OracleFAQ

Can I use Torque with Oracle

Yes, several people are using Torque and Oracle together successfully. Oracle Version 8i and up are supported. You do however need to be aware of the issues outlined in the rest of this FAQ.

So what if I am using BLOBS, etc. ?

1. You will need a patched version of Village. The most recent version as of today (December 2005) is village-2.0-dev-oraclepatched-20041022.jar which is available at https://sweb.uky.edu/~skkann2/village/index.html

See also http://db.apache.org/torque/runtime/oracle-howto.html

What should i know when using Oracle's datatype DATE and TIMESTAMP

Oracle datatype DATE supports date and time with an accuracy up to seconds, which is fine for many applications. Since Version 9i Oracle introduced additional datatype TIMESTAMP that supports additional accuracy within fractions of seconds. So far so good, but:

1. The village library used by Torque to read/write data from/into databases truncates dates by setting hours, minutes and seconds all to zero. So although oracle has a better precision in its DATE datatype, you can not use the better precision in Torque.
2. In Version 9.2 of the JDBC-driver Oracle changed the mapping of datatype DATE from java.sql.Timestamp (prior 9.2) to java.sql.Date (since 9.2) leading to a loss of time information.
3. In Version 11.1 of the JDBC-driver, Oracle switched back to mapping datatype DATE to java.sql.Timestamp

See also http://www.oracle.com/technology/tech/java/sqlj_jdbc/htdocs/jdbc_faq.html#08_01

So what if I need exact date and time?

1. When generating your Torque-classes specify all columns as type="TIMESTAMP" in the schema.xml for Torque-Generator, regardless if they are datatype DATE or TIMESTAMP in your Oracle-schema.
2. In your Oracle schema use datatype TIMESTAMP for all columns where you need full date and time. This is recommended when creating a new database schema. You can easily convert DATE-columns to TIMESTAMP as Oracle casts these datatype automatically (UPDATE mytable set newTimestampColumn=oldDateColumn). For existing database (especially in production environment) this may not be feasible:
3. You may still use Oracles datatype DATE. If you want correct time portion of the data you must specify Java-VM parameter "-Doracle.jdbc. V8Compatible=true" that causes the JDBC-driver to use old behaviour of JDBC-driver prior to Oracle 9i (returning full date and time for data type DATE)