## If you have a question or need any help...

Please use [the mailing list](http://groups.google.com/group/mybatis-user) instead of creating issues on the tracker. Thank you!

## Reporting a bug

- Create a new issue on [the tracker](https://github.com/mybatis/generator/issues).

- The best way to report a bug is to create a failing test case. Please see the [Contributing code](CONTRIBUTING.md#contributing-code) section.

## Proposing a new feature

- It is a good idea to discuss your changes on [the mailing list](http://groups.google.com/group/mybatis-user) to get feedback from the community.

- If you have a patch with unit tests, send a pull request. Please see the [Contributing code](CONTRIBUTING.md#contributing-code) section.

- MyBatis Generator has a very powerful plugin mechanism that can be used to alter many aspects of the code generation process. It is our preference

to limit new features in the base product to features that cannot be implemented easily with a plugin. In other words, please do not send pull

requests or make feature requests for items that can be implemented with a plugin.

- We also prefer to limit the number of plugins supplied with the generator and added to the code base. In most cases, plugins are very

specific to an individual use case. If you feel that your plugin has wide potential use, feel free to promote it on the mailing list

and you can add a description of your plugin on the Generator WIKI (https://github.com/mybatis/generator/wiki/Third-Party-Tools)

## Improving documentation

- Documentation is placed under [src/site](https://github.com/mybatis/generator/tree/master/core/mybatis-generator-core/src/site) directory, so

it is basically the same as creating a patch to contribute documentation changes. Please see the [Contributing code](CONTRIBUTING.md#contributing-code) section.

## Contributing code

### General Information

MyBatis generator has very high code coverage and thousands of unit tests. The generator also has extensive documentation that covers

virtually every aspect of using it. We aim to keep it that way. Pull requests that contain unit tests and documentation are much more likely

to be accepted. If you send a pull request without documentation or unit tests, then know that the team will likely not accept that request

unless and until someone else writes the documentation and/or tests.

Testing a code generator is very complex. The tests for the code generator are primarily in the sub-projects like

mybatis-generator-systests-mybatis3 (https://github.com/mybatis/generator/tree/master/core/mybatis-generator-systests-mybatis3). That project

runs the generator to generate code on a sample database, and then executes unit tests on the generated code. If you are unsure of how to add

a unit test, please feel free to ask on the mailing list.

### Copyright and License

- You are the author of your contributions and will always be.

- Everything you can find it this project is licensed under the Apache Software License 2.0

- Every contribution you do must be licensed under the Apache Software License 2.0. Otherwise we will not be able to accept it.

- Please make sure that all the new files you create hold the following header:

```

/\*

\* Copyright [year] the original author or authors.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* http://www.apache.org/licenses/LICENSE-2.0

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

```

### How to send your modifications as a pull request

The best way to submit a patch is to send a pull request.

Here are the steps of a typical workflow.

1. Fork the repository on GitHub.

2. Clone your fork to your local machine.

3. Create a topic branch with a descriptive name.

4. Make changes with unit tests in the topic branch.

5. Push commits to your fork on GitHub.

6. Send a [pull request](https://help.github.com/articles/using-pull-requests).

For steps 1 to 3, please read [this GitHub help](https://help.github.com/articles/fork-a-repo) if you are not familiar with these operations.

Step 4 and 5 are basic [git](http://git-scm.com/) operations. Please see the [online documentation](http://git-scm.com/documentation) for its usage.