

Transforming Eclipse into Emacs

This post was originally written in Chinese and published on CSDN.

As an Emacs user for half a year, I simply cannot imagine how I used to code, moving my hands away from the standard position to click the mouse or press the arrow keys without feeling it was cumbersome and unbearable. Now, when I tell my friends that I've set up Alt+P and Alt+N shortcuts to quickly switch between XML files and the Graphical Layout, their response is just "okay," implying that using the mouse to switch is also fine. For me, that's a nightmare; it's just not fast enough! If you're an Emacs user, you understand...

This article describes simple techniques to build a fast Eclipse "editing" environment. Basically, your hands can stay in the standard position, allowing you to code with maximum efficiency!

The most important thing is to install the Emacs+ plugin. See "Emacs+: Emacs Experience in Eclipse".

To make good use of the code assistant, you need to enable it to be triggered by any character and prevent automatic completion when pressing space or =. I recommend downloading this jar file from CSDN. With it, and a quick Google search, you can import packages in no time.

Next, let's customize some shortcuts:

- 1) Bind Alt+P to "Previous Sub-Tab" and Alt+N to "Next Sub-Tab."

The sub-tab is the tab bar below an editor, such as the "Graphical Layout" and "XML" tabs when editing an XML file. This allows you to instantly view the layout.

- 2) Bind Ctrl+C, Ctrl+C to "Run."

This is copied from sbcl's configuration. The default is Ctrl+F11, which is too far away for such a frequently used shortcut, making Emacs users feel terrible! I foolishly pressed Ctrl+F11 for a few days before changing it.

- 3) Bind Ctrl+X, Ctrl+O to "Next View." When In Windows and In Editing Text.

This allows you to instantly jump from the Editor to the Console when writing Java code.

- 4) Bind Ctrl+X, O to "Next Editor." When In Windows and In Editing Text.

This allows you to quickly switch between Java files.

- 5) Bind Ctrl+Q to "Quick Fix."

This way, when you type @string/xx, with the cursor on xx, pressing Ctrl+Q and then Enter will instantly jump to string.xml, with the cursor positioned at the TODO within <string name="xx">TODO</string>.

- 6) Bind Ctrl+Shift+W to “Close”(when in windows) and remove the original binding (close all). The original close shortcut is Ctrl+W, which aligns with our habits in browsers, chat boxes, and file explorers. However, it conflicts with Emacs’cut command. In reality, pressing Ctrl+Shift+W for a second can close many files. So, changing Ctrl+Shift+W from “close all”to “close”doesn’t lose anything.

There’s one problem: after installing Emacs+, when editing code and the code assistant appears, pressing the up and down arrow keys doesn’t select items in the code assistant’s candidate list; instead, it moves up and down in the code being edited. The default is to activate it with F2. Activating the code assistant puts the focus on the code assistant, but then you can only use the up and down arrow keys to select. How great would it be if we could use Ctrl+P and Ctrl+N! Editing code would be lightning fast! But the problem is that after installing the Emacs+ plugin, while Eclipse becomes more like Emacs, this functionality is lost. In a clean Eclipse, with the Emacs (not Emacs+Scheme) keyboard layout selected, you can use Ctrl+N and Ctrl+P to select completion items when the code assistant appears. Someone asked about this on Stack Overflow, but there’s no answer yet.

If we could use Ctrl+P and Ctrl+N to select completion items, that would be truly amazing!