Change Signature Dialog for Java

Refactor | Change Signature

Use the **Change Signature** dialog to perform the Change Method Signature in Java refactoring.

Use the available controls to make changes to the method signature. Specify how the method calls should be handled. Optionally, select the calling methods that the added parameters and exceptions (if any) should be propagated to.

Click **Refactor** to perform the refactoring right away. Click **Preview** to see the potential changes prior to actually performing the refactoring. (These will be shown in the Find tool window.)

- Parameters tab
- Exceptions tab

Item	Description
Visibility	Select the method visibility scope (access level modifier) from the list.
Return type	Use this field to modify the method return type. Code completion (`Ctrl+Space`) is available in this field, and also in other fields used for specifying the types.
Name	Use this field to modify the method name.
Parameters	See the description of the Parameters tab.
Exceptions	See the description of the Exceptions tab.
Method calls	Select one of the following options to specify how the method calls should be handled: - **Modify.** The existing method calls are modified so that the method with the new signature is called. - **Delegate via overloading method.** The existing method calls don't change. A new overloading method with the old signature is created. This new method calls the method with the new signature.
Signature Preview	In this area, the current method signature is shown. (The information in this area is synchronized with the changes you are making to the method signature.)

Parameters tab

Use the **Parameters** tab to manage the method parameters.

The available controls let you add new parameters, remove the existing ones, reorder the parameters and also propagate new parameters to the calling methods (see the descriptions that follow).

In addition to that, you can change the type and name for the existing parameters.

To start editing a parameter, just click it. Alternatively, use the `Up` and `Down` arrow keys to move to the parameter of interest and `Enter` to start modifying it.

Item	Tooltip and shortcut	Description
+	Add `Alt+Insert`	Use this icon or shortcut to start adding a new parameter. Specify the type, name, and default value in the corresponding fields. (The default parameter value is the value (or the expression) to be passed to the method in the method calls.) If necessary, select the **Use Any Var** option. As a result, IntelliJ IDEA will search for a variable of the corresponding type near the method call. If such a variable is found, it will be placed in the method call instead of the default value. If more than one variable is found, or the **Use Any Var** option is not selected, IntelliJ IDEA will use the default value in the call. You can also propagate the parameters you have added to the calling methods.
—	Remove `Alt+Delete`	Use this icon or shortcut to delete the selected parameter.
⬆	Up `Alt+Up`	Use this icon or shortcut to move the selected parameter one line up in the list of parameters.
⬇	Down `Alt+Down`	Use this icon or shortcut to move the selected parameter one line down in the list of parameters.
Ⓟ	Propagate Parameters `Alt+G`	Use this icon or shortcut to propagate the added parameters to the calling methods. You can propagate the changes made to the method parameters to any method that directly or indirectly calls the method whose signature you are changing. (There may be the methods that call the current method. These, in their turn, may be called by other methods. You can propagate the changes to any of the methods in such sequences.) In the dialog that opens, select the methods you want the changes to be propagated to. Note that only the selected calling methods and the method calls within them will be affected. That is, the default values will be added into other method calls.

Exceptions tab

Use the **Exceptions** tab to manage the exceptions thrown by the method.

The available controls let you add new exceptions, remove the existing ones, reorder the exceptions and also propagate new exceptions to the calling methods (see the descriptions that follow).

In addition to that, you can edit the existing exceptions.

To start editing an exception, just click it.

Item	Description
＋ or **`Alt+Insert`**	Use this icon or shortcut to add a new exception. Start typing in the field and then select the required exception type from the suggestion list. Note that you can propagate the exceptions you have added to the calling methods.
— or **`Alt+Delete`**	Use this icon or shortcut to delete the selected exception.
↑ or **`Alt+Up`**	Use this icon or shortcut to move the selected exception one line up in the list of exceptions.
↓ or **`Alt+Down`**	Use this icon or shortcut to move the selected exception one line down in the list of exceptions.
or **`Alt+X`**	Use this icon or shortcut to propagate the added exceptions to the calling methods. You can propagate the exceptions you have added to any method that directly or indirectly calls the method whose signature you are changing. (There may be the methods that call the current method. These methods, in their turn, may be called by other methods. You can propagate the changes to any of the methods in such sequences.) In the dialog that opens, select the methods you want the exceptions to be propagated to.

See Also

Code Examples:

- Change Method Signature Examples for Java

Procedures:

- Change Method Signature in Java

Web Resources:

- Developer Community