don't want to require the entire conversion process to be used, when we only need to convert small snippets of code/expressions.
3. The progress.g already has the expr rule that is a valid entry point for use in parsing a single expression.
4. Re-use the TRPL conversion rulesets as much as is possible.
5. Make ruleset changes as needed to allow common rulesets to differentially process depending on whether used in the full process or the dynamic database processing at runtime.
6. I expect that some custom rulesets will be needed to manage the overall dynamic database runtime conversion process. I just want those to reuse as much of the common code as is reasonably possible.
7. Long term, we want the standard rules to be loaded from the p2j.jar instead of from the file system.
8. Long term, we want the customer-specific rules and the customer-specific project configuration (e.g. p2j.cfg.xml) to be loaded from the customer's application jar.

**#6 - 01/27/2014 10:23 AM - Eric Faulhaber**

- Status changed from New to Closed

**#7 - 11/16/2016 11:42 AM - Greg Shah**

- Target version changed from Milestone 7 to Runtime Support for Server Features