

MyBatis Generator Generated Java Model Classes

模型类 (Bean)

与“数据库表字段”一一对应

MyBatis Generator (MBG) generates Java classes that correspond to the fields in a database table. The generated classes are a type of domain object, but should in no way be confused with a rich domain model (see the [Philosophy](#) page for more on this subject). MBG generates different types of "domain" objects based on the characteristics of the table and configuration options. 基于“表特性和配置选项”生成不同类型的模型对象

Every field and method generated by MBG includes the non-standard JavaDoc tag `@mbggenerated`. When run from the Eclipse plugin, on subsequent runs of MBG every Java element that includes this JavaDoc tag will be deleted and replaced. Any other Java element in the class will be untouched by MBG. With this in mind, you can add other fields and methods to the classes without fear of losing them in subsequent runs of MBG - simply DO NOT include the `@mbggenerated` JavaDoc tag on anything that you add to the class.

Outside of the Eclipse plugin, Java files need to be merged by hand, but you can use the `@mbggenerated` JavaDoc tag to know what is safe to delete from a prior version of a file.

The following sections describe the different types of "domain" classes that will be generated. MBG will generate different types of domain classes depending on the value of the `defaultModelType` attribute of the `<context>` configuration element and the `modelType` attribute of the `<table>` configuration element.

Any column ignored by an `<ignoreColumn>` configuration element will be ignored and not added to any generated Java class.

Note: in the following descriptions, the term "BLOB" is used to refer to any column with a data type of BLOB, CLOB, LONGVARCHAR, or LONGVARBINARY.

Primary Key Class

主键类

本类会包含表的主键中的每个字段，属性名基于表中的列名称

This class will contain properties for each field in the primary key of a table. The property names will be generated automatically, and will be based on the column name in the table. The generated property names can be overridden with a `<columnOverride>` configuration element.

类名

The name of the class will be `TableNameKey` by default, or `domainObjectNameKey` if the `domainObjectName` attribute is specified on the `<table>` configuration element.

This class will be generated in the hierarchical model if the table has a primary key. This class will be generated in the conditional model if the table has more than one column in the primary key. This class will not be generated in the flat model.

Record Class

This class will contain properties for each non-BLOB and non-primary key column in the table. The class will extend the primary key class if there is one. The property names will be generated automatically, and will be based on the column name in the table. The generated property names can be overridden with a `<columnOverride>` configuration element.

The name of the class will be `TableName` by default, or `domainObjectName` if the `domainObjectName` attribute is specified on the `<table>` configuration element.

This class will be generated in the hierarchical model if the table has non-BLOB and non-primary key columns. This class will be generated in the conditional model if the table has non-BLOB and non-primary key columns, or if there is only one primary key column or one BLOB column. This class is always generated in the flat model.

Record With BLOBs Class

This class will contain properties for each BLOB column in the table. The class will extend the record class, if there is one, or it will extend the primary key class (note that MBG does not support tables that only contain BLOB columns). The property names will be generated automatically, and will be based on the column name in the table. The generated property names can be overridden with a `<columnOverride>` configuration element.

This class will be the return value from the `selectByPrimaryKey` method, or the `selectByExampleWithBLOBs` method.

The name of the class will be `TableNameWithBLOBs` by default, or `domainObjectNameWithBLOBs` if the `domainObjectName` attribute is specified on the `<table>` configuration element.

This class will be generated in the hierarchical model if the table has any BLOB columns. This class will be generated in the conditional model if the table has more than one BLOB column. This class will not be generated in the flat model.

Example Class

示例类 本类用于动态选择能力

This class is used to work with MBG's dynamic select capability. The class holds a set of criteria that are used to generate a dynamic WHERE clause at runtime for the following methods:

- `selectByExample`
- `selectByExampleWithBLOBs`
- `deleteByExample`
- `countByExample`
- `updateByExample`

This class does not extend any of the other model classes.

The name of the class will be `TableNameExample` by default, or `domainObjectNameExample` if the `domainObjectName` attribute is specified on the `<table>` configuration element.

This class will be generated if any of the `*ByExample` methods are enabled. Note that this class can grow quite large if there are many fields in a table, but the DAO methods are small as is the generated XML fragment. If you do not plan to use the dynamic WHERE clause features, you may prefer to disable the generation of these methods.

See the [Example Class Usage](#) page for details on using the example class.