Emacs Lisp display-buffer commands/variables 🚧

Topic	Description	
Introduction **References** **See also: Σ 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,	
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.
In atom window	(**display-buffer-in-atom-window** BUFFER ALIST)	Display BUFFER in an atomic window. • This function displays BUFFER in a new window that will be combined with an existing window to form an atomic window. • If the existing window is already part of an atomic window, add the new window to that atomic window. • Operations like '**split-window**' or '**delete-window**', when applied to a constituent of an atomic window, are applied atomically to the root of that atomic window. The following two symbols have a special meaning: • '***window***' specifies the existing window the new window shall be combined with. • Use '**window-atom-root'** to make the new window a sibling of an atomic window's root. • If an internal window is specified here, all children of that window become part of the atomic window too. • If no window is specified, the new window becomes a sibling of the selected window. • By default, the '***window-atom***' parameter of the existing window is set to 'main' provided the window is live and the parameter is not set yet. • '***side***' denotes the side of the existing window where the new window shall be located. • Valid values are '***below***', '***right***', '***above***' and '***left***'. The default is '***below***'. • By default, the '***window-atom***' parameter of the new window is set to this value.
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.

Topic	Description	
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**).

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.
Action list symbols	The symbols are listed below in alphabetical order for quick reference.	
allow-no-window	If the value is non-'nil', '**display-buffer**' does not necessarily have to display the buffer and the caller is prepared to accept that. • ⚠ This entry is not intended for user customizations, since there is no guarantee that an arbitrary caller of 'display-buffer' will be able to handle the case that no window will display the buffer. '**display-buffer-no-window**' is the only action function that cares about this entry.	
body-function	The value must be a function taking one argument (a displayed window). • This function can be used to fill the displayed window's body with some contents that might depend on dimensions of the displayed window. • It is called after the buffer is displayed, and before the entries '***window-height***', '***window-width***' and '***preserve-size***' are applied that could resize the window to fit it to the inserted contents.	

bump-use-time	If non-'nil', such an entry will cause 'display-buffer' to bump the use time (see Selecting Windows) of the window it uses.
	• This should avoid later use of this window by action functions like '**display-buffer-use-some-window**' and '**display-buffer-use-least-recent-window**' for showing another buffer.
	• There is a fine difference between using this entry and using the action function '**display-buffer-use-least-recent-window**'.
	  • Calling the latter means to only bump the use times of windows that function uses for displaying the buffer.
	  • The entry described here will cause 'display-buffer' to bump the use time of *any* window used for displaying a buffer.	

child-frame-parameters	The value specifies an alist of frame parameters to use when the buffer is displayed on a child frame. • This entry is used only by '**display-buffer-in-child-frame**'.	
dedicated	If non-'nil', such an entry tells '**display-buffer**' to mark any window it creates as dedicated to its buffer (see Dedicated Windows). • It does that by calling '**set-window-dedicated-p**' with the chosen window as first argument and the entry's value as second. • Side windows are by default dedicated with the value 'side' (see Side Window Options and Functions).	
direction	The value specifies a direction which, together with a '***window***' entry, allows '**display-buffer-in-direction**' to determine the location of the window to display the buffer.	
frame-predicate	The value must be a function taking one argument (a frame), supposed to return non-'nil' if that frame is a candidate for displaying the buffer. • This entry is used by '**display-buffer-use-some-frame**'.	

Topic	Description
inhibit-same-window	If the value is non-'nil', this signals that the selected window must not be used for displaying the buffer. • All action functions that (re-)use an existing window should respect this entry.
inhibit-switch-frame	A non-'nil' value prevents another frame from being raised or selected, if the window chosen by 'display-buffer' is displayed there. • Primarily affected by this are '**display-buffer-use-some-frame**' and '**display-buffer-reuse-window**'. • Ideally, '**display-buffer-pop-up-frame**' should be affected as well, but there is no guarantee that the window manager will comply.
lru-frames	The value specifies the set of frames to search for a window that can be used to display the buffer. • It is honored by '**display-buffer-use-some-window**' and '**display-buffer-use-least-recent-window**' when trying to find a less recently used window showing some other buffer. Its values are the same as for the '***reusable-frames***' entry described below.
lru-time	The value is supposed to specify a use time (see Selecting Windows). • This entry is honored by '**display-buffer-use-some-window**' and '**display-buffer-use-least-recent-window'** when trying to find a less recently used window showing some other buffer. If a window's use time is higher than the value specified by this option, these action functions will not consider such a window for displaying the buffer.
mode	The value is either a major mode or a list of major modes. • '**display-buffer-reuse-mode-window**' may reuse a window whenever the value specified by this entry matches the major mode of that window's buffer. • Other action functions ignore such entries.
parent-frame	The value specifies the parent frame to be used when the buffer is displayed on a child frame. • This entry is used only by '**display-buffer-in-child-frame**'.
pop-up-frame-parameters	The value specifies an alist of frame parameters to give a new frame, if one is created. • '**display-buffer-pop-up-frame**' is its one and only addressee.

preserve-size	If non-'nil' such an entry tells Emacs to preserve the size of the window chosen (see Preserving Window Sizes).
	• The value should be either:
	  • '(t . nil)' to preserve the width of the window,
	  • '(nil . t)' to preserve its height or
	  • '(t . t)' to preserve both its width and its height.
	
	This entry should be processed only under certain conditions which are specified right after this list.

previous-window	The value must specify a window that may have displayed the buffer previously. • '**display-buffer-in-previous-window**' will give preference to such a window provided it is still live and not dedicated to another buffer.

reusable-frames	The value specifies the set of frames to search for a window that can be reused because it already displays the buffer.
	It can be set as follows:
	  • 'nil' means consider only windows on the selected frame. (Actually, the last frame used that is not a minibuffer-only frame.)
	  • 'visible' means consider windows on all visible frames.
	  • 0 means consider windows on all visible or iconified frames.
	  • A frame means consider windows on that frame only.
	• 't' means consider windows on all frames. (Note that this value is rarely the right thing to use—it might also return a tooltip frame.)
	
	Note that the meaning of 'nil' differs slightly from that of the ALL-FRAMES argument to '**next-window**' (see Cyclic Window Ordering).
	
	A major client of this is '**display-buffer-reuse-window**', but all other action functions that try to reuse a window are affected as well.
	'**display-buffer-in-previous-window**' consults it when searching for a window that previously displayed the buffer on another frame.

side	The value denotes the side of the frame or window where a new window displaying the buffer shall be created. • This entry is used by '**display-buffer-in-side-window**' to indicate the side of the frame where a new side window shall be placed (see Displaying Buffers in Side Windows). • It is also used by '**display-buffer-in-atom-window**' to indicate the side of an existing window where the new window shall be located (see Atomic Windows).

slot	If non-'nil', the value specifies the slot of the side window supposed to display the buffer.
	The values are:
	  • **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.
	This entry is used only by '**display-buffer-in-side-window**'.

window	The value specifies a window that is in some way related to the window chosen by 'display-buffer'. • This entry is currently used by '**display-buffer-in-atom-window'** to indicate the window on whose side the new window shall be created. • It is also used by '**display-buffer-in-direction**' to specify the reference window on whose side the resulting window shall appear.

window-height	The value specifies whether and how to adjust the height of the chosen window and can be one of the following:
	  • 'nil' means to leave the height of the chosen window alone.
	  • An integer number specifies the desired total height of the chosen window in lines.
	  • A floating-point number specifies the fraction of the chosen window's desired total height with respect to the total height of its frame's root window.
	  • A cons cell whose CAR is 'body-lines' and whose CDR is an integer that specifies the height of the chosen window's body in frame lines.
	  • If the value specifies a function, that function is called with one argument—the chosen window. The function is supposed to adjust the height of the window; its return value is ignored. Suitable functions are '**fit-window-to-buffer**' and '**shrink-window-if-larger-than-buffe**r', see Resizing Windows.
	
	By convention, the height of the chosen window is adjusted only if the window is part of a vertical combination (see Windows and Frames) to avoid changing the height of other, unrelated windows. Also, this entry should be processed only under certain conditions which are specified right below this list.

window-min-height	The value specifies a minimum height of the window used, in canonical frame lines. • The special value '***full-height***' means the chosen window should be a full-height window, one that has no other windows above or below it in its frame. This entry is currently honored by '**display-buffer-below-selected'** which does not use a window that is not as high as specified by this entry. It's also honored by '**display-buffer-use-some-window**' and '**display-buffer-use-least-recent-window**' which try hard to avoid returning a less recently used window if it does not satisfy this constraint. Note that providing such an entry alone does not necessarily make the window as tall as specified by its value. • To actually resize an existing window or make a new window as tall as specified by that value, a '***window-height***' entry specifying that value should be provided as well. • Such a '***window-height***' entry can, however, specify a completely different value or ask the window height to be fit to that of its buffer in which case the '***window-min-height***' entry provides the guaranteed minimum height of the window used.
window-min-width	The value specifies a minimum width of the window used, in canonical frame columns. • The special value '***full-width***' means the chosen window should be one that has no other windows on the left or right of it in its frame. This entry is currently honored by '**display-buffer-use-some-window**' and '**display-buffer-use-least-recent-window**', which try hard to avoid returning a less recently used window that does not satisfy the entry. Note that providing such an entry alone does not necessarily make the window as wide as specified by its value. • To actually resize an existing window or make a new window as wide as specified by this entry's value, a '***window-width***' entry specifying that value should be provided as well. • Such a '***window-width***' entry can, however, specify a completely different value, or ask the window width to fit that of its buffer, in which case the '***window-min-width***' entry provides the guaranteed minimum width of the window.
window-parameters	The value specifies an alist of window parameters to give the chosen window. • All action functions that choose a window should process this entry

window-size	This entry is a combination of the '***window-height***' and '***window-width***' and can be used to adjust the chosen window's height and width.
	• Since windows can be resized in one direction only without affecting other windows, '***window-size'*** is effective only to set up the size of a window appearing alone on a frame.
	• The value can be one of the following:
	  • 'nil' means to leave the size of the chosen window alone.
	  • A cons cell of two integers specifies the desired total width and height of the chosen window in lines and columns. It's effect is to adjust the size of the frame accordingly.
	  • A cons cell whose CAR equals 'body-chars' and whose CDR is a cons cell of two integers—the desired body width and height of the chosen window in frame columns and lines. It's effect is to adjust the size of the frame accordingly.
	  • If the value specifies a function, that function is called with one argument—the chosen window. The function is supposed to adjust the size of the window's frame; its return value is ignored.
	
	This entry should be processed under only certain conditions which are specified right below this list.

window-width	This entry is similar to the '***window-height***' entry described before, but used to adjust the chosen window's width instead.
	The value can be one of the following:
	  • 'nil' means to leave the width of the chosen window alone.
	  • An integer specifies the desired total width of the chosen window in columns.
	  • A floating-point number specifies the fraction of the chosen window's desired total width with respect to the total width of the frame's root window.
	  • A cons cell whose CAR is '***body-columns***' and whose CDR is an integer that specifies the width of the chosen window's body in frame columns.
	  • If the value specifies a function, that function is called with one argument—the chosen window. The function is supposed to adjust the width of the window; its return value is ignored.