

[Java development user guide](#) > [Reference](#) > [Refactoring](#)

Refactoring Wizard

A dialog based user interface guides you through the steps necessary to execute a selected refactoring. Depending on the complexity of the refactoring, either a wizard or a simple dialog is used to gather information that is required for the refactoring.

Input pages

The input pages gather information that is required for the refactoring. After you have provided the required input you can click **Ok** or **Finish** to carry out the refactoring without previewing the results. If you want to preview the changes press **Preview** or **Next**

Preview page

The JDT allows you to preview the results of a refactoring action before you execute it.

The preview page consists of two parts:

- A tree at the top containing all Java elements affected by the refactoring. Each top-level node in the tree represents one compilation unit.
Some refactorings allow to filter the tree by different kind of changes made by the refactoring. Use the **Filter Changes** drop down to change the filtering.
- A compare viewer at the bottom. The left side of the compare viewer shows the original, the right side displays the refactored source.

Problem page

The Refactoring Problem page indicates if there are suspected, potential, or definite problems with the refactoring action you are attempting.

Four types of problems are possible:

Information

A problem described as Information will not affect the refactoring in any way, nor will it negatively affect the code in the workbench. You can most likely ignore this type of problem.

Warnings

Warnings attempt to predict compile warnings. This type of problem most likely will not negatively affect the code in your workbench.

Errors

A problem described as an Error is very likely to cause compile errors or change your workbench code semantically. You can choose to continue with the refactoring in spite of these errors, although it is not recommended.

Stop problems

This type of problem prevents the refactoring from taking place. For example, if you select a comment and choose the Extract Method command from it, the workbench will issue a stop problem on the refactoring attempt because you cannot extract a comment.

If there aren't any stop problems then the refactoring can be carried out by pressing the **Finish** button. To preview the results of the refactoring action, press the **Next >** button.

Refactoring without Dialog

It is also possible to rename a Java element without showing a dialog. This can be enabled and disabled on the [Java](#) preference page. If enabled, then the new name for an Java element can be typed into the editor when the rename refactoring is invoked.

Related concepts

[Refactoring support](#)

Related references

[Refactoring actions](#)
[Icons](#)