

Effective Eclipse: II. Shortcut keys

You should try to keep your hands on keyboard. The less you touch the mouse, the more code you can write. I am trying to keep the mouse laying still and control the IDE completely using keyboard.

Use your hands to write code

You should try to keep your hands on keyboard. The less you touch the mouse, the more code you can write. I am trying to keep the mouse laying still and control the IDE completely using keyboard. What do you think is faster: pressing ALT + C or right clicking the project, selecting Team -> Commit?

It is said, that if a function does not have a key binding, it is useless. Below you will find a set of essential keyboard shortcuts that I love. These shortcuts are set up by default, they should all work.

CTRL + D

Delete row. Try it! You no more need to grab the mouse and select the line, no more Home, Shift + End, Delete. Quick and Clean.

ALT + Up/Down Arrow

Move the row (or the entire selection) up or down. Very useful when rearranging code.

ALT + Left/Right Arrow

Move to the last location you edited. Imagine you just created a class Foo, and now you are working on a class Boo. Now, if you need to look at the Foo class, just press Alt+Left Arrow. Alt+Right Arrow brings you back to Boo.

CTRL+SHIFT+O

Organize imports. What happens when you first use a class you have not yet imported? You will see an error. But when you press this magical combination, all your missing classes will be imported, and the unused imports will vanish.

CTRL+1

Probably the most useful one. It activates the quick fix. Imagine you create a class, which implements some interface. You will get an error, because the inherited methods are not yet implemented. While you are on line where the error occurs, press this combination to activate the quick fix. Now, select the "Add unimplemented methods" option. You can use the quick fix at every error you ever receive.

Quick fix comes handy in other situations too. My favorite is the "Split variable declaration". Sometimes I need to broaden the scope of a variable. I activate the quick fix, split declaration, and use alt + arrow to put it where it belongs. You can find even more usages: Convert local variable to field, rename in file, Inline local variable..

```
public class Foo {
    public void boo() {
        try {
            String bar = "BAR";
        } finally {
            System.out.println(bar);
        }
    }
}
```

You could use the "Split variable declaration" on the bar variable, and then move it with Alt+Arrows above the try block..

```
public class Foo implements ActionListener {
}
```

Or you could use the "Add unimplemented methods" fix here.

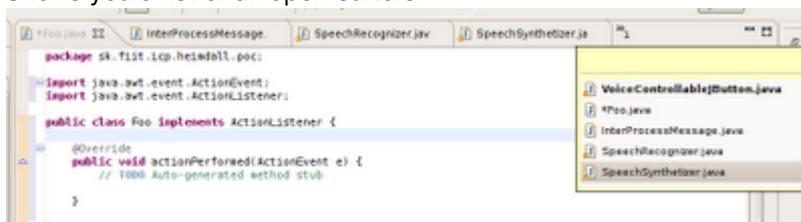
The best thing you can do if you see an error is to use the quick fix.

CTRL+SHIFT+T

Open Type. Imagine, that you need to have a look at the Foo class. But, where is the Foo class? Is it in the Boo project and in the foo.bar package? Or somewhere else? With this shortcut, you don't need to know. Just press it, type Foo and you are in.

CTRL+E

Shows you a list of all open editors.



CTRL+F6

Use to move between open editors. This is an slower alternative to Ctrl + E. Comes handy in a situation when you want to periodically switch between two editors, something, what is nearly impossible with Ctrl+E as it sorts entries quite randomly. Or you might just use Alt+Arrows..

CTRL+F7

Move between views. When in editor, press Ctrl+F7 to switch to the Package Explorer, or hold Ctrl and press F7 multiple times to switch to other views.

CTRL + F11

Runs the application. What gets launched depends on your settings. It will either launch the last launched class (my preferred way) or it will launch currently selected resource (the default way). If you want to change its behavior read the previous post.

CTL + N

Open new type wizard. This is not very quick because you have to select the wizard type (weather you want to create new class, jsp, xml or something else) in the next step. Much faster way would be if you could just hit the shortcut and invoke the particular wizard. It is possible, just keep reading..

CTRL + M

Maximize or umaximize current tab.

CTRL + I

Corrects indentation.

CTRL + SHIFT + F

Formats code. You can make a beautiful looking code out of a mess with this. It requires a bit of setup, but it is well worth it. You can find its settings under Window->Preferences->Java->Code style->Formatter

CTRL + J

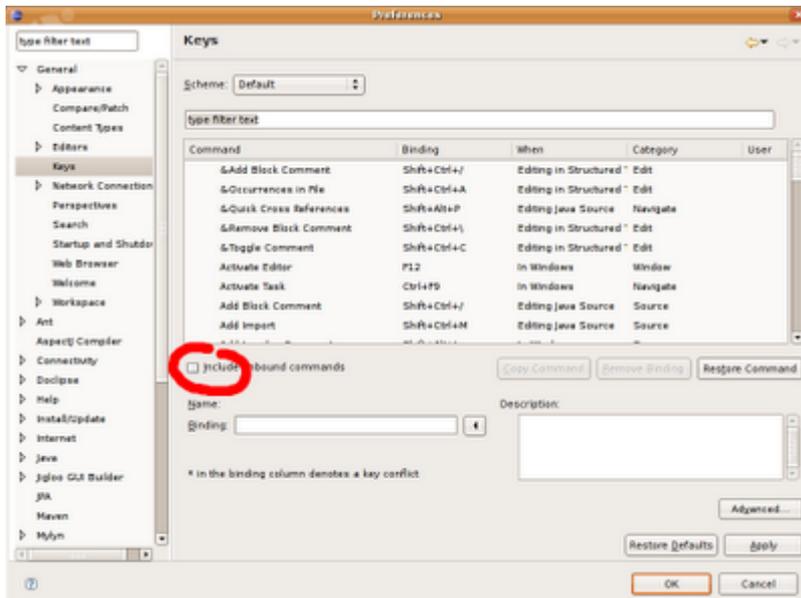
Incremental search. Similar to the search in firefox. It shows you results as you type. Don't be surprised, if you hit this combination, nothing happens - at the first glance. Just start typing and eclipse will move your cursor to the first ocurrence.

CTRL + SHIFT + L

Shows you a list of your currently defined shortcut keys.

I don't like your shortcuts

Such is life nowadays. Remember, you can always change those bindings to match your preferences. Open Windows->Preferences->General->Keys. Now you can use the filter to find your shortcut and change its binding. The real fun begins when you cannot find the command you are looking for. The key here, is to have the "Include unbounds commands" checkbox checked. It will show you all commands, even those, which have no keys bound.



While you are here, I recommend to add the following bindings:

CTRL+SHIFT+G

Bind this to "Generate getters and setters". This is a "must have".

ALT+C

Bind this to SVN/ CVS "Commit".

ALT+U

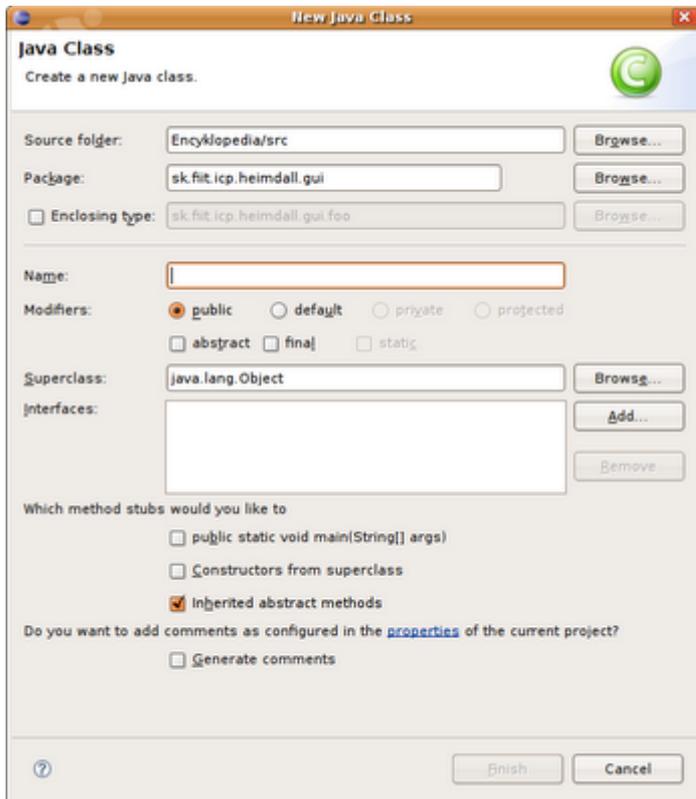
Bind this to SVN/ CVS "Update".

Now, type "new" (without quotes) in the filter text. You should see a list of all new type wizards. Choose the most frequently used and assign them a shortcut. For example, the most used wizard for me is the new class wizard. Thus I assigned it the CTRL+SHIFT+N keys.

Let me demonstrate a quick way to create new class now.

Hit CTRL + SHIFT + N (or the combination you assigned in the previous step). This should bring up new class wizard. Type in the name and press ALT+E. You can now select a class which will be a superclass for the newly created class. Hit ALT+A and select all implemented interfaces . Now hit ALT+F and your class will be generated. Eclipse will also provide the default implementation for all abstract and interface methods you inherited.

Did you notice the weird underscores everywhere in the dialog? They give you a hint about the shortcut key. Hit ALT and the underlined letter to press the button, check the checkbox or get focus for a textfield.



Did you notice the underscores?

I think that using shortcut keys is the fastest way to productivity and if not, then at least your wrists will say you a silent thanks. Now, don't wait, go on and assign keys to the features you use most.

Source: [http://tkramar\(...\)-shortcut-keys_16.html](http://tkramar(...)-shortcut-keys_16.html)

Reprint done with permission of the author

Author: Tomáš Kramár

Article downloaded from page [ioba.com](http://www.ioba.com)