ibuffer-mode

Operation	Keystroke	Function	Note
iBuffer Mode See also: ∑ **Buffers**	Emacs ibuffer is a mode where the buffers currently opened are shown as a list of entries, like a directory view. The commands available in this buffer are listed in this table. This information is also available in the ∑ **Buffers** table.		
Open this PDF file. See also: ∑ **Help/Info**	`<f12> <f1>`	(**pel-help-pdf** &optional OPEN-WEB-PAGE)	Open the ∑ **Buffers** local PDF. If the prefix argument (like `C-u` or `M--`) is used, then it opens the remote GitHub hosted raw PDF instead. If the **pel-flip-help-pdf-arg** user-option is set it's the other way around.
∑ Customize Emacs Ibuffer support	`<f12> <f3>`	(**pel-customize-library** &optional OTHER-WINDOW)	Customize Emacs ibuffer. When a prefix argument (like `C-u`) opens the buffer inside another window.
iBuffer Mode	The commands available in the ibuffer window. WIth PEL, the `C-x  C-b` key binding open the ibuffer window.		
IBuffer Mode commands	`S` : Save the marked buffers. `A` : View the marked buffers in the selected frame. `H` : View the marked buffers in another frame. `V` : Revert the marked buffers. `T` : Toggle read-only state of marked buffers. `L` : Toggle lock state of marked buffers. `D` : Kill the marked buffers. `M-s a C-s` : Do incremental search in the marked buffers. `M-s a C-M-s` : Isearch for regexp in the marked buffers. `r` : Replace by regexp in each of the marked buffers. `Q` : Query replace in each of the marked buffers. `I` : As above, with a regular expression. `P` : Print the marked buffers. `O` : List lines in all marked buffers which match a given regexp (like the function 'occur'). `X` : Pipe the contents of the marked buffers to a shell command. `N` : Replace the contents of the marked buffers with the output of a shell command. `!` : Run a shell command with the buffer's file as an argument. `E` : Evaluate a form in each of the marked buffers. This is a very flexible command. For example, if you want to make all of the marked buffers read-only, try using (read-only-mode 1) as the input form. `W` : As above, but view each buffer while the form is evaluated. `k` : Remove the marked lines from the *Ibuffer* buffer, but don't kill the associated buffer. `x` : Kill all buffers marked for deletion.		
IBuffer Mode Marking commands	`m` : Mark the buffer at point. `t` : Unmark all currently marked buffers, and mark all unmarked buffers. `* c` : Change the mark used on marked buffers. `u` : Unmark the buffer at point. `DEL` : Unmark the previous buffer. `M-DEL` : Unmark buffers marked with MARK. `U` : Unmark all marked buffers. `* M` : Mark buffers by major mode. `* u` : Mark all "unsaved" buffers. This means that the buffer is modified, and has an associated file. `* m` : Mark all modified buffers, regardless of whether they have an associated file. `* s` : Mark all buffers whose name begins and ends with '*'. `* e` : Mark all buffers which have an associated file, but that file doesn't currently exist. `* r` : Mark all read-only buffers. `* /` : Mark buffers in 'dired-mode'. `* h` : Mark buffers in 'help-mode', 'apropos-mode', etc. `.` : Mark buffers older than 'ibuffer-old-time'. `d` : Mark the buffer at point for deletion. `% n` : Mark buffers by their name, using a regexp. `% m` : Mark buffers by their major mode, using a regexp. `% f` : Mark buffers by their filename, using a regexp. `% g` : Mark buffers by their content, using a regexp. `% L` : Mark all locked buffers.		
IBuffer Mode Filtering commands	`/ RET` : Add a filter by any major mode. `/ m` : Add a filter by a major mode now in use. `/ M` : Add a filter by derived mode. `/ n` : Add a filter by buffer name. `/ c` : Add a filter by buffer content. `/ b` : Add a filter by basename. `/ f` : Add a filter by filename. `/ .` : Add a filter by file extension. `/ i` : Add a filter by modified buffers. `/ e` : Add a filter by an arbitrary Lisp predicate. `/ >` : Add a filter by buffer size. `/ <` : Add a filter by buffer size. `/ *` : Add a filter by special buffers. `/ v` : Add a filter by buffers visiting files. `/ s` : Save the current filters with a name. `/ r` : Switch to previously saved filters. `/ a` : Add saved filters to current filters. `/ &` : Replace the top two filters with their logical AND. `/ |` : Replace the top two filters with their logical OR. `/ p` : Remove the top filter. `/ !` : Invert the logical sense of the top filter. `/ d` : Break down the topmost filter. `/ /` : **Remove all filtering currently in effect.**		
IBuffer Mode Filter commands	`/ g` : Create filter group from filters. `/ P` : Remove top filter group. `TAB` : Move to the next filter group. `M-p` : Move to the previous filter group. `/ \` : Remove all active filter groups. `/ S` : Save the current groups with a name. `/ R` : Restore previously saved groups. `/ X` : Delete previously saved groups.		
IBuffer Mode Sorting commands	`,` : Rotate between the various sorting modes. `s i` : Reverse the current sorting order. `s a` : Sort the buffers lexicographically. `s f` : Sort the buffers by the file name. `s v` : Sort the buffers by last viewing time. `s s` : Sort the buffers by size. `s m` : Sort the buffers by major mode.		

Operation	Keystroke	Function	Note
IBuffer Mode **Other commands**	**g**	: Regenerate the list of all buffers. Prefix arg means to toggle whether buffers that match 'ibuffer-maybe-show-predicates' should be displayed.	
	`	: Change the *current display format*. 👆 Use this to see the complete file name when the file name is long.	
	SPC	: Move point to the next line.	
	C-p	: Move point to the previous line.	
	h	: Show this help.	
	=	: View the differences between this buffer and its associated file.	
	RET	: View the buffer on this line.	
	o	: As above, but in another window.	
	C-o	: As both above, but don't select the new window.	
	b	: Bury (not kill!) the buffer on this line.	