

# Code style for KTH MDA & Eclipse Lyo projects

Andrew Berezovskyi

## Motivation

Establish easy to follow rules that will make the code look consistent, clean, and understandable. This document is based on previous experience with code, styled in different ways.

This document sets a foundation for future code quality guidelines, such as static checks, code coverage by unit tests etc.

## Test code style

Prefer as little assertions per test method.

Try to always organize the test method according to the Arrange-Act-Assert principle.

Name unit test classes as `ClassTest`.

Name integration test classes as `ClassIT`. Use *maven-failsafe* plugin to run them at the *verify* stage.

Name test methods as `testObjectActionExpectation()`, like `testConnectionAcceptsRequests()`.

## Formatting rules

Formatting rules are not to be enforced strictly in the coming future, but the code that is being contributed has to be formatted consistently.

Recommended configuration file for Eclipse can be downloaded [here](#). In order to import its rules, open Eclipse Preferences and navigate to *Java > Code Style > Formatter* and use Import to open the file:

## Applying the rules

In order to manually format a file in Eclipse, press `Ctrl+Shift+F` on Windows or Linux and `Cmd+Shift+F` on Mac.

In order to apply formatting to the code you have changed since last commit automatically, enable automatic formatting under *Java > Editor > Save Actions*:

Automatic formatting is not removing trailing spaces reliably (especially in non-Java files). In order to overcome this problem, install AnyEdit plugin from the Eclipse Marketplace and configure it to remove all trailing whitespace and enforce spaces.

Settings for XML are located under *XML > XML Files > Editor*.

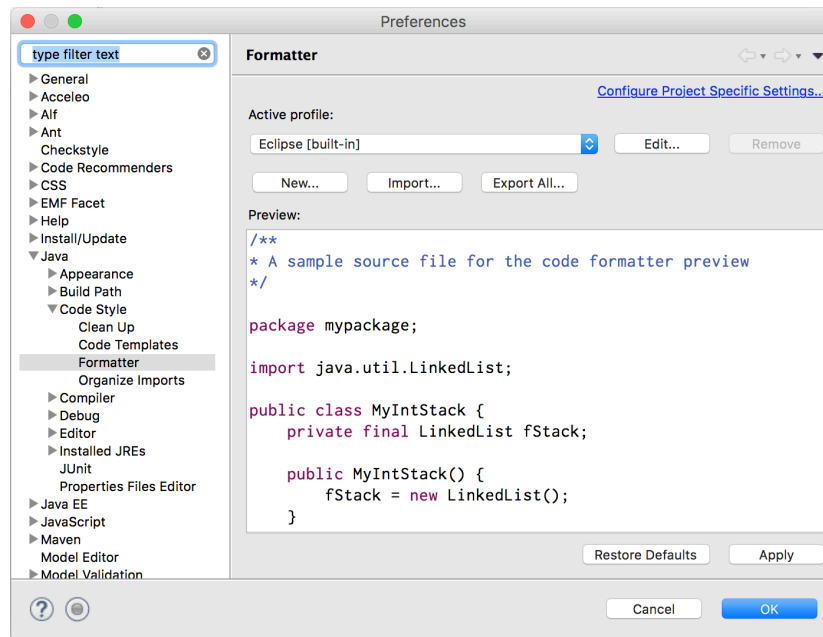


Figure 1: Eclipse Preferences dialog for code formatting

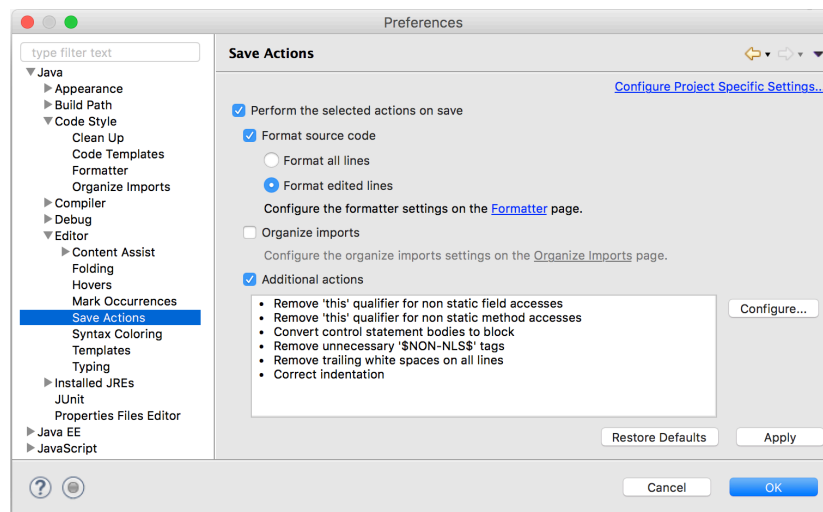


Figure 2: Eclipse Preferences dialog for formatting on save

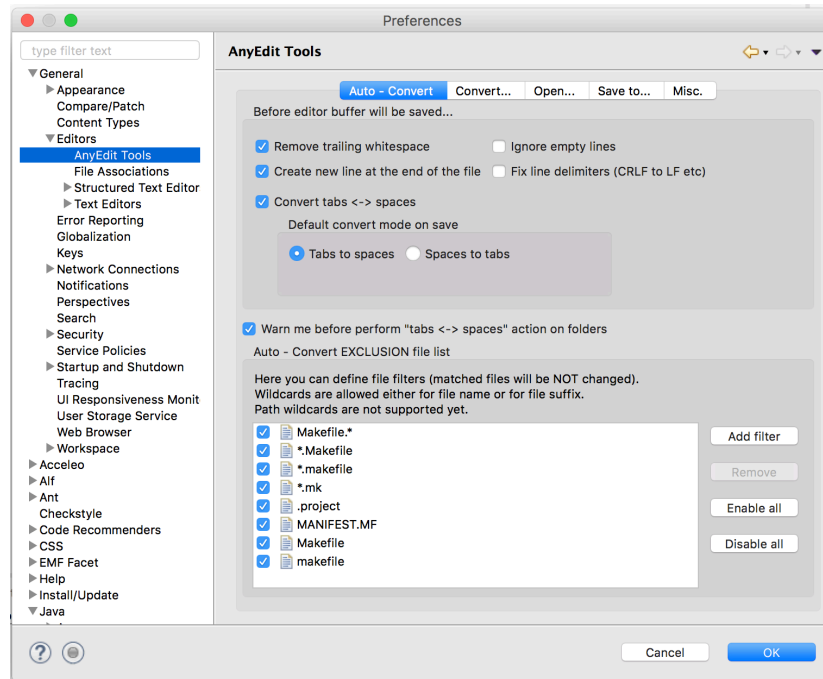


Figure 3: Eclipse Preferences dialog for AnyEdit

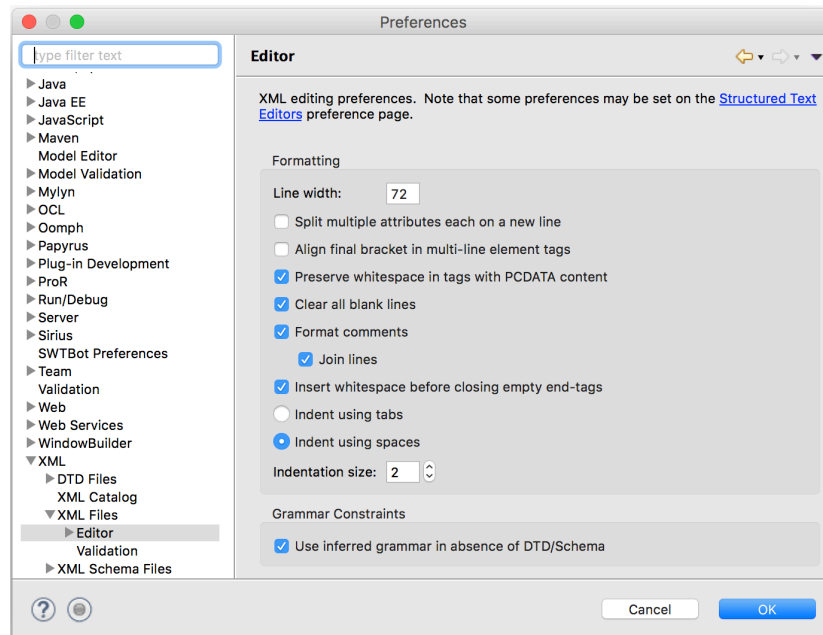


Figure 4: Eclipse Preferences dialog for XML

Settings for JSP are by default controlled by HTML settings under *Web > HTML Files > Editor*.

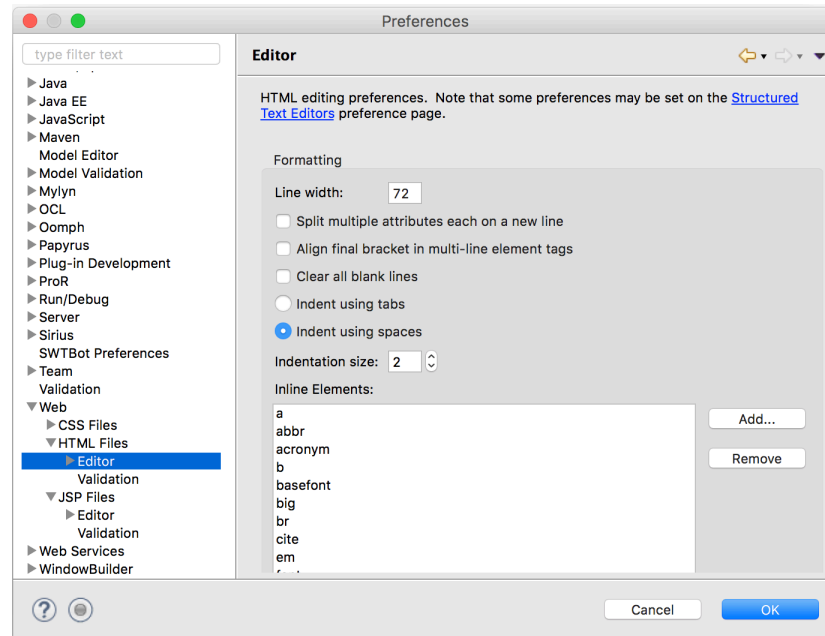


Figure 5: Eclipse Preferences dialog for HTML

## Space delimiter motivation

Project code is viewed in different tools, such as IDEs, IDE and online diff tools, text-based documentation like this one.

Different programming languages have different preferred delimiter sizes and many tools do not allow setting a delimiter size based on the file type, use the default of 8 spaces, and are not intuitive to configure.

Therefore, spaces are preferred over tabs in order to maintain compatibility over many tools.

Tab delimiter sizes:

- JavaScript, HTML, CSS, XML: 2 spaces.
- Java and other files: 4 spaces.

## State of the art

For further improvement of the code quality, look into the following resources:

- [Google Java Style Guide](#)
- [Checkstyle](#) for enforcing the aforementioned style
- Cobertura for testing code coverage
- [Qulice](#) for using multiple static analysis tools
- SonarQube for continuous monitoring of the code quality
- [Working Effectively with Legacy Code](#)
- [Refactoring: Improving the Design of Existing Code](#)
- [Clean Code: A Handbook of Agile Software Craftsmanship](#)