

We show you how to set the Eclipse formatting actions to perform when a file is saved.

Save Actions

Open up the Eclipse Preferences and select

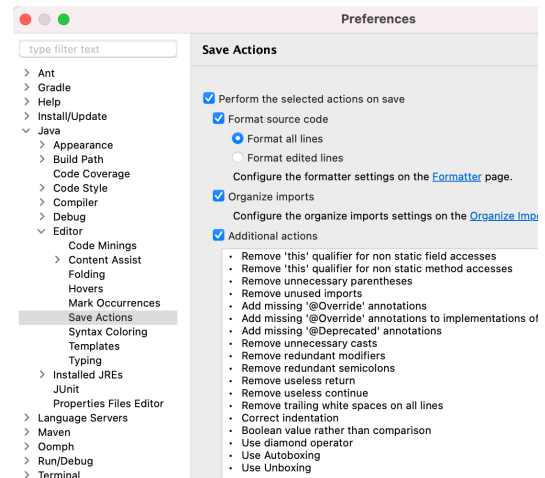
Java -> Editor -> Save Actions

The window should look as it does to the right.

First, make sure that all the boxes are checked as in this image.

Second, look at all the *Additional actions*. If these 18 actions are present, then there is no need to do anything. If they are not all there, then please follow the directions below.

You will be making a bunch of changes. If you want, you can click OK at any time to save what you did so far and then get back in and continue.



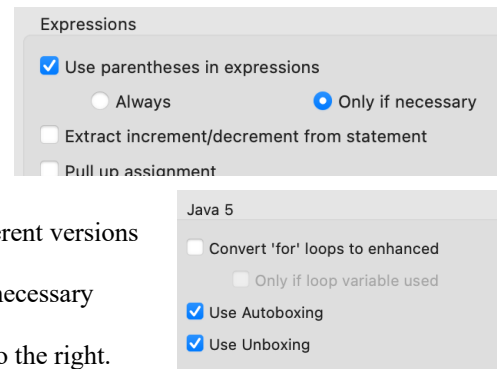
Configuring

To the right of the *Additional actions* field, you will see a button *Configure*. Press it, and a window title “Additional Save Actions” will appear (shown). It has 9 buttons in the horizontal bar. We show some of them to the right.

Code Organizing. Button *Code Organizing* has been selected. If you don't see that one selected, select it. Then make sure that three things are selected as shown: Remove trailing white space, All lines, and Correct indentation.



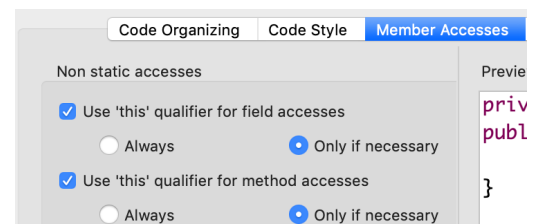
Code Style. Click button *Code Style*. You will see partitions for *Control statements*, *Expressions*, and three others. Select only items in the partition for *Expressions*, as shown to the right. Select “Use parentheses in expressions,” and also “Only if necessary”.



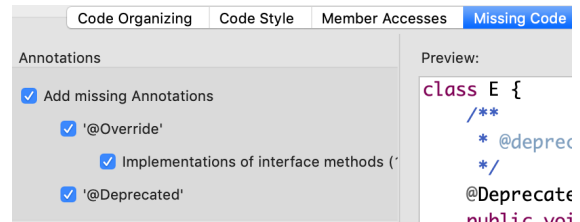
Java Feature. Click button *Java Feature*. You will see items for different versions of Java. We show only the one for Java 5 to the right. For Java 7, select "Use diamond operator". This is used to remove unnecessary types.

For Java 5, select "Use Autoboxing" and "Use Unboxing", as shown to the right.

Member Accesses. Click button *Member Accesses*. In the pane that opens, change the partition for *Non static accesses* as shown to the right. Thus, “this” will be used for field and method accesses only if necessary. There is no need to check anything in the partition for *Static accesses*.

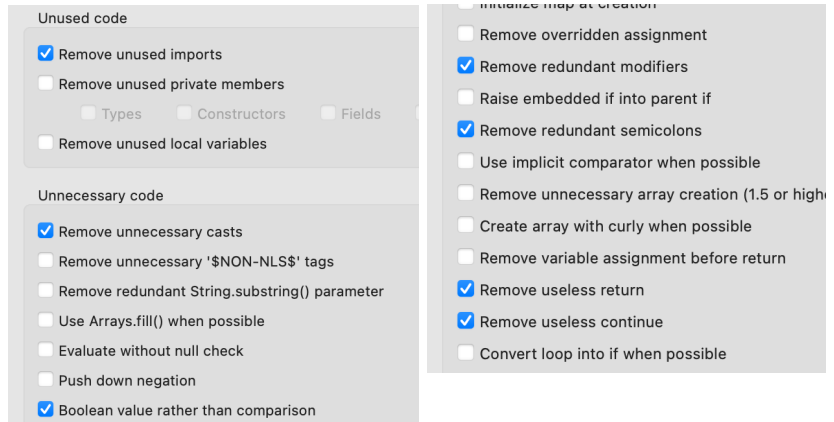


Missing Code. Click button *Missing Code*. Make sure the all four check boxes are checked, as shown to the right. You will soon learn what annotation `@Override` is for. To *deprecate* means to lessen in value. Some Java classes and methods within classes in the API package have been deprecated because they have been superseded by newer one that do a better job. You can still use the deprecated entities, but it is better to use their newer replacements.



Unnecessary Code. Click button *Unnecessary Code*. Click the boxes as shown to the right. Thus, these items will be removed:

- Unnecessary import
- Unnecessary casts
- Boolean value r.t. comparison
- Redundant modifiers
- Redundant semicolons
- Useless return
- Useless continue



The *Boolean value rather than comparison* feature does this:
If you type `isValid == true`, it replaces it with `isValid`.

Finishing up. Now it is time to click the blue *OK* button at the bottom right of the window. This opens up the *Additional Save Actions* window. Click the *Apply and Close* button to save all the changes you made.

That's it!