

NextRelease

Torque 6

TODO

Torque 5

See what [changed](#).

Torque 4

The plans of the Torque development team for Torque 4 are summarized below. Please change the consensus list only after discussing your proposal on the Torque developers list.

Consensus is to do the following things:

- Next main version should be 4.0. There can be major changes in the version, but the 'look and feel' should change as little as possible. The idea would be that one can recompile one's Torque 3.2 or 3.3 project with the new Torque version and it should run with minor changes only.
- Simplify the repository structure: No svn externals any more. This would mean that all Torque components can only be released simultaneously, but as we have done so in the past anyway and compatibility problems would arise if we would not do so, this should not be problematic.
- Kick out village. This would mean: create the sql to insert / update objects ourselves; do the Torque type / SQL type mapping of variables ourselves; change the way to create objects from jdbc result sets.
- Change the way to handle non-record-type queries (we used village for that until now).
- Criteria should not be modified any more inside queries.
- Disentangle Query description and SQL creation code. It would be ideal to port the MVC paradigm to the SQL creation: The criteria is the model (what do I want to query), the view would create the SQL from the Query, and the controller would be the Peer class which executes the query.
- Use the Criteria object only to hold query data and another object to hold update data. Criteria should not extend Hashtable any more nor should it implement the map interface, cause query data is not a Map in any sense (e.g. the same column names can appear more than once in a criteria)
- Use a column object to hold the column name in the runtime instead of Strings. So one could remove all that distributed guesswork of "what is the table name, do we have a schema name or not" etc. This would also remove the ambiguity which exists now that a String can be a value or a column name.
- Support creation of database views in the generator and access to database views in the runtime.
- Use a custom velocity renderer to have nicely trimmed generated code.
- Have a non-static version of Peer classes. Implementation details are under discussion.
- Evaluate whether ddlutils can be used for some generator tasks.

Under discussion:

- Switch to Maven 2 as build system. Keep the maven 1 plugin. For building the Maven 1 plugin, we need to retain a skeleton Maven 1 build.
- Make the generator more generic, i.e. change it to a general code generation tool.
- Remove support for Primitives in data objects.

Problematic:

- Use Java 5 style generics and enumerations. The problem is that Java 5 code does not run on jdk 1.4

In a later version of Torque 4:

- Support for object level views

In a later version of Torque 4, under discussion:

- Support for lazy loading

Further Suggestions

- If specified the torque.database.name property should be user rather than name attribute of the database tag. This would make it much easier to handle multiple instances of an application on a single system, e.g. dev and test database instances being generated from a single schema.
- Is it just me or are a number of the torque properties required to be declared outside of the file referred to in torque.contextProperties? Could the Torque Maven plugin read the properties in as well, making it so that project.properties need only define torque.contextProperties? This would make it much easier to handle multiple application instances built from the same environment (I run a number of filters across my properties file to customise it for the application/database instance I am generating before I execute the generator).
- Provide for an encoding attribute on the database element - to be used in create database statement for PostgreSQL (and presumably others).

– [ScottEade](#) 2003-10-09

- The jdbc target (torque:jdbc goal in Maven-speak) needs to generate the correct DDL SQL for PostgreSQL serial columns (see: [PostgreSQLFAQ](#))
– ?[ScottEade](#) 2004-08-20

-
- Allow override of native limit on the size of result set size. (ie. make it possible to *not* use a rownum query for Oracle, TRQS187)
 - Allow override of case insensitive sorting (using UPPER) on Strings (TRQS188)
-

- Correctly generate index

– Domenico Loiacono 2004-08-05

- move all db specific stuff from the generated classes to the base classes (so you don't have to regenerate the modell for different dbs). In other words removed the hardcoded database name inside [BasePeers](#) and [MapBuilder](#) classes, and replace it with a variable that is initialized through the properties file.