

## Database - Bug #7489

### Lazily initialize DMO signature

07/06/2023 08:36 AM - Alexandru Lungu

<b>Status:</b>	WIP	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Alexandru Lungu	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>case_num:</b>	
<b>billable:</b>	No	<b>version:</b>	
<b>vendor_id:</b>	GCD		
<b>Description</b>			
<b>Related issues:</b>			
Related to Database - Feature #7404: Trasform replace-mode into append-mode w...			<b>Closed</b>

### History

#### #1 - 07/06/2023 08:39 AM - Alexandru Lungu

- Status changed from New to WIP
- Assignee set to Ștefan Roman

DmoSignature is eagerly computed in the DmoMeta constructor. This may not be ideal for dynamic temp-tables that won't ever use the signature. There are several signatures here:

- quick signature: used to check if we can do a simple fast-copy between temp-tables
- explicit signature: used by buffer-copy
- schema signature: used when trying to optimize fast-copy between tables, but ensure they have the same SQL structure, so we can use INSERT INTO SELECT FROM without property mapping.

Some of the signatures won't be ever used, so it doesn't make sense to compute them.

#### #2 - 07/06/2023 08:40 AM - Alexandru Lungu

- Related to Feature #7404: Trasform replace-mode into append-mode when target table is empty added

#### #3 - 07/06/2023 08:46 AM - Greg Shah

Can these signatures be computed at conversion time? If so, let's compute them once and write them into the DMO as public static final constants.

#### #4 - 07/06/2023 08:54 AM - Alexandru Lungu

Do you think of any performance gain if we compute them at conversion time? The static DMOs will have their signatures computed only once and cached when registered for the first time at the server-side. This means that this computation is done on a cold run anyway. My concern was mostly related to how we handle dynamic tables, whose DMO is created, loaded and freed at run-time. These will have their signatures computed right at load time, so non-deterministically at server's run-time.

Also, note that the signatures aren't quite good looking :) So dumping them at conversion time in the DMO file will cause a very long string to be graft.

**#5 - 07/06/2023 09:35 AM - Greg Shah**

If there is a benefit to lazily computing them, then there is a benefit to compute them in advance. Of course, there is nothing we can pre-compute for the dynamic case.

Also, note that the signatures aren't quite good looking :) So dumping them at conversion time in the DMO file will cause a very long string to be graft.

I understand. Performance matters more than having a few ugly strings.

**#6 - 07/10/2023 02:01 AM - Alexandru Lungu**

Please delay the work here until I do some metrics on how much time is spent on creating this signatures first. This task was mostly created after the introduction of the new [#7404](#) (sqlSignature).

**#7 - 08/21/2023 05:23 AM - Alexandru Lungu**

- Assignee changed from Ștefan Roman to Alexandru Lungu

**#8 - 08/22/2023 09:02 AM - Alexandru Lungu**

On a server start-up with 2251 DMO loaded, I got 40ms spent in quick signature building, 50ms spent in explicit signature building and 35ms in schema signature building.

On a **cold-run**, another 1151 DMO were loaded. For there, 14ms were spent for quick, 22ms were spent for explicit and 14ms spent for schema. On **warm-runs**, I didn't had any DMO loaded. Now I see that DynamicTableHelper has a cache, so dynamic created DMO do not trigger new signatures computation.

The times are very low, so it the signature computation can't be further optimized.

This task can be rejected.