

Database - Feature #5165

disallow usage of an incompatible database instance

02/25/2021 05:08 AM - Greg Shah

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		vendor_id:	GCD
billable:	No		
Description			

History

#2 - 02/25/2021 05:48 AM - Greg Shah

In OE, the database metaschema has CRC values stored for each table, index and schema trigger. When compiling 4GL to r-code, the CRC values of the resources being used are stored in the r-code. At runtime, these are checked against the database metaschema to detect if the code was compiled against a different version of those resources. See [What are the different types of CRC stored in the database's metaschema?](#)

In FWD, at this time, we have no equivalent facility. We have found that it is all too easy to have converted application code which uses databases which are incompatible. There are 2 sources of incompatibility:

- structural differences in .df
 - The converted .jar (and artifacts) are used for both application execution and for database import.
 - When 2 different versions are used, any structural differences will cause an incompatibility that currently will only be seen in runtime failures.
 - These failures will occur only in parts of the application which are:
 - executed
 - incompatible in structure
 - This means that the application might operate for some time without anyone noticing that there is a problem.
 - It might cause serious issues such as data corruption and it will certainly cause instability and what appear as bugs in FWD.
 - Some .df changes are not structural (e.g. labels, format strings, descriptions).
 - I think that changes to data types, extents, names and other structural elements are the problem here.
- changes in FWD
 - Changes to how FWD implements some persistence features can cause incompatibility.
 - schema conversion
 - data import
 - query conversion
 - ?
 - The identical .df used with 2 versions of FWD can result in an incompatible environment.
 - This has all the same implications as a structurally different .df.

I think the OE approach is over-complicated and makes little sense. We would not want the application to only break for changed tables or indexes. I see no way to ensure data integrity with such an approach.

On the other hand, at least they detect the incompatibility before it tries to execute. We need to implement a mechanism to detect this problem. I suggest the following:

- At conversion time, we should calculate a unique "fingerprint" for the structural elements of the .df. Only a single value is calculated for the entire .df. We don't need per-table or per-index values.
- At FWD build time, we should calculate a unique "fingerprint" for all schema/persistence conversion rulesets and Java code upon which compatibility depends.
 - This could be something that is automated and it may be somewhat simplistic (e.g. any change to certain files is detected as an incompatible difference).
 - An alternative is to manually encode a fingerprint that changes only when a programmer determines an incompatible change was made.
 - The manual approach will be error prone. It might be forgotten (leading to data corruption/failures) or it might be changed unnecessarily because it is hard to know when to change it. This second problem is no worse than the automated build solution.
- Both fingerprints would be stored in:
 - conversion artifacts that can be read from the application jar
 - the database at data import time
- At every database connection, both fingerprints would be compared between the application jar and the database instance. Any mis-match would disallow the connection.
 - If this occurs for a database that is connected by default, the server would not be allowed to start (and a very obvious/understandable error would be written to the log).
 - If this is on a connection that is made dynamically then we probably should do whatever the 4GL does in this case. I suspect they raise a

STOP condition.

We need to consider how to handle updates to these fingerprints over time.

- Needed because of a new FWD version that doesn't require dump/load.
- Needed due to manual changes applied to the structure of the database (e.g. an application upgrade that doesn't require dump/load).

#3 - 02/25/2021 09:42 AM - Constantin Asofiei

We can also validate the schema specified at the permanent DMOs with the database tables/indexes.