

Emacs Modifier Keys

Emacs Modifier Keys	Emacs makes extensive use of modifier keys and has special notation to describe them. PEL documentation describes the key bindings and use the standard Emacs key notation for describing keys used with modifier keys.			
 Apple Keyboard Symbol	Emacs modifier prefix	Description	Replacement Key Sequence (if any)	Notes
	S	Shift		
	C	Control		Note that in text terminal mode, the keys available to the Control key are limited to the 32 ASCII control sequences. In terminal mode you can have C-a but not C-A, nor C-/.
	M	Meta		- In Windows and Linux, the Meta modifier key is normally mapped to the Alt key. - On macOS, the Meta modifier is best assigned to the <code>⌘ Option</code> key: it can then be used in graphics mode and in text terminal modes running under the Terminal.app and other terminal applications (such as iTerm.app). - Under Terminal.app the <code>⌘ Option</code> key can be easily toggled better a macOS option key and the Meta modifier key by typing the <code>⌘%o</code> keystroke. It can also be configured in Terminal Preferences.
	Esc	Escape		The meta modifier can also be used as the Escape key. For example, the M-x key can be typed as Esc-x .
fn	H	Hyper	C-x @ h	Note: the laptop keyboards position of the Fn key is nicer than stand-alone keyboards, making the Fn key more attractive. for laptops. This attractiveness disappears if the laptop is docked and a standalone keyboard is used often though. Activated in macOS with: <i>(setq ns-function-modifier 'hyper)</i>
	s	Super	C-x @ s	Some of the Mac OS Command keys are used.
	A	Alt	C-x @ a C-x 8	The Alt modifier. Can't find a key for {Windows, Mac OS}

Special Keys - in Strings

Key	String Representation
	<code>"\d"</code>
<ESC>	<code>"\e"</code>
<RET>	<code>"\r"</code>
<tab>	<code>"\t"</code>

Emacs Special Keys

Special character	Definition
<code>\C-x</code>	C-x (where x is any letter)
<code>\C-[</code> or <code>\e</code>	Esc
<code>\M</code>	Meta
<code>\C-j</code> or <code>\n</code>	Newline
<code>\C-m</code> or <code>\r</code>	Enter
<code>\C-i</code> or <code>\t</code>	Tab

Key Sequence Examples

Example	String Format	Vector Format	(kbd argument_string)
a	<code>"a"</code>	<code>[?a]</code>	<code>"a"</code>
Control-a	<code>"\C-a"</code>	<code>[?\C-a]</code>	<code>"C-a"</code>
Control-x 1	<code>"\C-x1"</code>	<code>[?\C-x ?1]</code>	<code>"C-x 1"</code>
Meta-a	<code>"\M-a"</code>	<code>[?\M-a]</code>	<code>"M-a"</code>
Meta-Shift-a	<code>"\M-A"</code>	<code>[?\M-A]</code>	<code>"M-A"</code>
Super-a	<code>"\s-a"</code>		
Hyper-a	<code>"\H-a"</code>		
Control-Meta-q	<code>"\M\C-q"</code>	<code>[?\M-\C-q]</code>	<code>"M-C-q"</code>
Control-z Meta-l	<code>"\C-z \M-l"</code>	<code>[?\C-z ?\M-l]</code>	<code>"C-z M-l"</code>
<f5>		<code>[f5]</code>	<code>"<f5>"</code>
Control-<f5>			<code>"C-<f5>"</code>
Mouse button 1		<code>[mouse-1]</code>	<code>"<mouse-1>"</code>
Control mouse button 1		<code>[C-mouse-1]</code>	<code>"C-<mouse-1>"</code>
Right cursor key			<code>"<right>"</code>
Control-right			<code>"C-<right>"</code>
Control-x <tab>	<code>"\C-x\t"</code>		<code>"C-x <tab>"</code>
ESC-a	<code>"\ea"</code>		
Keypad -		<code>[kp-subtract]</code>	<code>"<kp-subtract>"</code>

Emacs Lisp Functions Used to Bind Keys

define-key	(define-key keymap "keystroke" 'command-name)
global-set-key	(global-set-key "keystroke" 'command-name)
local-set-key	(local-set-key "keystroke" 'command-name)

Typing non-ascii characters - References

Article/URL	Notes
Mastering Key Bindings @ Mastering Emacs	
GNU Emacs Manual - Key Sequences	
GNU emacs manual - Rebinding Keys in your Init file	
Example of key bindings in the GNU Emac Manual - Rebinding Keys in your Init File	Shows how to use the (kbd) function.
Modifier Keys @ EmacsWiki	
Emacs binding discussions and links. @ StackOverflow	
Emacs' Key Syntax Explained @ ErgoEmacs	
Emacs Key Sequences @ ErgoEmacs	
Emacs: Define Key Sequence @ ErgoEmacs	
Prefix Key @ Emacs Wiki	
Define key chords for windmove	
Rebinding the ESC key	https://github.com/mbriggs/emacs.d-oldv2/blob/master/init/init-keymaps.el https://stackoverflow.com/questions/48105812/rebinding-esc-in-insert-mode https://stackoverflow.com/questions/20026083/how-to-use-escape-conditionally-as-a-modifier-key https://stackoverflow.com/questions/11939243/additional-modifier-keys-in-emacs#12010437 https://tex.stackexchange.com/questions/50758/rebinding-keys-in-auctex#50981
macOS modifier key bindings	
Mac OS Keys (symbols)	
A Hyper key for the Mac	
ErgoEmacs: How to Bind Super Hyper Keys	
ErgoEmacs: Mac Keyboard Tools	
How to enter special characters in Bash/Terminal	

Article/URL	Notes
Enhanced key mapper for Mac OS: Karabiner	
Adding Hyper and Super Keys for Emacs @ Irreal	This mostly describes macOS bindings. A reply from Grant Retkke refers to the ns-win.el Emacs Lisp file that provides the means to distinguish the left and right hand side modifier keys on a macOS keyboard for Emacs running in graphics mode. With this file it becomes possible to add the Alt, Super and Hyper key modifiers to Emacs running in graphics mode on macOS!
StackOverflow: Is it possible to make emacs interpret an Fn key as a modifier key?	
Microsoft Windows modifier key bindings	The PEL code binds the Windows Apps key to Emacs Hyper modifier. No extra keyboard control code is used for that. I have not yet identified ways to provide the Super and Alt Emacs modifier. Some of the Windows keys are used as Super, but that is not a very good match since a lot of the Windows keys are intercepted by the OS and never reach Emacs (at least I have not yet found how to do this without modifying the registry or using key control application). In a strict IT controlled environment it is sometimes impossible to install keyboard remapping software. The following articles provide information of what is available.
Emacs: How to Bind Super Hyper Keys @ ErgoEmacs	Xah Lee has a suggestion on his page but his code did not work on my test on Windows 7 and Windows 10.
Emacs super key modifier on Caps Lock in Windows native and Cygwin emacs-w32 @ StackExchange	This entry has several suggestions for adding the Hyper and Super modifier to Emacs running under Windows.
Super/Hyper key in Windows 10 @ reddit	This reddit thread describes an attempt to provide Super and Hyper but the author is asking why it does not work and he gets no answers.
Linux modifier key bindings	
Use Super in emacs -nw in linux	This discussion has several interesting answers, even the ones not directly addressing the question. - Vladimir Panteleev wrote an Emacs package (term-keys.el) that “can teach Emacs and terminals how to properly recognize all PC keyboard keys and modifier combinations”. That might be very helpful for Windows and Linux system running on a PC. - He wrote:”To enable encoding key combinations that include the Super modifier, set term-keys/want-key-p-func to a function which returns t when (elt mods 3) is non-nil (see the default implementation for an explanation)” - Apparently the Konsole terminal emulator application maps the Window logo key to Super. - Another idea from Realraptor is to do the following. I tried it inside macOS terminal running Emacs 26.3 and it does work but you have to type the keys in sequence: H-a is typed <f8> a, one after the other; it's not a true modifier key. It does not work with the cursor key. Here's the code proposed:<pre>(global-set-key [f7] nil) (global-set-key [f8] nil) (define-key function-key-map [(f7)] 'event-apply-super-modifier) (define-key function-key-map [(f8)] 'event-apply-hyper-modifier)</pre>
Are there any linux terminals which can handle all key combinations? @ StackExchange	
General Ideas	
Banish Key Chords @ ErgoEmacs	
Make use of the Super key	A suggested list of Super key bindings