Emacs Lisp display-buffer commands/variables 🚧

Topic	Description	
Introduction **References** **See also: Windows**	Third party introduction material, listed in recommended access order: • **Emacs: control where buffers are displayed (the 'display-buffer-alist')**, a great 30 minutes presentation of the problem & solution by Protesilaos Stavrou. 2024-02-08 • **Emacs: window rules and parameters (`display-buffer-alist' and extras)**, an 2020-01-07 presentation from Protesilaos Stavrou with more info. • **Demystifying Emacs's Window Manager,** a Mastering Emacs article, by Mickey Petersen. Emacs Lisp Manual: • 29.13 Displaying a Buffer in a Suitable Window 1. Choosing a Window for Displaying a Buffer 2. Action Functions for Buffer Display 3. Action Alists for Buffer Display 4. Additional Options for Displaying Buffers 5. Precedence of Action Functions 6. The Zen of Buffer Display • 29.17.1 Displaying Buffers in Side Windows describes display-buffer-in-side-window, side, slots, dedicated side window. • **side**: denotes the side of the frame where the side window shall be located: *left*, *top*, *right* and *bottom*. • **slot** : specify a window position inside the side window: **< *0*** : above/left, **= *0*** : centre, **> *0***: below/right • **Dedicated side window**: the dedicated flag is set to *side* in a side window to prevent display-buffer to use the window in other action functions. • Side Window Options and Functions • Frame Layouts with Side Windows,	
Displaying buffer in side window	References: • Displaying Buffers in Side Windows • Frame Layouts with Side Windows,	
display-buffer action functions **Display window in:**	In the description cells on the right-hand side, the font and colors have the following meaning: • *'alist-key-symbol'* : alist entry key symbols • *'window-or-frame-parameter-symbol'*: a parameter symbol • '***symbol***' : some other symbol • **'function-variable-or-user-option'** • link to Emacs lisp manual section • `(Elisp source code example)`	
Side window Emacs >= 26.1	(**display-buffer-in-side-window** BUFFER ALIST)	Display BUFFER in a side window of the selected frame. • ALIST is an association list of action symbols and values. • The following two symbols, when used in ALIST, have a special meaning: • *'side'* denotes the side of the frame where the new window shall be located. Valid values are 'bottom', 'right', 'top' and 'left'. The default is 'bottom'. • *'slot'* if non-nil, specifies the window slot where to display BUFFER. (The default is zero.) • **zero** or nil means use the middle slot on the specified side. • A **negative** value means use a slot preceding (that is, above or on the left of) the middle slot. • A **positive** value means use a slot following (that is, below or on the right of) the middle slot. • If the current frame size or the settings of '**window-sides-slots**' do not permit making a new window, a suitable existing window may be reused and have its 'window-slot' parameter value accordingly modified. • Unless '**display-buffer-mark-dedicated**' is non-nil, dedicate the side window used to BUFFER so that it does not get reused by other 'display-buffer' action functions. Return the window used for displaying BUFFER, nil if no suitable window can be found. • This function installs the 'window-side' and 'window-slot' parameters and makes them persistent. It neither modifies ALIST nor installs any other window parameters unless they have been explicitly provided via a '*window-parameters*' entry in ALIST. • This is an action function for buffer display, see '(elisp) Buffer Display Action Functions'. It should be called only by 'display-buffer' or a function directly or indirectly called by the latter.
Same window	(**display-buffer-same-window** BUFFER ALIST)	Tries to display BUFFER in the selected window. It fails if the selected window is a minibuffer window or is dedicated to another buffer (see Dedicated Windows). It also fails if ALIST has a non-'nil' '*inhibit-same-window*' entry.
Reuse window	(**display-buffer-reuse-window** BUFFER ALIST)	Tries to display BUFFER by finding a window that is already displaying it. Windows on the selected frame are preferred to windows on other frames. • If ALIST has a non-nil '*inhibit-same-window'* entry, the selected window is not eligible for reuse. • If ALIST contains a '*reusable-frames'* entry, its value determines which frames to search for a reusable window: • nil -- the selected frame (actually the last non-minibuffer frame) • A frame -- just that frame • 'visible' -- all visible frames • 0 -- all frames on the current terminal • t -- all frames. • If ALIST contains no '*reusable-frames'* entry, search just the selected frame if '**display-buffer-reuse-frames**' and '**pop-up-frames**' are both nil; search all frames on the current terminal if either of those variables is non-nil. • If ALIST has a non-nil '*inhibit-switch-frame'* entry, then in the event that a window on another frame is chosen, avoid raising that frame.
Reuse mode Emacs >= 26.1	(**display-buffer-reuse-mode-window** BUFFER ALIST)	Tries to display BUFFER by finding a window that is displaying a buffer in a given mode. • If ALIST contains a '*mode*' entry, its value specifies a major mode (a symbol) or a list of major modes. • If ALIST contains no '*mode*' entry, the current major mode of BUFFER is used instead. A window is a candidate if it displays a buffer whose mode derives from one of the modes specified thusly. The behavior is also controlled by ALIST entries for '*inhibit-same-window*', '*reusable-frames*' and '*inhibit-switch-frame*', like 'display-buffer-reuse-window' does.
Split largest or LRU window	(**display-buffer-pop-up-window** BUFFER ALIST)	Tries to display BUFFER by splitting the largest or least recently-used window (usually located on the selected frame). It actually performs the split by calling the function specified by '**split-window-preferred-function**' (see Choosing Window Options). • The size of the new window can be adjusted by supplying '*window-height*' and '*window-width*' entries in ALIST. • If ALIST contains a '*preserve-size*' entry, Emacs will also try to preserve the size of the new window during future resize operations (see Preserving Window Sizes). This function fails if no window can be split. More often than not, this happens because no window is large enough to allow splitting. Setting '**split-height-threshold**' or '**split-width-threshold**' to lower values may help in this regard. Splitting also fails when the selected frame has an '*unsplittable*' frame parameter; see Buffer Parameters.
Previously used window Emacs >= 24.3	(**display-buffer-in-previous-window** BUFFER ALIST)	Tries to display BUFFER in a window where it was displayed previously. • If ALIST contains a non-'nil' '*inhibit-same-window*' entry, the selected window is not eligible for use. • A dedicated window is usable only if it already shows BUFFER. • If ALIST contains a '*previous-window*' entry, the window specified by that entry is usable even if it never showed BUFFER before. • If ALIST contains a '*reusable-frames*' entry (see Buffer Display Action Alists), its value determines which frames to search for a suitable window: • nil -- the selected frame (actually the last non-minibuffer frame) • A frame -- just that frame • '*visible*' -- all visible frames • 0 -- all frames on the current terminal • t -- all frames. • If ALIST contains no '*reusable-frames*' entry, this function searches just the selected frame if '**display-buffer-reuse-frames**' and '**pop-up-frames**' are both 'nil'; it searches all frames on the current terminal if either of those variables is non-'nil'. If more than one window qualifies as usable according to these rules, this function makes a choice in the following order of preference: • The window specified by any '*previous-window*' ALIST entry, provided it is not the selected window. • A window that showed BUFFER before, provided it is not the selected window. • The selected window if it is either specified by a '*previous-window*' ALIST entry or showed BUFFER before.
Some existing window Emacs >= 28.1	(**display-buffer-use-some-window** BUFFER ALIST)	Tries to display BUFFER by choosing an existing window and displaying the buffer in that window. • It first tries to find a window that has not been used recently (see Cyclic Window Ordering) on any frame specified by a '*lru-frames*' ALIST entry, falling back to the selected frame if no such entry exists. • It also prefers windows that satisfy the constraints specified by '*window-min-width*' and '*window-min-height*' ALIST entries; preferring full-width windows if no '*window-min-width*' entry is found. • Finally, it will not return a window whose use time is higher than that specified by any '*lru-time*' entry provided by ALIST. If no less recently used window is found, this function will try to use some other window, preferably a large window on some visible frame. It can fail if all windows are dedicated to other buffers (see Dedicated Windows).

Topic	Description	

LRU window
 	
Emacs >= 28.1
	(**display-buffer-use-least-recent-window** BUFFER ALIST)	This function is similar to '**display-buffer-use-some-window'**, but will try harder to not use the a recently used window.
		It:
		• Does not use the selected window.
		• Try first to reuse a window that shows BUFFER already on a frame specified by a '***reusable-frames***' ALIST entry, using the selected frame if no such entry has been specified.
		• Next try to show BUFFER in the least recently used window. The frames to search for such a window can be specified via a '***lru-frames***' ALIST entry; if no such entry exists, search the selected frame only. In addition, try to satisfy constraints specified by the following ALIST entries, if present:
		  • '***lru-time***' specifies a use time. Do not return a window whose use time is higher than this. When calling this action function repeatedly (presumably to display several buffers in a row), an application should first save the use time of the selected window and pass that same value via such an entry in each call of 'display-buffer'. This reduces the probability that 'display-buffer' uses the same window as a previous call.
		  • '***window-min-width***' specifies a preferred minimum width in canonical frame columns. If it is the constant '***full-width***', prefer a full-width window.
		  • '***window-min-height***' specifies a preferred minimum height in canonical frame lines. If it is the constant '**full-height**', prefer a full-height window.

Select window by direction
 	
Emacs >= 27.1
	(**display-buffer-in-direction** BUFFER ALIST)	Try to display BUFFER in a direction specified by ALIST.
		
		• ALIST has to contain a '***direction***' entry whose value should be one of '***left***', '***above***' (or '***up***'), '***right***' and '***below***' (or '***down***'). Other values are usually interpreted as '***below***'.
		  • Four special values for '***direction***' entries allow to implicitly specify the selected frame's main window as reference window: '***leftmost***', '***top***', '***rightmost***' and '***bottom***'.
		  Hence, instead of '`(direction . left) (window . main)`' one can simply write '`(direction . leftmost)`'.
		
		• If ALIST also contains a '***window***' entry, its value specifies a reference window. That value can be a special symbol like 'main' (which stands for the selected frame's **main window**) or 'root' (standings for the selected frame's **root window**) or an arbitrary valid window. Any other value (or omitting the '***window***' entry) means to use the selected window as reference window.
		
		This function tries to reuse or split a window such that the window produced this way is on the side of the reference window specified by the '***direction***' entry.

Below selected window Emacs >= 24.3	(**display-buffer-below-selected** BUFFER ALIST)	This function tries to display BUFFER in a window below the selected window. • If there is a window below the selected one and that window already displays BUFFER, it reuses that window. • If there is no such window, this function tries to create a new window by splitting the selected one, and displays BUFFER there. • It will also try to adjust that window's size provided ALIST contains a suitable '***window-height***' or '***window-width***' entry, see above. • If splitting the selected window fails and there is a non-dedicated window below the selected one showing some other buffer, this function tries to use that window for showing BUFFER. • If ALIST contains a '***window-min-height***' entry, this function ensures that the window used is or can become at least as high as specified by that entry's value. Note that this is only a guarantee. In order to actually resize the window used, ALIST must also provide an appropriate '***window-height***' entry.
Bottom of frame Emacs >= 24.4	(**display-buffer-at-bottom** BUFFER ALIST)	Tries to display BUFFER in a window at the bottom of the selected frame. • This either tries to split the window at the bottom of the frame or the frame's root window, or to reuse an existing window at the bottom of the selected frame.
New frame	(**display-buffer-pop-up-frame** BUFFER ALIST)	Creates a new frame, and displays the buffer in that frame's window. • It actually performs the frame creation by calling the function specified in '**pop-up-frame-function**' (see **Additional Options for Displaying Buffers**). • If ALIST contains a 'pop-up-frame-parameters' entry, the associated value is added to the newly created frame's parameters.
Entire current frame Emacs >= 29.1	(**display-buffer-full-frame** BUFFER ALIST)	Displays the buffer on the current frame, deleting all other windows so that it takes up the full frame.
Child frame	(**display-buffer-in-child-frame** BUFFER ALIST)	Display BUFFER in a **child frame**. By default, this either reuses a child frame of the selected frame or makes a new child frame of the selected frame. If successful, return the window used; otherwise return nil. • If ALIST has a non-nil '***child-frame-parameters***' entry, the corresponding value is an alist of frame parameters to give the new frame. A '***parent-frame***' parameter specifying the selected frame is provided by default. • If the child frame shall be or become the child of any other frame, a corresponding entry must be added to ALIST.
Existing frame Emacs >= 25.1	(**display-buffer-use-some-frame** BUFFER ALIST)	Tries to display BUFFER by finding a frame that meets a predicate (by default any frame other than the selected frame). • If this function chooses a window on another frame, it makes that frame visible and, unless ALIST contains an '***inhibit-switch-frame***' entry, raises that frame if necessary. • If ALIST has a non-'nil' '***frame-predicate***' entry, its value is a function taking one argument (a frame), returning non-'nil' if the frame is a candidate; this function replaces the default predicate. • If ALIST has a non-'nil' '***inhibit-same-window***' entry, the selected window is not used; thus if the selected frame has a single window, it is not used.
Do not display Emacs >= 24.4	(**display-buffer-no-window** BUFFER ALIST)	Display BUFFER in no window. • If ALIST has a non-'nil' '***allow-no-window***' entry, then this function does not display BUFFER and returns the symbol 'fail'. • This constitutes the only exception to the convention that an action function returns either 'nil' or a window showing BUFFER. If ALIST has no such '***allow-no-window***' entry, this function returns 'nil'. If this function returns 'fail', 'display-buffer' will skip the execution of any further display actions and return 'nil' immediately. If this function returns 'nil', 'display-buffer' will continue with the next display action, if any. • It is assumed that when a caller of 'display-buffer' specifies a non-'nil' 'allow-no-window' entry, it is also able to handle a 'nil' return value.