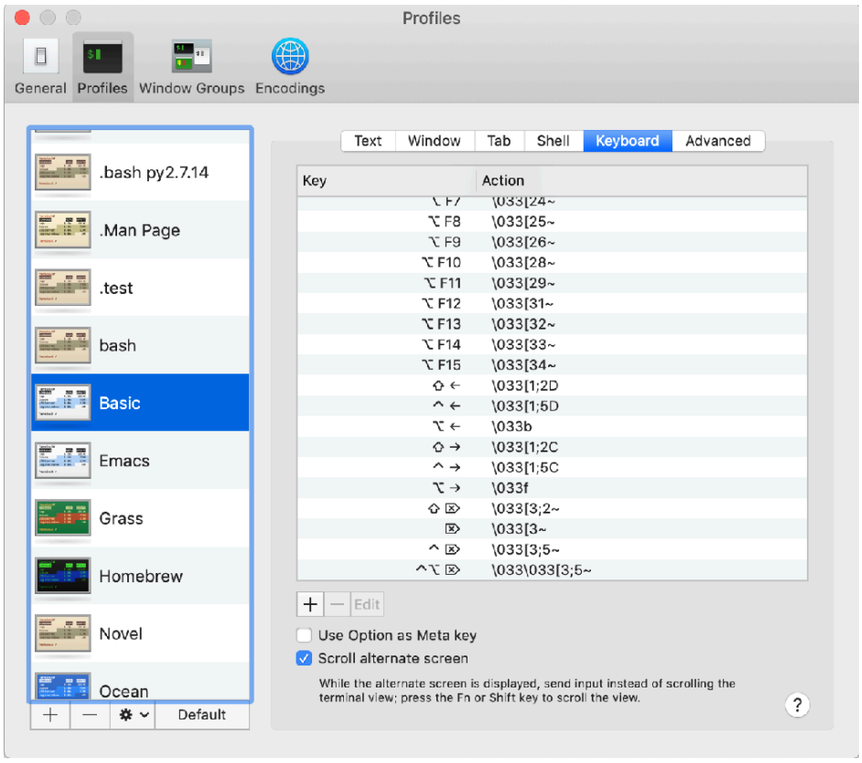
↖ Topic Index

Terminal Settings — Tools For investigation

Application	Type	Description
macOS Tools	The following tools to investigate the keyboard behaviour in macOS terminal emulators and the OS in general are listed in this table.	
Character Viewer	Builtin macOS Application	Used to get passable symbols that represent keys
Key Codes	Third party macOS Application	Used to get Unicode key codes for the keyboard key pressed.
terminal	Builtin macOS Application	• Type **^V followed by the key** in terminal to display the character sequence sent to the application for this key. • Use the Terminal Profiles, section Keyboard to add key mappings. The new mappings are available in the current terminal. If the mapping exists in Emacs it takes affect in Emacs as well. Both profiles are available as different bash shells in Terminal.app
iTerm2	Third party macOS Application	• Type **^V followed by the key** in terminal to display the character sequence sent to the application for this key. Used to check for codes that are not sent in terminal, so we can add them to Terminal Profiles Keyboard mapping.

Use the Terminal Preference dialog, in the Profiles section, then in terminal, to identify extra key codes for missing keys in the Terminal.App terminal emulator.

The following screenshot is an example of the dialog.



The table below shows all codes I was able to configure for the macOS Terminal.app in macOS 10.14.6 (Mojave).

Terminal.app Keys — Profile Mappings

Key Label	Modifier/ Unicode (hex)	Terminal.app Profile mapping	Sequence shown in Terminal after ^V	Defined inside the Keyboard list of macOS Terminal Preferences	Value string extracted in xml file	Notes
Terminal.app Keys	The following table mainly contains the Terminal input ANSI Escape Sequence key codes I was able to configure for Terminal.app on macOS 10.14.6 (Mojave). • This is stored inside a .terminal file, an XML PropertyList-1.0.dtd file. • The Terminal.app supports a large set of key built-in, but not all. For example it supports all the ASCII codes for keys. It also support the cursor keys. For example the <right> key corresponds to the sequence **`ESC [ C`** . This is a built-in sequence. Terminal.app supports a limited number of built in sequences. • Other sequences must be identified inside the Terminal.app Profile and Apple configures several of them, as we can see above in the screenshot of the Basic profile of Terminal.app. But again, that does not include all combination of keys we would need to use Emacs effectively. • It is possible to manually enter more sequences via the dialog shown in the table above. I have entered several key combinations by learning what sequence the terminal generates by using Terminal.app feature of displaying the sequence by first typing Control-V and then the key. This trick unfortunately does not work for all keys; for instance I could not find a way to distinguish the numeric keypad **/** from the main **/** key. This table list the key sequences that are not already part of the Basic Terminal.app profile that I was able to identify and added inside a new profile file. • Several people are annoyed by the current state of terminal emulators and their limitations in identifying all key combinations. Some have written description of the problem, some made proposals and some have implemented packages that try to circumvent the problem, proposing something often called "*lossless keyboard input*". See the references at the bottom of this table. Notes: • **\033** is the octal notation for decimal 27 which is the ASCII value for the **<Esc>** key. • **^[** is the terminal notation to describe Control-[, which also identifies the decimal value 27: the ASCII ASCII value for the **<Esc>** key. The **key names** have the following color codes: • **BLACK**: key available on most keyboards • **PINK** : key available on large desktop keyboards, not available on laptops • **RED** : key only available on specialized keyboards. The Modifier/Unicode(hex) column identifies the main key Unicode value, prefixed with zero, one or several modifier identifier characters. The characters are: • **$** : Shift key (⇧) • **^** : Control key (^) • **~** : Option key (⌥) • **#** : Numpad The Profile mapping is the escape sequence identified. • Sequences in bold are identifiable from the Terminal.app shell by pressing **Control-V** on macOS 10.14.6 (Mojave) and can therefore be entered manually inside the Terminal.app Profile. Note that this also works in some Line terminals. • It is also possible to see the values by executing **`sed -n l`** followed by the key. The last character in the command is a lower-case L.					
Clear (keypad)	F739	**Num Lock**		**Yes**		
F1	F704	**\033OP**	^[OP	**Yes**		
F2	F705	**\033OQ**	^[OQ	**Yes**		
F3	F706	**\033OR**	^[OR	**Yes**		
F4	F707	**\033OS**	^[OS	**Yes**		

Key Label	Modifier/ Unicode (hex)	Terminal.app Profile mapping	Sequence shown in Terminal after ^V	Defined inside the Keyboard list of macOS Terminal Preferences	Value string extracted in xml file	Notes
F5	F708	**\033[15~**	^[[15~	**Yes**		
F6	F709	**\033[17~**	^[[17~	**Yes**		Note that this has the same code as ⌥F1
F7	F70A	**\033[18~**	^[[18~	**Yes**		Note that this has the same code as ⌥F2
F8	F70B	**\033[19~**	^[[19~	**Yes**		Note that this has the same code as ⌥F3
F9	F70C	**\033[20~**	^[[20~	**Yes**		Note that this has the same code as ⌥F4
F10	F70D	**\033[21~**	^[[21~	**Yes**		
F11	F70E	**\033[23~**	^[[23~	**Yes**		
F12	F70F	**\033[24~**	^[[24~	**Yes**		
F13	F710	**\033[25~**	^[[25~	**Yes**		
F14	F711	**\033[26~**	^[[26~	**Yes**		
F15	F712	**\033[28~**	^[[28~	**Yes**		
F16	F713	**\033[29~**	^[[29~	**Yes**		
F17	F714	**\033[31~**	^[[31~	**Yes**		
F18	F715	**\033[32~**	^[[32~	**Yes**		
F19	F716	**\033[33~**	^[[33~	**Yes**		
F20	F717	**\033[34~**	^[[34~	**Yes**		Key not available on standard keyboards.
⇧F1	$F704	**\033[1;2P**	^[[1;2P	**Yes**		Emacs (even in graphics mode) does not support Shift-F1
⇧F2	$F705			No		^V with this key beeps in Terminal
⇧F3	$F706			No		^V with this key beeps in Terminal
⇧F4	$F707			No		^V with this key beeps in Terminal
⇧F5	$F708	**\033[15;2~**	^[[15;2~	**Yes**		
⇧F6	$F709	**\033[17;2~**	^[[17;2~	**Yes**		
⇧F7	$F70A	**\033[18;2~**	^[[18;2~	**Yes**		
⇧F8	$F70B	**\033[19;2~**	^[[19;2~	**Yes**		
⇧F9	$F70C	**\033[20;2~**	^[[20;2~	**Yes**		
⇧F10	$F70D	**\033[21;2~**	^[[21;2~	**Yes**		
⇧F11	$F70E	**\033[23;2~**	^[[23;2~	**Yes**		
⇧F12	$F70F	**\033[24;2~**	^[[24;2~	**Yes**		
⇧F13	$F710			No		
⇧F14	$F711			No		
⇧F15	$F712			No		
⇧F16	$F713			No		
⇧F17	$F714			No		
⇧F18	$F715			No		
⇧F19	$F716			No		
^F1	^F704			No		
^F2	^F705			No		
^F3	^F706			No		
^F4	^F707			No		
^F5	^F708	**\033[15;5~**	^[[15;5~	**Yes**		
^F6	^F709	**\033[17;5~**	^[[17;5~	**Yes**		
^F7	^F70A	**\033[18;5~**	^[[18;5~	**Yes**		
^F8	^F70B	**\033[19;5~**	^[[19;5~	**Yes**		
^F9	^F70C	**\033[20;5~**	^[[20;5~	**Yes**		
^F10	^F70D	**\033[21;5~**	^[[21;5~	**Yes**		
^F11	^F70E	**\033[23;5~**	^[[23;5~	**Yes**		
^F12	^F70F	**\033[24;5~**	^[[24;5~	**Yes**		
^F13	^F710			No		
^F14	^F711			No		
^F15	^F712			No		
^F16	^F713			No		
^F17	^F714			No		
^F18	^F715			No		
^F19	^F716			No		
⌥F1	~F704	**\033[17~**	^[[17~	**Yes**		This has the same code as F6. Emacs see F6
⌥F2	~F705	**\033[18~**	^[[18~	**Yes**		This has the same code as F7. Emacs see F7.
⌥F3	~F706	**\033[19~**	^[[19~	**Yes**		This has the same code as F8. Emacs see F8.
⌥F4	~F707	**\033[20~**	^[[20~	**Yes**		This has the same code as F9. Emacs see F9.
⌥F5	~F708	**\033[15;3~**	^[[15;3~	**Yes**		
⌥F6	~F709	**\033[17;3~**	^[[17;3~	**Yes**		

Key Label	Modifier/ Unicode (hex)	Terminal.app Profile mapping	Sequence shown in Terminal after ^V	Defined inside the Keyboard list of macOS Terminal Preferences	Value string extracted in xml file	Notes
⌥F7	~F70A	\033[18;3~	^[[18;3~	Yes		
⌥F8	~F70B	\033[19;3~	^[[19;3~	Yes		
⌥F9	~F70C	\033[20;3~	^[[20;3~	Yes		
⌥F10	~F70D	\033[21;3~	^[[21;3~	Yes		
⌥F11	~F70E	\033[23;3~	^[[23;3~	Yes		
⌥F12	~F70F	\033[24;3~	^[[24;3~	Yes		
⌥F13	~F710	\033[32~	^[[32~	Yes		
⌥F14	~F711	\033[33~	^[[33~	Yes		
⌥F15	~F712	\033[34~	^[[34~	Yes		
⌥F16	~F713			No		
⌥F17	~F714			No		
⌥F18	~F715			No		
⌥F19	~F716			No		
^⌥F1				No		
^⌥F2				No		
^⌥F3				No		
^⌥F4				No		
^⌥F5		\033[15;7~		Yes		
^⌥F6		\033[17;7~		Yes		
^⌥F7		\033[18;7~		Yes		
^⌥F8		\033[19;7~		Yes		
^⌥F9		\033[20;7~		Yes		
^⌥F10		\033[21;7~		Yes		
^⌥F11		\033[23;7~		Yes		
^⌥F12		\033[24;7~		Yes		
^⌥F13				No		
^⌥F14				No		
^⌥F15				No		
^⌥F16				No		
^⌥F17				No		
^⌥F18				No		
^⌥F19				No		
^⌥⇧F1				No		
^⌥⇧F2				No		
^⌥⇧F3				No		
^⌥⇧F4				No		
^⌥⇧F5		\033[15;8~		Yes		
^⌥⇧F6		\033[17;8~		Yes		
^⌥⇧F7		\033[18;8~		Yes		
^⌥⇧F8		\033[19;8~		Yes		
^⌥⇧F9		\033[20;8~		Yes		
^⌥⇧F10		\033[21;8~		Yes		
^⌥⇧F11		\033[23;8~		Yes		
^⌥⇧F12		\033[24;8~		Yes		
^⌥⇧F13				No		
^⌥⇧F14				No		
^⌥⇧F15				No		
^⌥⇧F16				No		
^⌥⇧F17				No		
^⌥⇧F18				No		
^⌥⇧F19				No		
⌥⇧F1				No		
⌥⇧F2				No		
⌥⇧F3				No		
⌥⇧F4				No		
⌥⇧F5		\033[15;4~		Yes		
⌥⇧F6		\033[17;4~		Yes		
⌥⇧F7		\033[18;4~		Yes		
⌥⇧F8		\033[19;4~		Yes		
⌥⇧F9		\033[20;4~		Yes		

Key Label	Modifier/ Unicode (hex)	Terminal.app Profile mapping	Sequence shown in Terminal after ^V	Defined inside the Keyboard list of macOS Terminal Preferences	Value string extracted in xml file	Notes
⌥⇧F10		\033[21;4~		Yes		
⌥⇧F11		\033[23;4~		Yes		
⌥⇧F12		\033[24;4~		Yes		
⌥⇧F13				No		
⌥⇧F14				No		
⌥⇧F15				No		
⌥⇧F16				No		
⌥⇧F17				No		
⌥⇧F18				No		
⌥⇧F19				No		
^⇧F1	$^F704			No		
^⇧F2	$^F705			No		
^⇧F3	$^F706			No		
^⇧F4	$^F707			No		
^⇧F5	$^F708	\033[15;6~		Yes		
^⇧F6	$^F709	\033[17;6~		Yes		
^⇧F7	$^F70A	\033[18;6~		Yes		
^⇧F8	$^F70B	\033[19;6~		Yes		
^⇧F9	$^F70C	\033[20;6~		Yes		
^⇧F10	$^F70D	\033[21;6~		Yes		
^⇧F11	$^F70E	\033[23;6~		Yes		
^⇧F12	$^F70F	\033[24;6~		Yes		
^⇧F13	$^F710			No		
^⇧F14	$^F711			No		
^⇧F15	$^F712			No		
^⇧F16	$^F713			No		
^⇧F17	$^F714			No		
^⇧F18	$^F715			No		
^⇧F19	$^F716			No		
^⇧F20	$^F717			No		
⌥←		~~\033b~~ \033Y		Yes		This original key sequence here is **\033b** However this keys sequence is problematic: **\033b** corresponds to "**`Esc b`**" which is translated to M-b by Emacs. • The consequence is that it becomes impossible to distinguish **`M-b`** from **`<M-left>`**. • PEL provides a work-around to allow terminal to distinguish **`M-b`** from **`<M-left>`**: set the **pel-map-meta-left-right-to-Y-Z** user-option on so that PEL expects **`<M-Y>`** for the commands that are supposed to be mapped to **`<M-left>`**. Then setup the terminal profile to generate \033Y. • The **`<M-Y>`** key sequence was selected because it is not normally used and also because the **`M-y`** key sequence does not use the Shift marking concept.
^←		\033[1;5D	`^[[1;5D`	Yes		
⇧←		\033[1;2D		Yes		
^⌥←		\033[1;7D		Yes		
^⌥⇧←		\033[1;8D		Yes		
⌥⇧←		\033[1;4D		Yes		
^⇧←		\033[1;6D		Yes		
⌥↑		\033[1;3A		Yes		
^↑		\033[1;5A	`^[[1;5A`	Yes		
⇧↑	$F700	\033[1;2A		Yes	`^[[1;2A`	
^⌥↑		\033[1;7A		Yes		
^⌥⇧↑		\033[1;8A		Yes		
⌥⇧↑		\033[1;4A		Yes		
^⇧↑		\033[1;6A		Yes		
⌥→		~~\033f~~ \033Z		Yes		This original key sequence here is **\033f** However this keys sequence is problematic: **\033f** corresponds to "**`Esc f`**" which is translated to M-f by Emacs. • The consequence is that it becomes impossible to distinguish **`M-f`** from **`<M-right>`**. • PEL provides a work-around to allow terminal to distinguish **`M-f`** from **`<M-right>`**: set the **pel-map-meta-left-right-to-Y-Z** user-option on so that PEL expects **`<M-Z>`** for the commands that are supposed to be mapped to **`<M-right>`**. Then setup the terminal profile to generate \033Z. • The **`<M-Z>`** key sequence was selected because it is not normally used and also because the **`M-z`** key sequence does not use the Shift marking concept.

Key Label	Modifier/ Unicode (hex)	Terminal.app Profile mapping	Sequence shown in Terminal after ^V	Defined inside the Keyboard list of macOS Terminal Preferences	Value string extracted in xml file	Notes
^→		**\033[1;5C**	^[[1;5C	**Yes**		
⇧→		**\033[1;2C**		**Yes**		
^⌥→		**\033[1;7C**		**Yes**		
^⌥⇧→		**\033[1;8C**		**Yes**		
⌥⇧→		**\033[1;4C**		**Yes**		
^⇧→		**\033[1;6C**		**Yes**		
⌥↓		**\033[1;3B**		**Yes**		
^↓		**\033[1;5B**	^[[1;5B	**Yes**		
⇧↓		**\033[1;2B**		**Yes**		
^⌥↓		**\033[1;7B**		**Yes**		
^⌥⇧↓		**\033[1;8B**		**Yes**		
⌥⇧↓		**\033[1;4B**		**Yes**		
^⇧↓		**\033[1;6B**		**Yes**		
^Del>		**\033[3;5~**		**Yes**		
Del>		**\033[3~**		**Yes**		
⇧Del>		**\033[3;2~**		**Yes**		
^⌥Del>		**\033\033[3;5~**		**Yes**		
End						
⇧End						
^End						
^⇧End						
⌥End						
⌥⇧End						
^⌥End						
^⌥⇧End						
Home						
⇧Home						
^Home						
^⇧Home						
⌥Home						
⌥⇧Home						
^⌥Home						
^⌥⇧Home						
^`	^0060	**^[^_*b^_**				
⇧^`	$^0060	**^[^_*c^_**				
^⌥`	^~0060					
⇧^⌥`	$^~0060					

Mappings available in iTerm2 not available in Terminal

Key Label	Mapping	iTerm2 Emacs	Note
End	\033[F	<end>	
⇧End	\033[1;2F		
^End	\033[1;5F	<C-end>	
^⇧End	\033[1;6F		
⌥End	\033[1;9F		
⌥⇧End	\033[1;10F		
^⌥End	\033[1;13F		
^⌥⇧End	\033[1;14F		
Home	\033[H	<home>	
⇧Home	\033[1;2H		
^Home	\033[1;5H	<C-home>	
^⇧Home	\033[1;6H		
⌥Home	\033[1;9H		
⌥⇧Home	\033[1;10H		
^⌥Home	\033[1;13H		
^⌥⇧Home	\033[1;14H		

Terminal Emulator Concepts — References

Topic & Link	Description and Notes
Background Information	The first list of references provide the knowledge on character encoding and escape sequence used by terminal emulators required to understand the way keys are encoded and the limitations of terminal emulators. Understanding this is required if one which to understand the various proposals for "lossless keyboard input" for terminal emulators.
Wikipedia - ASCII simple	A quick overview of what ASCII standard is. The ASCII table shows the control codes in the first column. Those control codes are called Control-*X* where *X* is the character shown in the third column of the table. Which makes **`Ctrl-@`**, **`CTRL-A`** up to **`Ctrl-_`** . Note that has historically been type by holding the **Control** key and the key **A**, without holding the Shift key.
Wikipedia - ASCII	More complete description of the ASCII standard and its history.
Wikipedia - ANSI escape code	The basis of terminal emulator software taking information from typed keys is the ANSI escape sequence codes, more specifically the CSI sequences. This page explains the overall concepts and their history. Note the following: • The **`ESC`** ASCII character is value 27 (base 10), which is 033 octal and 0x1B hexadecimal. • All **escape sequences** start with ESC followed by a second byte in the range 0x40-0x5F (ASCII **`@A-Z[\]^_`**) . • This is the same range of characters selected to represent control characters. • That represent a total of 32 escape sequences. • This 2 byte sequence can be replaced by a single byte, but we can't use that now: it clashes with UTF-8 values. • The **CSI** (Control Sequence Introducer) is a sequence of several bytes: • starting with **`ESC [`** • followed by any number (could be none) of parameter bytes in the range 0x30-0x3F (ASCII **`0-9:;<=>?`**) • sequences containing the parameter bytes **`<=>?`** are considered "private" to the manufacturer. • followed by any number of *intermediate bytes* in the range of 0x20-0x2F (ASCII **`<space>`** and **`!"#$%&'()*+,-./`**) • ending with a *final byte* in the range 0x40-0x7E (ASCII **`@A–Z[\]^_`a–z{|}~`** )   • final byte in the range 0x70-0x7E (**`p–z{|}~`**) are private.
Wikipedia - Unicode range 0000-0FFF	The Unicode range 0000-0FFF holds all letters, numbers and punctuation available on US and most European keyboards. Those values, augmented with modifier keys can be used to represent values normally not supported by terminal emulators, such as **`C-S-a`** and **`C-``** (which do not correspond to ASCII control characters).
Wikipedia - Unicode range E000-F8FF used as private use area	The macOS Unicode value for the cursor and function keys are in 0xF700 - 0xF72F range, which makes them part of the "private use area".
Limitations of Terminal Emulators and improvement proposals	
🚧	TODO
Packages providing Lossless Keyboard Input	
Editing Property Lists with plutil	macOS provides the **plutil** command line utility to test, read, convert and modify macOS Property list files, like the file **~/Library/Preferences/com.apple.Terminal.plist** which contains all Terminal.app preferences. This is the file that needs to be modified to add key bindings, you can use the instructions in term-keys.el package (see below) to do so. ⚠️ Before modifying the file with plutil, **make a backup copy,** in case something goes wrong! ⚠️
Github - term-keys - lossless keyboard input for Emacs	This package allows creating binding to several keys that are not available to Emacs running inside a text (termcap) terminal emulator process. For example, the **`C-``** and **`C-/`** key-chords are normally not accessible in terminal mode, simply because these do not correspond to ASCII control character values. • The term-key package can build the list of translation codes to make these key-chords accessible in terminal-base Emacs. The mechanism used is specific to the terminal emulator software, and several terminal emulators are supported, including the macOS Terminal.app. • Term-key uses a byte sequence prefix that is used for all the extra key definitions. To be able to bind the new keys in Emacs **the prefix used by term-key must not be already used in any Emacs binding**. • The default (but customizable) prefix is "\033\037" which corresponds to **`ESC C-_`** which is **`C-M-_`** binding in Emacs, normally not bound to anything. The term-keys.el readme describes how to make modifications to the Terminal.app Property to support new keys for Emacs. See the macOS Terminal section of the file (⚠️ make a backup of the file first!). • To edit a macOS plist file, use the open command from the shell. It will open the plist file inside Xcode.