

DataBaseGen

DataBaseGen

This tool probably belongs in its own maven sub-project, because although it is useful for the TCK in order to generate DDL for tests, it is a useful end-user tool in its own right.

The tool consists of a Main class that can be run as a command-line tool to create DDL for a set of persistent classes and a class that Main delegates to after processing the command line and system properties. The delegate can be invoked from other application or system components (e.g. the deployer in a J2EE server).

Input to the tool comes from three sources: one or more .jdo files that describe the persistent classes, one or more .class files that contain the byte-codes for the classes, one or more .orm files that describe the mapping, a policy file that governs the generation of DDL specific to a database, and user overrides for generic or specific mappings.

The output is one or more files that contain the DDL to drop and create the schema entries (tables, columns, and constraints).

The command line identifies the classes that will be used to generate DDL and options for the generation.

```
java <java-options> org.apache.jdo.gen.DDLGen <ddlgen-options> <classes>
```

java-options:

-D<key>=<value> provides specific overrides as command line parameters

ddlgen-options:

-h	print usage
?	print usage
-v	print verbose output
-vv	print very verbose output
-s <path>	identifies the classpath to use for loading classes; this is used to create a class loader to load the classes, .jdo metadata, and .orm metadata; must include jdo.jar and jdori.jar unless the system variable \$JDO_HOME is set
-oc <path>	identifies the output file to contain the generated create DDL; default is create.ddl
-od <path>	identifies the output file to contain the generated drop DDL; default is drop.ddl
-p	policy file for DDL generation overrides; default is org.apache.jdo.gen.<database-type>.policy
-q	database-type for default policy; default is SQL-92

classes:

enumeration of classes for which to generate DDL, either in com.acme.Employee or com/acme/Employee.class format if no classes are listed, all classes contained in the META-INF/package.jdo and WEB-INF/package.jdo are assumed

example:

This command is issued where the working directory contains the hr.jar and into which generates hr-create.ddl, and hr-drop.ddl.

```
java org.apache.jdo.gen.DDLGen -vv -s hr.jar -oc hr-create.ddl -od hr-drop.ddl -p Oracle.policy -q oracle
```