

Refactoring Source Code

IntelliJ IDEA offers a wide variety of code refactorings, which track down and correct the affected code references automatically.

In this part you will find:

- [General refactoring procedure](#)
- [Procedures and examples of the available refactorings](#)

To perform refactoring, follow these general steps

1. Select a symbol or code fragment to refactor. The set of available refactorings depends on your selection. You can select symbols in the following IntelliJ IDEA components:
 - Project view
 - Structure tool window
 - Editor
 - Commander

The following is available only in the Ultimate edition of IntelliJ IDEA

- UML Class diagram
2. Do one of the following:
 - On the main **Refactor** menu or on the context menu of the selection, choose the desired refactoring or press the corresponding keyboard shortcut (if any).
 - On the main menu, choose **Refactor | Refactor This**, or press **Ctrl+Shift+Alt+T**, and then select the desired refactoring from the pop-up window.
 3. In the dialog box that opens, specify the refactoring options.
 4. To apply the changes immediately, depending on the refactoring type, click **Refactor** or **OK**.
 5. For certain refactorings, there is an option of previewing the changes prior to actually performing the refactoring. In such cases the **Preview** button is available in the corresponding dialog.

To preview the potential changes and make the necessary adjustments, click **Preview**. IntelliJ IDEA displays the changes that are going to be made on a dedicated tab of the [Find tool window](#).

One of the possible actions at this step is to exclude certain entries from the refactoring. To do so, select the desired entry in the list and press **Delete**.

If conflicts are expected after the refactoring, IntelliJ IDEA displays a dialog with a brief description of the encountered problems. If this is the case, do one of the following:

- Ignore the conflicts by clicking the **Continue** button. As a result, the refactoring will be performed, however, this may lead to erroneous results.
 - Preview the conflicts by clicking the **Show in View** button. IntelliJ IDEA shows all conflicting entries on the **Conflicts** tab in the [Find tool window](#), enabling you to navigate to the problematic lines of code and to make the necessary fixes.
 - Cancel the refactoring and return to the editor.
6. When you are satisfied with the proposed results, click **Do Refactor** to apply the changes.

IntelliJ IDEA provides the following common refactorings:

- [Change Class Signature](#)
- [Change Signature](#)

- Convert Anonymous to Inner
- Convert to Instance Method
- Copy
- Encapsulate Fields
- Extract Delegate
- Extract Include File
- Extract Interface
- Extract Method
- Extract Method Object
- Extract Parameter Object
- Extract Superclass
- Generify Refactoring
- Inline
- Extract Constant
- Extract Field
- Extract Parameter
- Extract Property
- Extract Variable
- Invert Boolean
- Make Class Static
- Make Method Static
- Migrate
- Move Refactorings
- Pull Members up
- Push Members down
- Remove Middleman
- Rename Refactorings
- Replace Constructor with Builder
- Replace Constructor with Factory Method
- Replace Inheritance with Delegation
- Replace Method Code Duplicates
- Replace Temp with Query
- Safe Delete
- Type Migration
- Use Interface Where Possible
- Wrap Return Value

The following language-specific refactorings are available:

- Change Signature in JavaScript
- Extract Parameter in JavaScript
- Extract Variable in JavaScript

See Also

Procedures:

- [Working with Search Results](#)

Reference:

- [Refactoring Dialogs](#)

External Links:

- <http://www.refactoring.com/> 

Web Resources:

- [Developer Community](#) 