

Rename Refactorings

Rename refactorings allow you to rename symbols , automatically correcting all references in the code.

The following rename refactorings are available in IntelliJ IDEA:

- *Rename Package*. The following usages are renamed:
 - Package statements
 - Import statements
 - Qualified names of classes
- *Rename Class*. The following usages are renamed:
 - Import statements
 - Qualified names of classes
 - Variables with the selected class type
 - Class inheritors
- *Rename Method*. The following usages are renamed:
 - All calls of the method.
 - All overridden/implemented methods in subclasses.
- *Rename Field*.
- *Rename Function*.
- *Rename Variable*.
- *Rename Parameter*. The following usages are renamed:
 - All usages of the parameter.
 - The corresponding param tag in documentation comment.
- *Rename CSS color value*.
- *Rename File*.
- *Rename Directory*.
- *Rename views and references to views* in the Grails applications.

Example

Before	After
<pre>public class MyClass { // some code here } ... public void myMethod() { MyClass myClass = new MyClass(); }</pre>	<pre>public class YourClass { // some code here } ... public void myMethod() { YourClass yourClass = new YourClass(); }</pre>

To rename a symbol, follow these general steps

1. Select the item to be renamed.
 - To select a file, click the desired file in the [Project tool window](#).
 - To select a symbol in the editor, place the caret at the name of the symbol to be renamed.
 - To select a symbol in the [Project tool window](#), make sure that the members are shown, and then click the desired symbol.
 - To select a symbol in the [Structure view](#), click the desired symbol in the Structure tool window.
2. Choose **Refactor | Rename** on the main menu or on the context menu of the selection or press **Shift+F6**.
3. The subsequent behavior depends on the check box [Enable in-place mode](#) (**Settings | IDE Settings | Editor**).
 - If this check box is selected, the suggested name appears right below the symbol in question. You can either accept suggestion, or type a new name. However, if you press **Shift+F6** once more, IntelliJ IDEA will display the [Rename](#) dialog box with more options.
 - If this check box is not selected, the [Rename](#) dialog box opens immediately.

The set of controls and their names depend on the type of the symbol to be renamed.

4. If you want IntelliJ IDEA to find and rename objects related to the renamed class, whose names contain the changed string, check one or more of the following options:
 - **Rename variables** to rename the variables of that class type.
 - **Rename inheritors** to rename class inheritors.
 - **Rename bound forms** to rename the GUI forms bound to the class.

If you chose to rename any of the objects bound to the renamed class, IntelliJ IDEA searches for appropriate items and displays them in a sequence of dialogs, sorted by type. In each dialog you may select the items you want to change.

5. [Preview and apply changes](#).

Local variables, by default, are renamed in-place:

```
public CrocoBuilder setPaws(int paws) {  
    int paws1 = paws;  
    return this;  
}
```

To be able to use the **Rename** dialog when renaming local variables, you should [disable in-place refactoring in the editor settings](#).

See Also

Reference:

- [Rename Dialogs](#)
- [Find Tool Window](#)

Web Resources:

- [Developer Community](#) 

