ATTACHEDWINDOW

An Interlisp-D package for manipulating windows in groups

Written by: Richard R. Burton
Last modified: June 2, 1984, van Melle
Located on: <LispCore>Sources>ATTACHEDWINDOW.DCOM

Introduction

ATTACHEDWINDOW is a package designed to make it easy to manipulate a group of window as a unit. Standard operations like MOVE , RESHAPE , OPEN and CLOSE can be done so that it appears to the user as if the windows are a single entity. Each collection of attached windows has one main window and any number of other windows that are "attached" to it. Moving or reshaping the main window causes all of the attached windows to be moved or reshaped as well. Moving or reshaping an attached window does not affect the main window. The initial motivation for attached windows was to allow multiple menus to be associated with the same window but there is no restriction on what windows can be attached.

(ATTACHWINDOW *WINDOWTOATTACH MAINWINDOW EDGE POSITIONONEDGE WINDOWCOMACTION*)

Associates *WINDOWTOATTACH* with *MAINWINDOW* so that shape, move, close, shrink and expand operations done to *MAINWINDOW* are also done to *WINDOWTOATTACH* . ATTACHWINDOW moves *WINDOWTOATTACH* to its position relative to *MAINWINDOW* but does not open it.

EDGE and *POSITIONONEDGE* indicate where *WINDOWTOATTACH* is to be positioned. The argument *EDGE* determines which edge: TOP , BOTTOM , LEFT , or RIGHT . The default, used if EDGE is NIL, is TOP. The argument *POSITIONONEDGE* determines where along edge the window is positioned:

JUSTIFY means that the attached window is to fill the entire edge. ATTACHWINDOW reshapes the window if necessary;

LEFT or RIGHT for the left or right (for a TOP or BOTTOM edge);

BOTTOM or TOP for the bottom or top (of a LEFT or RIGHT edge);

CENTER for the center of the edge.

The default for *POSITIONONEDGE* is JUSTIFY.

The size that is filled by the justification includes the extent of any other windows that have already been attached to *MAINWINDOW* . Thus (ATTACHWINDOW A MW 'RIGHT 'JUSTIFY) followed by (ATTACHWINDOW B MW 'TOP 'JUSTIFY) will put B across the top of both MW and A.

WINDOWCOMACTION provides a convenient way of setting the way the attached window responds to right buttoning. If *WINDOWCOMACTION* is MAIN, the DOWINDOWCOMFN of *WINDOWTOATTACH* is set to DOMAINWINDOWCOMFN. If *WINDOWCOMACTION* is HERE, the DOWINDOWCOMFN of *WINDOWTOATTACH* is not changed. If *WINDOWCOMACTION* is LOCALCLOSE, the DOWINDOWCOMFN of *WINDOWTOATTACH* is set to DOATTACHEDWINDOWCOM2. Otherwise, the DOWINDOWCOMFN of *WINDOWTOATTACH* is set to DOATTACHEDWINDOWCOM. These functions are described below in the section on "attached window menu commands."

(DETACHWINDOW *WINDOWTODETACH*)

Detaches *WINDOWTODETACH* from its main window. Returns a dotted pair whose CAR is EDGE and whose CDR is POSITIONONEDGE if *WINDOWTODETACH* was an attached window. Returns NIL otherwise. This does not close *WINDOWTODETACH* .

Behavior on standard window operations

When a window operation, such as moving or clearing, is performed on a window, there is a question about

whether or not that operation also be performed on the windows attached to it or performed on the window it is attached to. The following are the default behaviors of main and attached windows under the window operations when invoked programmatically, e.g., from the functions MOVEW , SHAPEW , etc.

The behavior when an operation is invoked from the window menu depends on the *WINDOWCOMACTION* argument to ATTACHWINDOW, or ultimately the window's DOWINDOWCOMFN property. Mention of "menu" below assumes that the window was attached using the default (NIL) *WINDOWCOMACTION* .

The behavior for any particular operation can, of course, be changed for particular windows by setting the standard window properties (e.g., MOVEFN or CLOSEFN) of the particular attached window.

Move: If the main window moves, all attached windows move with it, and the relative positioning between the main window and the attached windows is maintained. If the region is determined interactively, the prompt region for the move is the union of the extent of the main window and all attached windows.

MOVEW moves an attached window without affecting the main window. The Move command in the window menu is by default passed on to the main window, so that all windows in the group move.

Reshape: If the main window is reshaped, the minimum size of it and all of its attached windows is used as the minimum of the space for the result. Any space greater than the minimum is distributed among the main window and its attached windows.

SHAPEW reshapes an attached window independently. The Shape command in the window menu is by default passed on to the main window.

Close: If the main window is closed, all of the attached windows are closed also and the links from the attached windows to the main window are broken. This is necessary for the windows to be garbage collected.

CLOSEW closes an attached window without affecting the main window. Close in the window menu is by default passed on to the main window. If *WINDOWCOMACTION* of LOCALCLOSE was specified in the call to ATTACHWINDOW, the menu Close operates independently. Note that closing an attached window does *not* detach it.

Open: If the main window is opened, it opens all attached windows and reestablishes links from them to the main window.

Attached windows can be opened independently and this does not affect the main window.

Shrink: The collection of windows shrinks as a group. The SHRINKFN s of the attached windows are evaluated but the only icon displayed is the one for the main window. {How does the system know that the attached windows are shrunken rather than closed? It doesn't. Does this make any difference?}

Redisplay: The main or attached windows can be redisplayed independently.

Totop: If any main or attached window is brought to the top, all of the other windows are brought to the top also.

Expand: Expanding any of the windows expands the whole collection.

Scrolling: All of the windows involved in the group scroll independently.

Clear: All windows clear independently of each other.

Attached window menu commands

The question of how to handle the window command menu of any particular window (either by right buttoning or by a call to the function DOWINDOWCOM) is handled by the window's DOWINDOWCOMFN property. The *WINDOWCOMACTION* argument to ATTACHWINDOW

selects one of the following three functions to be the attached window's DOWINDOWCOMFN property, or leaves the property NIL if *WINDOWCOMACTION* is HERE, meaning to use the standard window command menu, ignorant of the window's attachments. The programmer can instead supply her own DOWINDOWCOMFN property if some other behavior is desired.

(DOATTACHEDWINDOWCOM *ATTACHEDW LOCALCLOSEFLG*)

The default (when *WINDOWCOMACTION* is NIL). Brings up the window command menu and then, depending upon the command selected, either passes the command to the main window of *ATTACHEDW* or performs it on *ATTACHEDW*. If *LOCALCLOSEFLG* is non-NIL, the CLOSE command is applied to *ATTACHEDW*. Otherwise, the CLOSE command is passed to the main window. The commands MOVE, RESHAPE, SHRINK and BURY are passed to the main window. The other commands are performed on *ATTACHEDW*.

(DOATTACHEDWINDOWCOM2 *ATTACHEDW*)

Used when *WINDOWCOMACTION* is LOCALCLOSE. Performs (DOATTACHEDWINDOWCOM *ATTACHEDW* T) so that the command CLOSE is performed on *ATTACHEDW*.

(DOMAINWINDOWCOMFN *ATTACHEDW*)

Used when *WINDOWCOMACTION* is MAIN. Performs DOWINDOWCOM on the window that is the MAINWINDOW property of *ATTACHEDW*. In other words, assuming the DOWINDOWCOMFN of the main window hasn't been changed, this passes all window commands on to the main window.

Attaching menus to windows

ATTACHEDWINDOW supersedes the MENUEDWINDOW package and users of it are encouraged to convert their code. The following functions are provided to associate menus to windows.

(ATTACHMENU *MENU MAINWINDOW EDGE POSITIONONEDGE*)

Creates a window that contains the menu *MENU* (by calling MENUWINDOW, see below) and attaches it to the window *MAINWINDOW* on edge *EDGE* at position *POSITIONONEDGE*.

(MENUWINDOW *MENU VERTFLG*)

Returns a closed window that has the menu *MENU* in it. If *MENU* is a list, a menu is created with *MENU* as its items. Otherwise, *MENU* should be a menu. The returned window has the appropriate RESHAPEFN, MINSIZE and MAXSIZE window properties to allow its use in a window group. *VERTFLG* is provided to allow convenient setting of the default menu shape and will only be considered if both the MENUROWS and MENUCOLUMNS fields of *MENU* are NIL. If *VERTFLG* is non-NIL, the MENUROWS field of *MENU* will be set to 1; otherwise the MENUCOLUMNS field of *MENU* will be set to 1.

(CREATEMENUEDWINDOW *MENU WINDOWTITLE LOCATION WINDOWSPEC*)

Creates a window with an attached menu and returns the main window. *MENU* is the only required argument, and may be a menu or a list of menu items. *WINDOWTITLE* is a string specifying the title of the main window. *LOCATION* specifies the edge on which to place the menu, as with ATTACHWINDOW's *EDGE* argument; the default is TOP. *WINDOWSPEC* is a REGION, specifying a region for the aggregate window; if NIL, the user is prompted for a region.

This function is similar to MENUEDWINDOW's function MAKEMENUEDWINDOW. However, note that it is not an exact replacement. In particular, the MENUWINDOW property is not used.

Examples:

```
(SETQ MENUW
```

```
(MENUWINDOW (create MENU
                ITEMS _ '(smaller LARGER)
                MENUFONT _ '(GACHA 12)
                TITLE _ "zoom controls"
                CENTERFLG _ T
                WHENSELECTEDFN _ (FUNCTION ZOOMMAINWINDOW))))
```

creates a menu window that contains the two items "smaller" and "LARGER" with the title "zoom controls" and that calls the function ZOOMMAINWINDOW when an item is selected.

```
(ATTACHWINDOW MENUW (CREATEW '(50 50 150 150)) 'TOP 'JUSTIFY)
```

creates a window on the screen and attaches the above created menu window to its top.

```
(CREATEMENUEDWINDOW (CREATE MENU
                ITEMS _ '(smaller LARGER)
                MENUFONT _ '(GACHA 12)
                TITLE _ "zoom controls"
                CENTERFLG _ T
                WHENSELECTEDFN _ (FUNCTION ZOOMMAINWINDOW))))
```

creates the same sort of window in one step, prompting the user for a region.

Attached Prompt Windows

Many packages have a need to display status information or prompt for small amounts of user input in a place outside their standard window. A convenient way to do this is to attach a small window to the top of the program's main window. The following functions do so in a uniform way that can be depended on among diverse applications.

(GETPROMPTWINDOW *MAINWINDOW #LINES FONT DONTCREATE*)

Returns the attached prompt window associated with *MAINWINDOW*, creating it if necessary. The window is always attached to the top of *MAINWINDOW*, has DSPSCROLL set to T, and has a PAGEFULLFN of NILL to inhibit page holding. The window is at least *#LINES* lines high (default 1); if a pre-existing window is shorter than that, it is reshaped to make it large enough. *FONT* is the font to give the prompt window (defaults to the font of *MAINWINDOW*), and applies only when the window is first created. If *DONTCREATE* is true, returns the window if it exists, otherwise NIL without creating any prompt window.

(REMOVEPROMPTWINDOW *MAINWINDOW*)

Detaches the attached prompt window, if any, associated with *MAINWINDOW*, and closes it.

Window properties of attached windows

Windows that are involved in a collection either as a main window or as an attached window have properties stored on them. The only properties that are intended to be set be set by the user are the MINSIZE and MAXSIZE properties. The other properties should be considered read only; they are maintained by the ATTACHEDWINDOW package.

MINSIZE, MAXSIZE: Each should be a dotted pair (width . height) or a function to apply to the window that returns a dotted pair. The numbers are used when the main window is reshaped. The MINSIZE is used to determine the size of the smallest region acceptable during reshaping. Any amount greater than the collective minimum is spread evenly among the windows until each reaches MAXSIZE. Any excess is given to the main window. This algorithm may change as experience is gained. Lobby for your favorite.

Note: This doesn't address the hard problem of overlapping attached windows side to side, for example if window A was attached as [TOP, LEFT] and B as [TOP, RIGHT]. Initially the reshape getregion won't worry about the overlap.

Default MAXSIZE is NIL, which will let the region grow indefinitely.

MAINWINDOW: pointer from attached windows to the main window of the group. This link is not available if the main window is closed.

ATTACHEDWINDOWS: pointer from a window to its attached windows. For functional access to this information, the function ATTACHEDWINDOWS is documented below.

WHEREATTACHED: for attached windows, a list whose first element is the EDGE and whose second element is the POSITIONONEDGE that determine the placement condition for this window.

The TOTOPFN property on attached windows and the properties TOTOPFN , DOSHAPEFN , MOVEFN , CLOSEFN, OPENFN, SHRINKFN, EXPANDFN and CALCULATEREGIONFN contain elements that implement the window manipulation facilities. Care should be used in modifying or replacing these properties.

Notes

A windows can be attached to only one other window. Attaching a window to a second window will detach it from the first.

Attachments can not form loops. That is, a window cannot be attached to itself or to a window that is attached to it. ATTACHWINDOW will generate an error if this is attempted.

Attached windows can have other windows attached to them. Thus, it is possible to attach window A to window B when B is already attached to window C. Similarly, if A has other windows attached to it it can still be attached to B.

Moving the main window will maintain the relationships between windows.

Reshaping the main window will restore the conditions established by the call to ATTACHWINDOW , moving the main window does not. Thus, if A is attached to the top of B and then moved by the user, its new position relative to B will be maintained if B is moved. If B is reshaped, A will be reshaped to the top of B. Additionally, if, while A is moved away from the top of B, C is attached to the top of B, C will position itself above where A used to be.

The attached windows can be closed by themselves. They will be reopened whenever the mainwindow is reshaped. The closefn for the main window breaks the links from the attached windows to it to allow them to be GC'ed. The reopenfn for the main window reestablishes these links. Thus it is possible to reopen a closed, attached window and not have it linked to its mainwindow.

Example of use

```
(ATTACHWINDOW ATWIN MAINWIN 'TOP 'CENTER)
```

will move ATWIN to immediately above MAINWIN and maintain its attachment there.

```
(ATTACHWINDOW NOTHERWIN MAINWIN 'TOP 'CENTER)
```

will move NOTHERWIN to immediately above ATWIN and maintain its attachment there.

Miscellaneous functions

(WINDOWREGION *WINDOW*)

Returns the screen region occupied by *WINDOW* and its attached windows, if it has any.

(WINDOWSIZE *WINDOW*)

Returns the size of *WINDOW* and its attached windows, if it has any.

(MINATTACHEDWINDOWEXTENT *WINDOW*)

Returns the minimum extent (a dotted pair of width and height) of *WINDOW* and its attached windows (if any) will accept. This is determined by using the window property `MINEXTENT` which is either a dotted pair or a function which when `APPLY`ed to the window returns a dotted pair.

`(ATTACHEDWINDOWS` *MAINWINDOW* `)`

Returns the list of windows attached to this window.

`(ALLATTACHEDWINDOWS` *MAINWINDOW* `)`

Returns a list of all of the windows attached to *MAINWINDOW* or attached to a window attached to it.

`(DETACHALLWINDOWS` *MAINWINDOW* `)`

Detaches and closes all windows attached to *MAINWINDOW* .