PostgreSQLFAQ

Can I use Torque with PostgreSQL

Yes, it would appear that several people are using Torque and PostgreSQL together successfully. You do however need to be aware of the issues outlined in the rest of this FAQ.

I have problems using idMethod='native'. What's up?

Note: This issue has been dealt with in cvs (HEAD and TORQUE_3_1_BRANCH) - if you are using release 3.1.1 or later you do not need to use the workaround described below. – ScottEade 2004-08-20

PostgreSQL creates a sequence with a special name for the autoincrement. In order to get Torque to interact gracefully with this sequence, you need to tell it the sequence name. In addition, Postgres restricts the sequence name to 30 characters, which means that the names get truncated strangely. Thus, to get everything to work, you basically need to do the following:

1. include a <id-method-parameter name="seqName" value="Tablename_columnname_seq"/> in every table that has an "autoincrement" column.
2. If the "tablename_columnname" combination is longer than 26 letters, it will get truncated to 26 letters by Postgres . You need to figure out the truncated name. The best way to find it is:
    - create the database via the usual Torque methods.
    - go into psql and execute "\ds tablename" - this will show you the name of the sequence.
    - go back to your schema file and set the value parameter to the name of the sequence.
    - recreate the database via the usual Torque methods. Now everything should work.

An example of a table definition (without truncation issues) is:

```
<table name="TypeCategory" createLogTable="true">
  <column name="TypeCategoryId" autoIncrement="true" required="true"
          primaryKey="true" type="INTEGER"/>
  <column name="Name" required="true" size="32" type="VARCHAR"/>
  <column name="Descr" required="true" size="64" type="VARCHAR" />
 <column name="CreateTime" type="TIMESTAMP" />
  <column name="UpdateTime" type="TIMESTAMP" />
  <unique>
    <unique-column name="Name"/>
  </unique>
  <id-method-parameter name="seqName" value="TypeCategory_typecategoryid_seq"/>
</table>
```

An example of a table definition with truncation issues is:

```
<table name="AssociateLogin">
  <column name="AssociateLoginId" required="true"
          autoIncrement="true" primaryKey="true"
          type="INTEGER"/>
  <column name="AssociateKey" type="INTEGER"/>
  <column name="Handle" size="64"
          type="VARCHAR" />
  <column name="Password" size="250"
          type="VARCHAR" />
  <column name="CreateTime" type="TIMESTAMP" />
  <column name="UpdateTime" type="TIMESTAMP" />
  <unique>
    <unique-column name="Handle"/>
  </unique>
  <foreign-key foreignTable="Associate" onDelete="cascade">
    <reference local="AssociateKey" foreign="AssociateId"/>
  </foreign-key>
  <id-method-parameter name="seqName" value="associatelogi_associatelogi_seq"/>
</table>
```

The situation with PostgreSQL 7.3 would appear to be quite different. It appears that in SQL it is enough to simply declare a column as "serial" and the necessary sequence will be created with the correct name (truncation to 26 characters no longer occurs) and will be dropped automatically when the table is dropped. This would mean that Tourque could be updated to not generate the code that creates and drops the sequence and the the id-method-parameter element is probably no longer necessary (if it is used as part of the id generation code, and I do not think it is, then the necessary value can be derrived). – ScottEade 2003-02-06

It's a valid point. The above code works for 7.2 and Torque 3.0b3, and may be different for other versions. I think the <id-method-parameter> element is necessary because the Torque model expects the sequence name to be "tablename_seq" but Postgresql creates a sequence which is called "tablename_columnname_seq". I think once an object is inserted into the database, Torque needs a method to find out the ID of the object. It does this by fetching the current value of the sequence (which should be equal to the last value inserted into the database.) If Torque cannot figure out the sequence name correctly, then it cannot find the appropriate ID and everything goes south. (I might be incorrect about this - it may be only an issue of making sure the sequence is created and dropped, in which case Scott is absolutely right.)

--PeterHamlen 2003-02-06

So this is basically a bug in Torque - i.e. it should generate "tablename_columnname_seq" rather than "tablename_seq" and perhaps still allow for the <id-method-parameter> element to allow for versions of PostgreSQL that truncate the sequence name to 26 characters. – ScottEade 2003-02-07

To get sequences to work I added the <id-method-parameter> elements mentioned above, but then had to remove the sql that drops/creates the sequences that was generated by torque:sql before executing it with torque:insert-sql. – ScottEade 2003-10-14

There might be another way to handle sequences for wiser heads to figure out. Using `maven torque:jdbc` with Postgresql 7.4.1-jdbc3, my SERIAL columns were converted to XML as follows

```
      <table name="company">
       <column default="nextval('public.company_id_seq'::text)"
           name="id" primaryKey="true" required="true" type="INTEGER"/>
       ...
```

While there are lots of problems with that snippet (such as the fact that the java produced by subsequent the `maven torque:om` goal is buggy), perhaps this could be recognized as calling the Postgresql internal function nextval Postgresql--Sequence-Manipulation Functions, and somehow link that with the ability to define such calls in java, as in Postgresql--Calling Stored Functions. Grepping the source tree, I saw calls to nextval in the Oracle driver, but I didn't explore further. Perhaps this code is similar?

When I examine the source code I see that the PostgreSQL currval() function is used for idMethod='native' - this cannot possibly be right since it should quite obviously be using nextval().

Please read the following thread in its entirety:

- http://nagoya.apache.org/eyebrowse/ReadMsg?listName=torque-dev@db.apache.org&msgNo=3537

The source also gives the impression that it is somehow using AUTOINCREMENT when SEQUENCE would seem to be the more obvious choice. This is more of an indication of whether or not the id should be returned before or after the insert - the names of these things could certainly be improved.

Do I need to patch the PostgreSQL driver to work with Torque?

You do not need to apply any patches to the Postgres driver in order to use Torque **unless** you want to use "large objects" (CLOBS/BLOBS).

The "patch issue" dates from the time when Torque was tightly coupled with Turbine. It arose primarily because Turbine stored a hashmap of user information directly in the database. This data was stored as a BLOB.
Now that Torque is decoupled from Turbine, this issue is only relevant if you use BLOBs into your schema (ie, use the datatypes BLOB or CLOB or LONGVARBINARY or LONGVARCHAR).

So what if I am using BLOBS, etc. ?

1. You will need to patch the driver.
2. The JDBC "BLOB" type should map to the PostgreSQL "oid" column type, but the JDBC driver does not support this automatically. The JDBC driver returns metadata that indicates that "oid" columns have JDBC type Types.INTEGER (the type of the oid ID pointer itself), so Torque (actually, the underlying Village API) doesn't treat the column properly. This can be fixed by patching the JDBC driver, as mentioned in the Postgre SQL Howto.
3. Also, for PostgreSQL 7.2 and higher, it is necessary to specify the parameter "compatible=7.1" on the JDBC URL for `PreparedStatement.setBytes()` to work correctly on oid columns. Note that this will also prevent the VARBINARY (bytea) types from working. Although BLOBs and best for large binary content, the LONGVARBINARY (bytea) types seem to be more portable between databases and JDBC drivers.

This http://db.apache.org/torque/postgres-howto.html seems out of date since the Field.java file has none of the code indicated. A recent, or better fix is where? --rnh

What other issues are there?

- Three patch issues in Scarab (quoting various values) TRQS109TRQS110TRQS111 - that these patches need to be applied in order to use PostgreSQL not totally clear (existing test cases all pass). (If someone can confirm that these are required, or better still provide test cases, I will commit them. – ScottEade 2003-02-05)
    - These three patches seek to introduce the use of quoted identifiers in PostgreSQL queries. While this would allow the use of all characters (other than double quotes) in identifiers, it would also make the identifiers case sensitive which may be quite undesirable. – ScottEade 2004-08-20
- Bill Schneider mentioned on torque-user that "serial" columns are by defaul defined to be of type int4 and that this could pose a problem (the comment may have been made prior to Torque switching from BigDecimal to int for object keys). Note that int4 corresponds to Java int, so this will only be an issue if you need to support a larger range of values than int provides.
- Foreign keys to TURBINE_USER. If you have extended your schema to include foreign key references to TURBINE_USER (by way of an alias table definition) the SQL generated to create your database will include foreign key references to the non-existant alias table. This would not have been a problem under MySQL as it ignores foreign ket definitions, however you may need to change your schema in order to be able to use it with PostgreSQL. Torque needs to use the alias table name rather than the alias when generating the sql.
    - With Turbine 2.3 and the new Torque Security Manager method of extending TurbineUser this is no longer an immediate problem, but it should still be addressed at some stage. – ScottEade 2003-10-09
- Multiple unique column definitions are currently generated with the same constraint name.
    - I have confirmed that this problem is still present in Torque 3.1 and that it applies to index as well as unique. The workaround is to provide the name of the constraint manually using a "name" attribute on the index and unique elements in cases where more than one exists for a given table. – ScottEade 2003-10-09
    - This has been addressed in 3.1.1 and later. – ScottEade 2004-08-20
- Columns declared as type CHAR are returned as space padded Strings by PostgreSQL but are not by MySQL. I'm wondering if Troque shouldn't strip the trailing spaces off these automatically. The workaround is to add a setter to the non-Base Torque class for the object that uses String. trim() to remove the spaces. – ScottEade 2003-11-05
- The jdbc target (torque:jdbc goal in Maven-speak) does not produce the correct DDL SQL for serial columns. – ScottEade 2004-08-20