MyBatis Generator简单介绍

# MyBatis Generator简单介绍--Introduction to MyBatis Generator

MBG-MyBatis Generator：

MyBatis Generator (MBG) is **a code generator** for MyBatis.( [MyBatis](http://mybatis.org/) and [iBATIS](http://ibatis.apache.org/)).

It will **generate code** for **all versions of MyBatis**, and versions of iBATIS after version 2.2.0. It will **introspect** a database table (or many tables) and will generate artifacts that can be used to access the table(s). This **lessens(减轻)** the initial **nuisance(麻烦事，讨厌的东西)** of setting up objects and configuration files to **interact with** database tables. MBG seeks to make a major impact on the large percentage of database operations that are simple **CRUD (Create, Retrieve, Update, Delete)**. You will still need to hand code SQL and objects for join queries, or **stored procedures**.

# MyBatis Generator will generate:

## Java POJOs that match the table structure. This may include:

* + a class **to match the primary key** of the table (if there is a primary key)
  + a class **to match the non-primary key fields** of the table (except BLOB fields)
  + a class **to include the BLOB fields** of a table (if the table has BLOB fields)
  + a class **to enable dynamic selects, updates, and deletes**

There is an **inheritance relationship** between these classes as appropriate. Note that the generator may be configured to generate different types of POJO hierarchies - for example, you may choose to generate **a single domain object** for each table if you so desire.

## MyBatis/iBATIS Compatible SQL Map XML Files. MBG generates SQL for simple CRUD functions on each table in a configuration. The generated SQL statements include:

* insert
* update by **primary key**
* update by example (using a dynamic **where clause**)
* delete by **primary key**
* delete by example (using a dynamic where clause)
* select by **primary key**
* select by example (using a dynamic where clause)
* count by example

There are different variations of these statements depending on the structure of the table (for example, if the table doesn't have a primary key, then MBG will not generate an update by primary key function).

## Java client classes that make appropriate use of the above objects. The generation of Java client classes is optional. MBG will generate Java clients of the following type for MyBatis 3.x:

* **A mapper interface** that works with the MyBatis 3.x mapper infrastructure

MBG will generate Java clients of the following types for iBATIS 2.x:

* **DAOs** that conform to(符合) the [Spring](http://www.springframework.org/) framework
* **DAOs** that only use the iBATIS SQL mapping API. These DAOs can be generated in two varieties: supplying the SqlMapClient through either constructor or setter injection.
* **DAOs** that conform to the iBATIS DAO Framework (an optional part of iBATIS, this framework is now deprecated and we suggest that you use the Spring framework instead)

**MyBatis generator** is designed to run well in an iterative development environment, and can be included as **an Ant task**, or a Maven plugin, in a continuous build environment. Important things to note when running MBG iteratively include:

1. MBG will automatically merge **XML files** if there is an existing file with the same name as the newly generated XML file. MBG will not overwrite any custom changes you make to the XML files it generates. You can run it over and over again without fear of losing custom changes to you XML. MBG will replace any XML elements that were generated in a previous run.
2. MBG will **not merge Java files**, it can either overwrite existing files or save newly generated files with a different unique name. If you make changes to the generated Java files and run MBG iteratively you will have to merge the changes by hand. When run as an [Eclipse](http://www.eclipse.org/) plugin, then MBG can automatically merge Java files.

# Running with Eclipse

When running as an Eclipse feature the generator can also merge Java files and save user modifications to the generated Java files. The generator uses the Eclipse Java parser and AST walker to accomplish this. The Eclipse feature also has a few user interface enhancements that make the generator somewhat easier to run. Lastly, the Eclipse feature contributes a full user manual for the generator to the Eclipse help system.

The Eclipse feature can be found on the Eclipse marketplace here: <https://marketplace.eclipse.org/content/mybatis-generator> .

# Dependencies

**MBG has no dependencies beyond the JRE**. JRE 6.0 or above is required. Also, a JDBC driver that implements the DatabaseMetaData interface, especially the getColumns and getPrimaryKeys methods is required.

# Support

Support for **MyBatis Generator** is provided through the MyBatis user mailing list. You may subscribe to or view the mailing list through Google code here:

<http://groups.google.com/group/mybatis-user>

If you think you have found a bug, please ask a question about it on the user list first, before creating a new ticket. If you find a bug, or have a new feature request, you may open a new issue at GitHub here:

<https://github.com/mybatis/generator/issues>