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

Save Actions

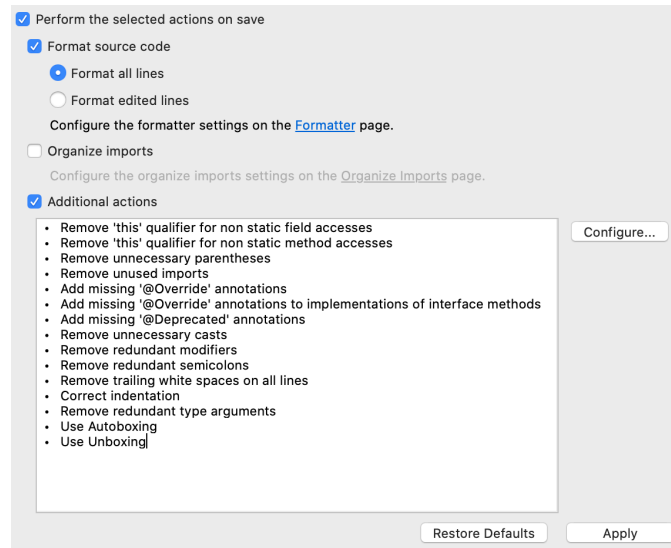Open up the Eclipse Preferences and select

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 15 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.
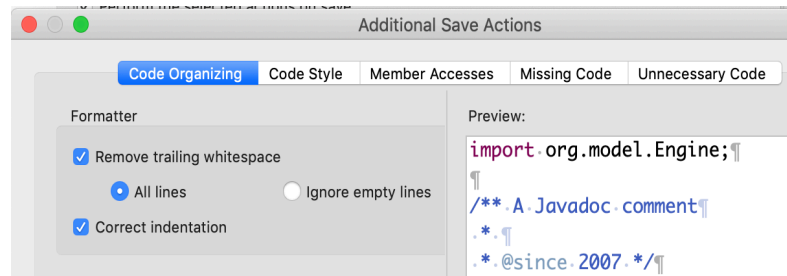
☑ Perform the selected actions on save
☑ Format source code
  ◉ Format all lines
  ○ Format edited lines
  Configure the formatter settings on the <u>Formatter</u> page.
☐ Organize imports
  Configure the organize imports settings on the <u>Organize Imports</u> page.
☑ Additional actions

- Remove 'this' qualifier for non static field accesses
- Remove 'this' qualifier for non static method accesses
- Remove unnecessary parentheses
- Remove unused imports
- Add missing '@Override' annotations
- Add missing '@Override' annotations to implementations of interface methods
- Add missing '@Deprecated' annotations
- Remove unnecessary casts
- Remove redundant modifiers
- Remove redundant semicolons
- Remove trailing white spaces on all lines
- Correct indentation
- Remove redundant type arguments
- Use Autoboxing
- Use Unboxing

[Configure...]

[Restore Defaults] [Apply]

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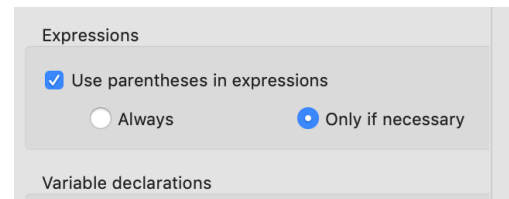
Code Organizing. Button *Code Organizing*, has been selected. If you don't see that one selected, select it. Then make sure that three things are selected as shown:
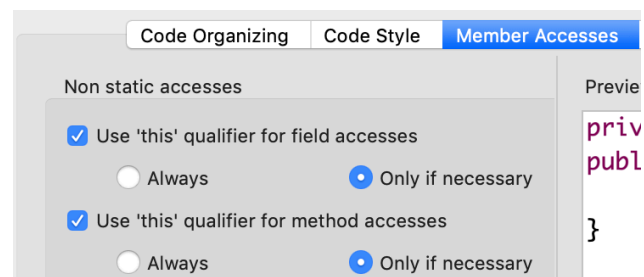Remove trailing white space, All lines, and Correct indentation.

Additional Save Actions

Code Organizing | Code Style | Member Accesses | Missing Code | Unnecessary Code

Formatter
☑ Remove trailing whitespace
  ◉ All lines   ○ Ignore empty lines
☑ Correct indentation

Preview:
```
import org.model.Engine;¶
¶
/** A Javadoc comment¶
 * ¶
 * @since 2007 */¶
```

Code Style. Click the *Code Style* button.
You will see partitions for *Control statements*, *Expressions*, *Variable declarations*, and *Functional interface instances*. Select only items in the partition for Expressions, as shown to the right. Select "Use parentheses in expressions," and also "Only if necessary".

Expressions
☑ Use parentheses in expressions
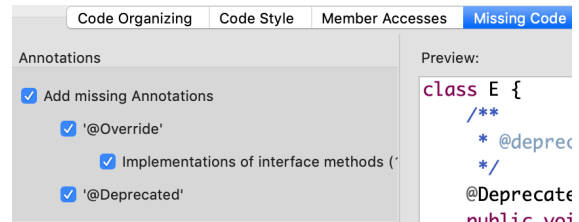 ○ Always   ◉ Only if necessary

Variable declarations

Member Accesses. Click the *Member Accesses* button. 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*.

Code Organizing | Code Style | Member Accesses

Non static accesses
☑ Use 'this' qualifier for field accesses
  ○ Always   ◉ Only if necessary
☑ Use 'this' qualifier for method accesses
  ○ Always   ◉ Only if necessary
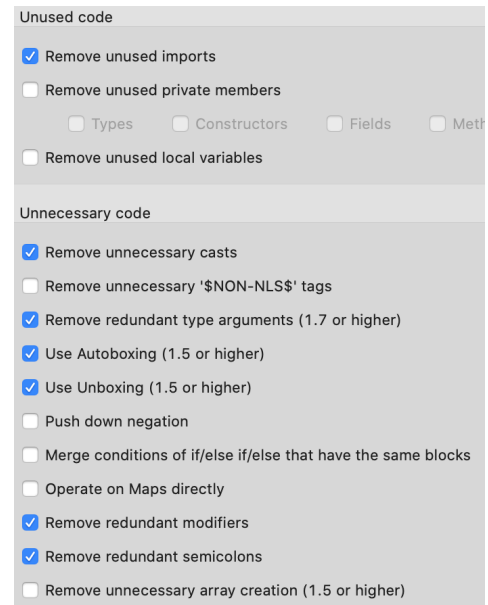
Previe
```
priv
publ

}
```

Missing Code. Click the *Missing Code* button. 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.

Code Organizing   Code Style   Member Accesses   **Missing Code**

Annotations

☑ Add missing Annotations

    ☑ '@Override'

       ☑ Implementations of interface methods (

    ☑ '@Deprecated'

Preview:

```
class E {
    /**
     * @deprec
     */
    @Deprecate
    public voi
```

Unnecessary Code. Click the *Unnecessary Code* button. Click the boxes as shown to the right. Thus, these items will be removed:
       Unnecessary imports
       Unnecessary casts
       Redundant type arguments
       Redundant modifiers
       Redundant semicolons.
And Autoboxing and Unboxing will be used where possible.

Unused code

☑ Remove unused imports

☐ Remove unused private members

    ☐ Types   ☐ Constructors   ☐ Fields   ☐ Meth

☐ Remove unused local variables

Unnecessary code

☑ Remove unnecessary casts

☐ Remove unnecessary '$NON-NLS$' tags

☑ Remove redundant type arguments (1.7 or higher)

☑ Use Autoboxing (1.5 or higher)

☑ Use Unboxing (1.5 or higher)

☐ Push down negation

☐ Merge conditions of if/else if/else that have the same blocks

☐ Operate on Maps directly

☑ Remove redundant modifiers

☑ Remove redundant semicolons

☐ Remove unnecessary array creation (1.5 or higher)

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

That's it!