

Extract Field

The *Extract Field* refactoring declares a new field and initializes it with the selected expression. The original expression is replaced with the usage of the field.

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

Example

Before	After
<pre>public class Class { AnotherClass anotherClass; public void method() { int a = 1; ... int b = a + anotherClass.intValue(); int c = b + anotherClass.intValue(); } }</pre>	<pre>public class Class { public AnotherClass anotherClass; private int number; public Class() { number = anotherClass.intValue(); } public void method() { int a = 1; ... int b = a + number; int c = b + number; } }</pre>
<pre>public class Class { AnotherClass anotherClass; public void method() { int a = anotherClass.innerClass.i; int b = anotherClass.innerClass.j; } }</pre>	<pre>public class Class { public AnotherClass anotherClass; private AnotherClass.InnerClass aClass = public void method() { int a = aClass.i; int b = aClass.j; } }</pre>

To extract a field in-place

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

1. Place the cursor within the expression or declaration of a variable to be replaced by a field.
2. Do one of the following:
 - Press **Ctrl+Alt+F**.
 - Choose **Refactor | Introduce Field** 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.

```
public void method() {
    int a = 1;
    int b = a + anotherClass.intValue();
    int c = b + anotherClass.intValue();
    String s = "Hello!" +
}
```

Expressions

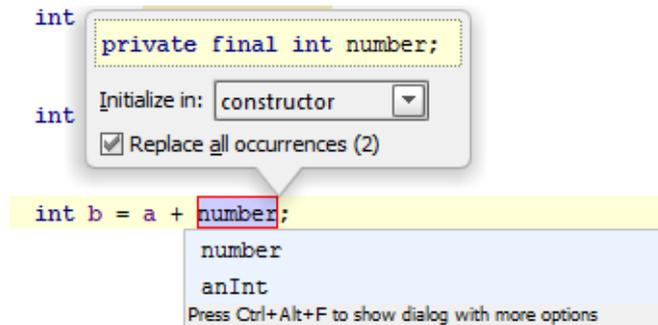
- anotherClass
- anotherClass.intValue()
- a + anotherClass.intValue()

- 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:
- If necessary, change the type of the new field.

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 read border.

Now, to move back to the field name, press Tab.

- If relevant, specify where the new field will be initialized - in the current method, or in a class constructor.
- Specify the name of the field. Select the name from the list or type the name in the box with a red border.



- 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.

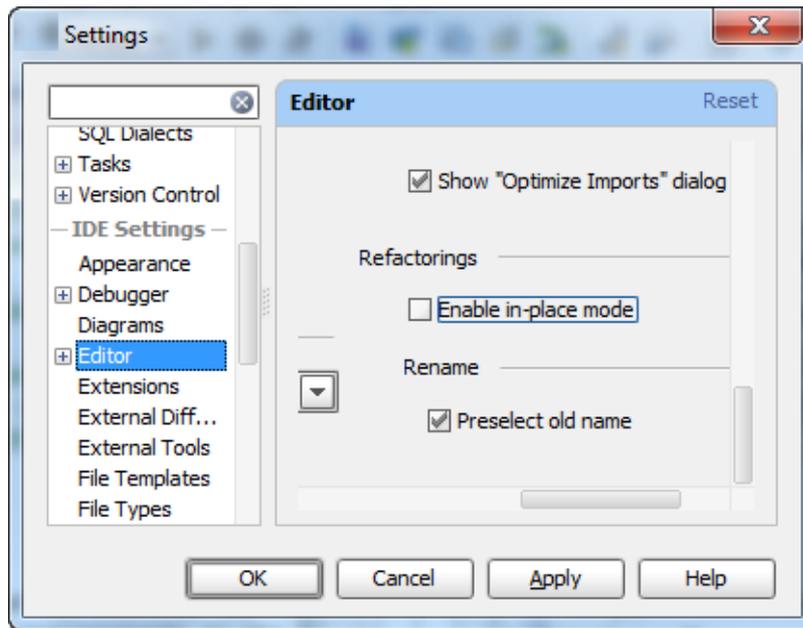
```
public class Class {
    AnotherClass anotherClass;
    private int number;

    public Class() {
        number = anotherClass.intValue();
    }

    public void method() {
        int a = 1;
        int b = a + number;
        int c = b + number;
    }
}
```

To extract a field using the dialog box

If the **Enable in place refactorings** check box is cleared on the Editor settings, the **Extract Field** refactoring is performed by means of the **Introduce Field** dialog box.



1. In the editor, select the expression or variable to be replaced with a field, or just place the cursor within such an expression or variable declaration.
2. In the main menu, or the context menu of the selection, choose **Refactor | Extract Field**, or press **Ctrl+Alt+F**.
3. In the **Expressions** pop-up menu, select the expression to be replaced. Note that IntelliJ IDEA highlights the selected expression in the editor.
4. In the **Extract Field Dialog** dialog that opens:
 1. Select the type of the new field from the **Field of type** list.
 2. Specify the name of the field.
 3. Specify where the new field should be initialized by selecting the necessary option under **Initialize in**.
 4. In the **Visibility** area, select the visibility scope for the new field.
 5. If you want the new field to be declared `final`, select the **Declare final** check box.
 6. If the new field 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.
 7. 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.
 8. Click **OK**.

See Also

Reference:

- [Extract Field Dialog](#)

Web Resources:

- [Developer Community](#) 