

Database - Feature #6347

change the name conversion to provide a configurable option to not emit "snake case"

05/10/2022 09:36 PM - Greg Shah

Status:	WIP	Start date:	
Priority:	Normal	Due date:	
Assignee:	Eric Faulhaber	% Done:	90%
Category:		Estimated time:	0.00 hour
Target version:		version:	
billable:	No		
vendor_id:	GCD		
Description			

History

#1 - 05/13/2022 10:45 AM - Greg Shah

The intention is to keep names verbatim except for reserved SQL keywords, hyphens and other invalid SQL name constructs.

#2 - 05/13/2022 10:59 AM - Eric Faulhaber

I am considering the best place to add this configuration. Currently, we configure persistence name conversion information in three places:

- the name converter code itself; not really "configuration", as it is hard-coded to snake case for persistence use at the moment, but this is one point where the determination of how to convert a name takes place.
- datanames.xml (for coarse name substitutions)
- schema hints (for granular name substitutions)

Another option is in p2j.cfg.xml, within a profile or in the global scope.

I am leaning toward p2j.cfg.xml, since the decision of the type of name conversion is coarse and as such does not require a lot of configuration. By putting it here, we can take advantage of the profile construct, in case a user needs the conversion to be different by profile.

#3 - 05/13/2022 11:35 AM - Greg Shah

I am leaning toward p2j.cfg.xml

Agreed.

#4 - 05/18/2022 12:55 PM - Eric Faulhaber

Ovidiu, Constantin, with the addition of the profiles feature in p2j.cfg.xml, what is the appropriate way now to check schema/namespace configuration

information?

For example, I've added a conversion attribute to the schema/namespace XML tag. I need to check it during name conversion from various places in Java or TRPL code. To get similar information previously, I would call `Configuration.getSchemaConfig()` and get more specific namespace information from there. But it seems this call now returns the default profile's schema configuration, not that of a particular profile.

How do I know if I'm operating in the context of a specific profile (i.e., not the default profile)? Will this same call get me the active profile transparently (i.e., is there only ever one active profile, whether it is the default one or a specific one?), or do I need to fetch the current profile's name from somewhere and request that specific profile's schema configuration? There does not seem to be an API for this (and it would be messy anyway), so I must be missing something.

#5 - 05/18/2022 01:29 PM - Constantin Asofiei

Eric, the default profile is meant to be 'switched' with the file's profile, when TRPL code is ran (see `config.withFileProfile(path)`; call in `PatternEngine.run(String)`).

So, when processing the AST of a 4GL program, this will switch both the `Configuration` and `SchemaConfig` to have the proper file.

But... I think there may be a problem when processing the middle part, the `.schema`, `.dict` or `.p2o` files... I'll check and get back to you.

#6 - 05/18/2022 01:42 PM - Constantin Asofiei

Eric, please confirm that your issue is with the 'middle' part, where the schemas/p2os are processed.

If yes, we have a problem: the same `.df` file may be part of multiple profiles. How do you choose the profile?

#7 - 05/18/2022 01:48 PM - Constantin Asofiei

The idea with the conversion profiles was that you are splitting the 4GL converted source code in unique sets, with each file in its own profile. This is not enforced for schema files, so the middle part has no knowledge about what profile a schema belongs to, and it uses the default profile.

#8 - 05/18/2022 02:00 PM - Constantin Asofiei

And something else to note: considering that a schema can be part of multiple profiles, I don't think is correct to keep this schema configuration at the profile, it should be in the 'global/default profile'. Because I assume each schema (no matter in how many profiles is used) must be converted the same.

#9 - 05/18/2022 02:16 PM - Eric Faulhaber

Yes, I think the schema/namespace configuration should only be defined once, in the default profile, and the individual namespaces needed for a specific profile should reference those namespaces as needed to satisfy the parsing/conversion needs of that profile.

That also suggests that M0/1 should not be run for persistent schemas in the context of non-default profiles, but I'm not sure how this should be implemented practically.

Consider also that M0/1 still needs to run for the temp-tables defined in each profile. However, I think we have a problem currently, in that M0/1 for temp-tables would overwrite those created in a previously run profile, if the names are the same across profiles. It seems multiple profiles should not be converted in the same conversion project currently, since there is only one output area for temp-table DMOs.

These are some fundamental profile issues that are not directly related to this task. We should open a separate tracker for these.

#10 - 05/18/2022 02:22 PM - Eric Faulhaber

Constantin Asofiei wrote:

Eric, please confirm that your issue is with the 'middle' part, where the schemas/p2os are processed.

If yes, we have a problem: the same .df file may be part of multiple profiles. How do you choose the profile?

I am not using multiple profiles in my current test project, so I am not choosing a profile currently. I was not so much reporting a problem as I was asking how this works now. I am writing code which needs to query schema namespace configuration, and I realized I wasn't taking the active profile into account.

But perhaps that *is* the way to do it. From what I understand you are saying, the active profile is determined up the stack (e.g., by the ConversionDriver) and it is transparent to TRPL and Java conversion code. Is that right?

#11 - 05/18/2022 02:42 PM - Constantin Asofiei

Eric Faulhaber wrote:

But perhaps that *is* the way to do it. From what I understand you are saying, the active profile is determined up the stack (e.g., by the ConversionDriver) and it is transparent to TRPL and Java conversion code. Is that right?

Yes, the profile support is transparent in terms of the TRPL and other conversion code. I have a patch for processing the program .schema/.p2o, for the temp-tables (this was not switching profiles properly), will commit tomorrow.

#12 - 05/20/2022 12:33 AM - Eric Faulhaber

- *Status changed from New to Review*

- *% Done changed from 0 to 100*

Implemented in 6129a/13860.

Adds a new, optional conversion attribute to the schema/namespace XML element in p2j.cfg.xml. Permitted values are minimal and verbatim.

minimal mode causes the conversion of SQL table/column/sequence names with minimal change: case is preserved (though it may not be honored by the target database internally); only illegal characters are replaced.

verbatim mode keeps the legacy name completely intact and surrounds the name in double quotes.

Neither mode has been fully tested, though I've spot-checked output for minimal mode and it seems correct. verbatim mode in particular probably will cause downstream conversion/compile problems without additional attention. For instance, when the converted name is used in DMO annotations, the double quotes likely will require some attention (i.e., escaping) to behave nicely as Java annotation information.

#13 - 05/20/2022 12:33 AM - Eric Faulhaber

- Assignee set to Eric Faulhaber

#15 - 06/07/2022 08:14 AM - Constantin Asofiei

- % Done changed from 100 to 90

- Status changed from Review to WIP

Eric, I'm placing this back to WIP as this prevents 6129a to be tested with other projects - the issue is #6277-128, the FWD's hard-coded meta table/field names.

#16 - 10/04/2022 11:07 AM - Eric Faulhaber

Constantin Asofiei wrote:

Eric, I'm placing this back to WIP as this prevents 6129a to be tested with other projects - the issue is #6277-128, the FWD's hard-coded meta table/field names.

Constantin, is this still an issue?

#17 - 10/04/2022 11:31 AM - Constantin Asofiei

Eric Faulhaber wrote:

Constantin Asofiei wrote:

Eric, I'm placing this back to WIP as this prevents 6129a to be tested with other projects - the issue is #6277-128, the FWD's hard-coded meta table/field names.

Constantin, is this still an issue?

Yes.

#18 - 10/31/2023 04:49 PM - Greg Shah

Can this be closed? Is there something remaining to do?