Overview (/documentation/maven) migrate (/documentation/maven/migrate) clean (/documentation/maven/clean) info (/documentation/maven/info)

validate (/documentation/maven/validate) undo (/documentation/maven/undo) baseline (/documentation/maven/baseline) repair (/documentation/maven/repair)

Maven Plugin

The Flyway Open-Source Maven plugin supports **Maven 3.x** running on **Java 8** or **Java 9**. The Flyway Enterprise Maven plugin also supports **Java 6** and **Java 7**.

Installation

Community Edition **Pro Edition** **Enterprise Edition**

pom.xml

```
<plugin>
    <groupId>org.flywaydb</groupId>
    <artifactId>flyway-maven-plugin</artifactId>
    <version>5.0.7</version>
</plugin>
```

Goals

Name	**Description**
migrate (/documentation/maven/migrate)	Migrates the database
clean (/documentation/maven/clean)	Drops all objects in the configured schemas
info (/documentation/maven/info)	Prints the details and status information about all the migrations
validate (/documentation/maven/validate)	Validates the applied migrations against the ones available on the classpath
undo (/documentation/maven/undo) **Flyway Pro (/download)**	Undoes the most recently applied versioned migration
baseline (/documentation/maven/baseline)	Baselines an existing database, excluding all migrations up to and including baselineVersion
repair (/documentation/maven/repair)	Repairs the schema history table

Configuration

The Flyway Maven plugin can be configured in a wide variety of following ways, which can all be combined at will.

Configuration section of the plugin

The easiest way is to simply use the plugin's configuration section in your `pom.xml` :

```
<plugin>
    ...
    <configuration>
        <user>myUser</user>
        <password>mySecretPwd</password>
        <schemas>
            <schema>schema1</schema>
            <schema>schema2</schema>
            <schema>schema3</schema>
        </schemas>
        <placeholders>
            <keyABC>valueXYZ</keyABC>
            <otherplaceholder>value123</otherplaceholder>
        </placeholders>
    </configuration>
</plugin>
```

Limitation: Due to a long standing Maven bug (http://mail-archives.apache.org/mod_mbox/maven-users/200708.mbox/%3C5a2cf1f60708090246l216f156esf46cc1e968b37ccd@mail.gmail.com%3E) it is **not possible to configure empty values** this way. You must use one of the other configurations ways instead.

Maven properties

To make it easy to work with Maven profiles and to logically group configuration, the Flyway Maven plugin also supports Maven properties:

```
<project>
    ...
    <properties>
        <!-- Properties are prefixed with flyway. -->
        <flyway.user>myUser</flyway.user>
        <flyway.password>mySecretPwd</flyway.password>

        <!-- List are defined as comma-separated values -->
        <flyway.schemas>schema1,schema2,schema3</flyway.schemas>

        <!-- Individual placeholders are prefixed by flyway.placeholders. -->
        <flyway.placeholders.keyABC>valueXYZ</flyway.placeholders.keyABC>
        <flyway.placeholders.otherplaceholder>value123</flyway.placeholders.otherplaceholder>
    </properties>
    ...
</project>
```

settings.xml

For storing the database user and password, Maven `settings.xml` files can also be used:</p>

```
<settings>
    <servers>
        <server>
            <!-- By default Flyway will look for the server with the id 'flyway-db' -->
            <!-- This can be customized by configuring the 'serverId' property -->
            <id>flyway-db</id>
            <username>myUser</username>
            <password>mySecretPwd</password>
        </server>
    </servers>
</settings>
```

Both regular and encrypted settings files are supported.

Environment Variables

To make it ease to work with cloud and containerized environments, Flyway also supports configuration via environment variables (/documentation/envvars). Check out the Flyway environment variable reference (/documentation/envvars) for details.

System properties

Configuration can also be supplied directly via the command-line using JVM system properties:

```
> mvn -Dflyway.user=myUser -Dflyway.schemas=schema1,schema2 -Dflyway.placeholders.keyABC=valueXYZ
```

Config files

Config files (/documentation/configfiles) are supported by the Flyway Maven plugin. If you are not familiar with them, check out the Flyway config file structure and settings reference (/documentation/configfiles) first.

Flyway will search for and automatically load the `<user-home>/flyway.conf` config file if present.

It is also possible to point Flyway at one or more additional config files. This is achieved by supplying the System property `flyway.configFiles` as follows:

```
> mvn -Dflyway.configFiles=path/to/myAlternativeConfig.conf flyway:migrate
```

To pass in multiple files, separate their names with commas:

```
> mvn -Dflyway.configFiles=path/to/myAlternativeConfig.conf,other.conf flyway:migrate
```

Relative paths are relative to the directory containing your `pom.xml` file.

Alternatively you can also use the `FLYWAY_CONFIG_FILES` environment variable for this. When set it will take preference over the command-line parameter.

```
> export FLYWAY_CONFIG_FILES=path/to/myAlternativeConfig.conf,other.conf
```

By default Flyway loads configuration files using UTF-8. To use an alternative encoding, pass the system property `flyway.configFileEncoding` as follows:

```
> mvn -Dflyway.configFileEncoding=ISO-8859-1 flyway:migrate
```

This is also possible via the configuration section of the plugin or Maven properties, as described above.

Alternatively you can also use the `FLYWAY_CONFIG_FILE_ENCODING` environment variable for this. When set it will take preference over the command-line parameter.

```
> export FLYWAY_CONFIG_FILE_ENCODING=ISO-8859-1
```

Overriding order

The Flyway Maven plugin has been carefully designed to load and override configuration in a sensible order.

Settings are loaded in the following order (higher items in the list take precedence over lower ones):

1. System properties
2. Environment variables
3. Custom config files
4. Maven properties
5. Plugin configuration section
6. Credentials from `settings.xml`
7. `<user-home>/flyway.conf`
8. Flyway Maven plugin defaults

The means that if for example `flyway.url` is both present in a config file and passed as `-Dflyway.url=` from the command-line, the JVM system property passed in via the command-line will take precedence and be used.

Maven: migrate ➔ (/documentation/maven/migrate)