# Effective Eclipse: Shortcut Keys

The less you touch the mouse, the more code you can write. Below you will find a set of essential keyboard shortcuts that I love for Eclipse.

by Tomas Kramar · Feb. 15, 08 · Web Dev Zone · Tutorial

You should try to keep your hands on the keyboard because the less you touch the mouse, the more code you can write. With these shortcuts, I am trying to keep the leave the mouse alone and control the IDE completely using the 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, so they should all work out of the box.

CTRL + D

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

ALT + Up/Down Arrow

Move the row (or the entire selection) up or down. This is very useful when rearranging code. You can even select more rows and move them all at once. Notice that it will be always correctly indented.

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 the line of code 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.

This quick fix comes in 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 uses: convert a local variable to a field, rename something in a file, create inline local variables, etc.

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

- 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 a slower alternative to Ctrl + E. It comes in handy in a situation when you want to periodically switch between two editors, something that 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 the editor, press Ctrl+F7 to switch to the Package Explorer, or hold Ctrl and press F7 multiple times to switch to other views.

CTRL+F8

Move between perspectives. The same as Ctrl+F7.

CTRL + F11

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

CTL + N

Open a new type wizard. This is not very quick because you have to select the wizard type (whether you want to create a new class, JSP, XML, or something else) in the next step. A 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 your current tab.

CTRL + I

Corrects indentation.

CTRL + SHIFT + F

Formats code. You can make 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, when you hit this combination, nothing happens, at first glance at least. Just start typing and Eclipse will move your cursor to the first occurrence.

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 the commands, even those which have no keys bound.

While you are here, I recommend adding 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 the new type wizards. Choose the most frequently used ones 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 key combination.

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

Hit **CTRL + SHIFT + N** (or the combination you assigned in the previous step). This should bring up the 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 focus on a text field.

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 give you a silent "thanks." Now, don't wait, go on and assign keys to the features you use most.

One final tip from Andriy:

The problem is that there are so many keyboard shortcuts. I used to keep a printout with all the shortcuts I wanted to use. Finally, I wrote an Eclipse plugin, MouseFeed, which reminds me of the keyboard shortcuts for the actions called with the mouse. You can even tell it to enforce some shortcuts — the action will run only if called with a keyboard shortcut.

So if you are struggling with it yourself and if you want to use shortcuts, but always subconsciously touch the mouse, install the plugin and let it enforce the shortcuts — the mouse will be useless and you will be forced to use the keyboard.