

Extract Constant

The *Extract Constant* refactoring is a special case of the [Extract Field refactoring](#) intended to provide a fast and convenient way to create final static fields.

- [Example](#)
- [Extracting a Java constant in-place](#)
- [Extracting a constant using the dialog box](#)

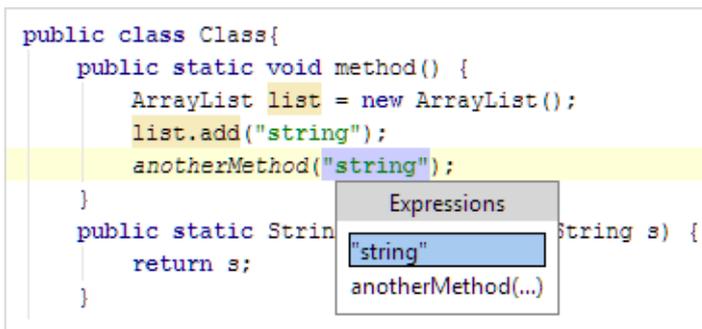
Example

Before	After
<pre>public class Class { public void method() { String string = "string"; ArrayList list = new ArrayList(); list.add(string); anotherMethod(string); ... } }</pre>	<pre>public class Class { @Nonnull private static final String STRING = "string"; public void method() { ArrayList list = new ArrayList(); list.add(STRING); anotherMethod(STRING); ... } }</pre>
<pre>public class Class { public void method() { ArrayList list = new ArrayList(); list.add("string"); anotherMethod("string"); ... } }</pre>	<pre>public class Class { @Nonnull private static final String STRING = "string"; public void method() { ArrayList list = new ArrayList(); list.add(STRING); anotherMethod(STRING); ... } }</pre>

To extract a Java constant in-place

The [in-place refactorings](#) are enabled in IntelliJ IDEA by default. So, if you haven't changed this setting, the Extract Constant refactoring for Java is performed in-place, right in the editor:

1. Place the cursor within the expression or declaration of a variable to be replaced by a constant.
2. Do one of the following:
 - Press **Ctrl+Alt+C**.
 - Choose **Refactor | Extract Constant** in the main menu, or on the context menu.
3. If more than one expression is detected for the current cursor position, the **Expressions** list appears. If this is the case, select the required expression. To do that, click the expression. Alternatively, use the **Up** and **Down** arrow keys to navigate to the expression of interest, and then press **Enter** to select it.

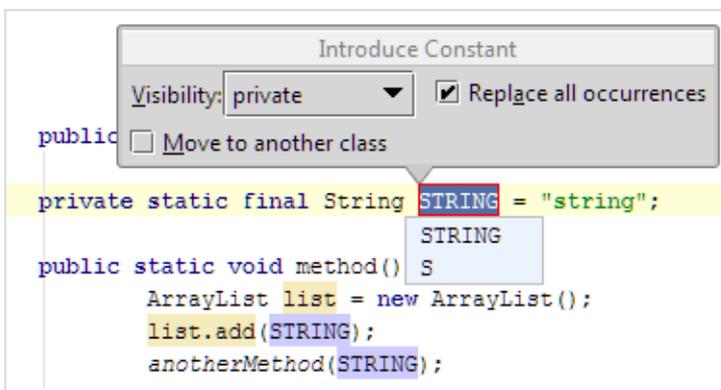


4. If more than one occurrence of the expression is found within the class, specify whether you wish to replace only the selected occurrence, or all the found occurrences with the new constant.
5. If you want the constant to be defined in a different class, select the **Move to another class** check box.
6. If necessary, change the type of the new constant.

To move to the type, press **Shift+Tab**. Then, select the required type from the list, or edit the type in the box with the red border.

Now, to move back to the constant name, press **Tab**.

7. Specify the name of the constant. Select the name from the list or type the name in the box with a red border.



8. To complete the refactoring, press **Tab** or **Enter**.

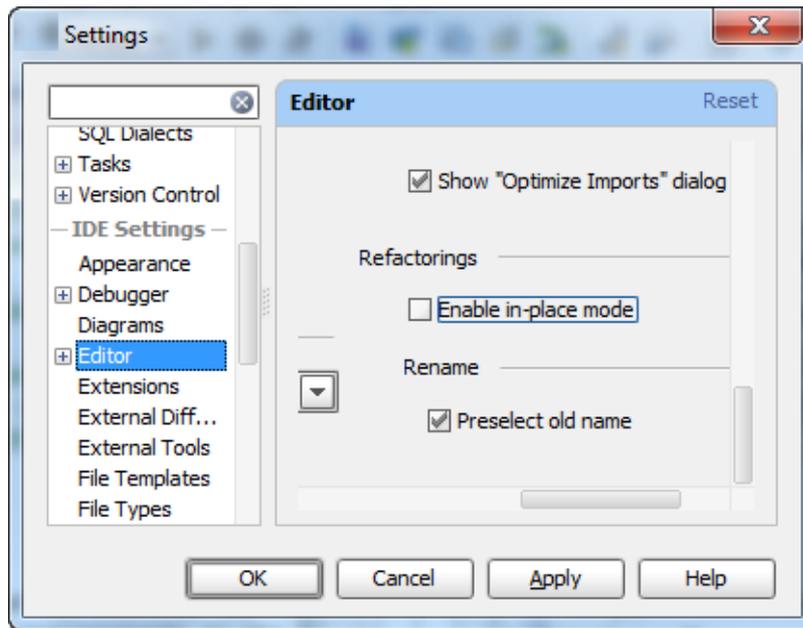
If you haven't completed the refactoring and want to cancel the changes you have made, press **Escape**.

Note that sometimes you may need to press the corresponding key more than once.

9. If you have selected to move the constant definition to a different class, specify the associated settings in the [Move Members dialog](#).

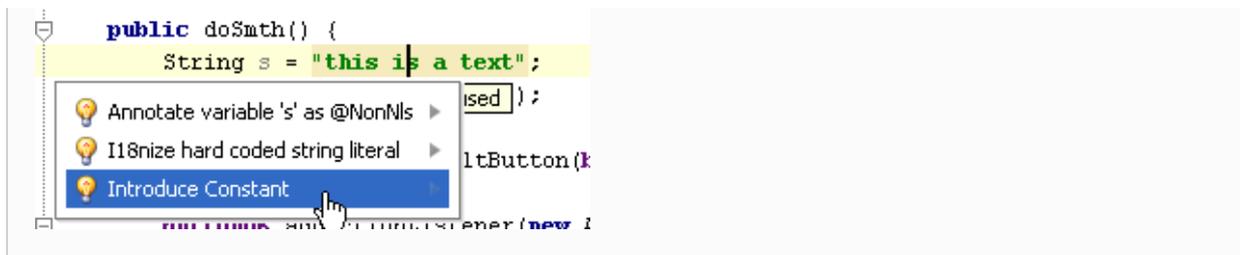
To extract a constant using the dialog box

If the **Enable in place refactorings** check box is cleared on the Editor settings, the **Extract Constant** refactoring is performed by means of the [Extract Constant Dialog](#) dialog box.



1. If you are working with Java code, make sure that the [Enable in place refactorings](#) option is off in the editor settings. (By default, the Extract Constant refactorings for Java are performed [in-place](#).)
2. In the editor, select the expression or variable to be replaced with a constant, or just place the cursor within such an expression or variable declaration.
3. In the main or the context menu, choose **Refactor | Extract Constant**, or press **Ctrl+Alt+C**.
4. In the **Expressions** pop-up menu, select the expression to be replaced. Note that IntelliJ IDEA highlights the selected expression in the editor.
5. In the [Extract Constant Dialog](#) dialog that opens:
 1. Specify the name of the new constant.
 2. Select the class where the constant will be introduced. If you select an enum class here, use the **Introduce as enum constant** option to specify whether the constant should be an enum constant, or a usual field.
 3. In the **Visibility** area, select the visibility scope for the new constant.
 4. If the new constant is going to replace an existing variable, you can choose to delete the corresponding variable declaration. To do that, use the **Delete variable declaration** check box.
 5. To replace all the occurrences of the selected expression (if the selected expression is found more than once in the class), select the **Replace all occurrences** check box.
 6. In the projects configured for using annotations, you can annotate the constant of the String type as `@NonNls` to prevent it from change during possible localizations. To do so, select the option **Annotate field as @NonNls**.
 7. Click **OK**.

You can also invoke the refactoring from the intention action drop-down menu:



See Also

Reference:

- [Extract Constant Dialog](#)

Web Resources:

- [Developer Community](#)