

Using the Push ITDs In Refactoring

For aspects, IntelliJ IDEA supports most of the basic refactorings such as [Move](#) and [Rename](#). In addition to that, you can also perform the Push ITDs In refactoring for [inter-type declarations](#).

The Push ITDs In refactoring lets you move the definitions of fields and methods from aspects to corresponding classes. (The *ITDs* in the refactoring name stands for *Inter-Type Declarations*.)

The refactoring may be performed for an individual field or method, for an aspect as a whole, or for all aspects in a package.

If when performing the refactoring certain aspects become "empty" (this may be the case when the corresponding aspects contain nothing but inter-type declarations), you can select to automatically delete all such aspects.

- [Examples](#)
- [Performing the Push ITDs In refactoring](#)

Examples

The following table shows examples of the Push ITDs In refactoring.

In the first example, the declaration of the field `closed` is moved from the aspect `MyAspect` to the class `Account`.

In the second example, the method `close()` is also moved from `MyAspect` to `Account`.

Note that both these declarations (that is, the declarations of `closed` and `close()`) may be moved to the class `Account` at once if the refactoring is performed for the aspect `MyAspect` as a whole.

Before	After
<pre>class Account { // some code here } aspect myAspect { boolean Account.closed = false; void Account.close() { closed = true; } // some code here }</pre>	<pre>class Account { // some code here boolean closed = false; } aspect myAspect { void Account.close() { closed = true; } // some code here }</pre>
<pre>class Account { // some code here boolean closed = false; } aspect MyAspect { void Account.close() { closed = true; } // some code here }</pre>	<pre>class Account { // some code here boolean closed = false; void close() { closed = true; } } aspect MyAspect { // some code here }</pre>

To perform the Push ITDs In refactoring

1. Depending on the intended refactoring scope:
 - To perform the refactoring for an individual field or method, open the aspect of interest in the editor and place the cursor within the declaration of the field or method.
 - To perform the refactoring for an aspect as a whole, select the aspect of interest in the **Project** tool window. Alternatively, open the aspect in the editor and place the cursor somewhere outside of individual inter-type declarations (e.g. within the declaration of the aspect).
 - To perform the refactoring for all aspects in a package, select the package in the **Project** tool window.
2. Select **Refactor | Push ITDs In** in the main or the context menu.
3. In the **Push Inter-Type Declarations In** dialog that opens:
 1. Select or clear the **Delete empty aspects** check box.
 2. Click **Refactor** to perform the refactoring right away, or **Preview** to be able to study the expected changes prior to actually performing the refactoring.
4. If at the previous step you clicked **Preview**, the **Find tool window** opens showing the inter-type declarations that are going to be affected. If happy with the expected result, click **Do Refactor**.

See Also

Procedures:

- [Creating Aspects](#)
- [Using the AspectJ \(ajc\) Compiler](#)
- [Refactoring Source Code](#)

Web Resources:

- [Developer Community](#) 