File created: 16-Jul-99 15:51:36 **{DSK}<project>medley3.5>sources>MENU.;3**

changes to: (FNS UPDATE/MENU/IMAGE)

previous date: 28-Jun-99 17:05:55 {DSK}<project>medley3.5>sources>MENU.;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;


```
(RPAQQ MENUCOMS
       ((COMS                                                    ; window functions
              (FNS MAXMENUITEMHEIGHT MAXMENUITEMWIDTH MENU MENUTITLEFONT ADDMENU DELETEMENU MENUREGION
                   BLTMENUIMAGE ERASEMENUIMAGE DEFAULTMENUHELDFN DEFAULTWHENSELECTEDFN BACKGROUNDWHENSELECTEDFN
                   GETMENUITEM MENUBUTTONFN MENU.HANDLER DOSELECTEDITEM SHOWSHADEDITEMS \AddShade \DelShade
                   \FDECODE/BUTTON MENUITEMREGION \MENUITEMLABEL \MENUSUBITEMS CHECK/MENU/IMAGE PPROMPT2
                   UPDATE/MENU/IMAGE \MAKE.ITEMS.VERT.ORDER \SHOWMENULABEL \POSITION.MENU.IMAGE
                   \SMASHMENUIMAGEONRESET CLOSE.PROCESS.MENU DEFAULTSUBITEMFN GETMENUPROP PUTMENUPROP
                   WAKE.MY.PROCESS \INVERTITEM \MENU.ITEM.SELECT \MENU.ITEM.DESELECT \ItemNumber \BOXITEM
                   NESTED.SUBMENU NESTED.SUBMENU.POS WFROMMENU)
              (BITMAPS MENUSUBITEMMARK)
              (INITVARS (MENUFONT (FONTCREATE 'HELVETICA 10)))
              (DECLARE%: DONTCOPY (MACROS MENU.HELDSTATE.RESET MENU.PRIN2.FLG)))
        (COMS                                                    ; scrolling menu functions and utilities
              (FNS MENUREPAINTFN))
        (COMS                                                    ; misc utility fns.
              (FNS MAXSTRINGWIDTH CENTEREDPRIN1 CENTERPRINTINREGION CENTERPRINTINAREA STRICTLY/BETWEEN))
        (COMS                                                    ; examples of use.
              (FNS UNREADITEM TYPEINMENU SHADEITEM RESHADEITEM MOST/VISIBLE/OPERATION %#BITSON BUTTONPANEL
                   BUTTONPANEL/SELECTION/FN GETSELECTEDITEMS)
              (VARS EDITCMDS MENUHELDWAIT)
              (CONSTANTS (BITSPERSHADE 16))
              (GLOBALVARS MENUSELECTSHADE)
              (VARS MENUSELECTSHADE)
              (FNS MENUDESELECT MENUSELECT))
        (DECLARE%: DOCOPY DONTEVAL@LOAD (VARS (MENUFONT)))
        (GLOBALVARS MENUFONT MENUHELDWAIT)
        (RECORDS MENU)))
```

;; window functions

```
(DEFINEQ
```

(MAXMENUITEMHEIGHT

```
  [LAMBDA (MENU)                                                 (* kbr%: "27-May-85 13:31")
                                                                 (* returns the height of the largest menu item label in the menu
                                                                 MENU.)
    (PROG (FONTHEIGHT LABEL ANSWER)
          (SETQ FONTHEIGHT (FONTPROP (fetch (MENU MENUFONT) of MENU)
                                     'HEIGHT))
          (SETQ ANSWER 0)
          [for ITEM in (fetch (MENU ITEMS) of MENU) do (SETQ LABEL (\MENUITEMLABEL ITEM))
                                                       (SETQ ANSWER (IMAX ANSWER (COND
                                                                                   ((BITMAPP LABEL)
                                                                                    (fetch (BITMAP BITMAPHEIGHT)
                                                                                       of LABEL))
                                                                                   (T FONTHEIGHT]
          (RETURN ANSWER])
```

(MAXMENUITEMWIDTH

```
  [LAMBDA (MENU NOSUBITEMMARK)                                   (* bvm%: "14-Oct-86 13:04")
                                                                 (* returns the width of the largest menu item label in the menu
                                                                 MENU.)
    (DECLARE (GLOBALVARS MENUSUBITEMMARK))
    (for I in (fetch (MENU ITEMS) of MENU) bind (ANSWER _ 0)
                                               (FONT _ (fetch (MENU MENUFONT) of MENU))
                                               (P2FLG _ (MENU.PRIN2.FLG MENU))
                                               LABEL SUBITEMS
       do (SETQ LABEL (\MENUITEMLABEL I))
          [SETQ SUBITEMS (COND
                             ((NOT NOSUBITEMMARK)
                              (\MENUSUBITEMS MENU I]
          [SETQ ANSWER (IMAX ANSWER (IPLUS (COND
                                             ((BITMAPP LABEL)
                                              (fetch (BITMAP BITMAPWIDTH) of LABEL))
                                             (T (IPLUS (STRINGWIDTH LABEL FONT P2FLG NIL)
                                                       2)))
                                           (COND
                                             (SUBITEMS (BITMAPWIDTH MENUSUBITEMMARK))
                                             (T 0]
```

```
      finally (RETURN ANSWER])
```

(MENU

```
  [LAMBDA (MENU POSITION RELEASECONTROLFLG NESTEDFLG)          ; Edited 21-Jun-88 19:00 by jds
    (DECLARE (LOCALVARS . T))

    ;; puts a menu on the screen and waits for the user to select one of the items

    (\DTEST MENU 'MENU)
    (PROG (IMAGE SELVAL DSP)                                  ; make sure the image is a window
          [SETQ IMAGE (COND
                         ((NOT (EQ POSITION 'INPLACE))
                          (\POSITION.MENU.IMAGE MENU POSITION))
                         (T (fetch (MENU IMAGE) of MENU]
          (SETQ DSP (WINDOWPROP IMAGE 'DSP))
          [SETQ SELVAL (RESETLST
                           (RESETSAVE (OPENW IMAGE)
                                  (LIST 'CLOSEW IMAGE))
                           (COND
                              (RELEASECONTROLFLG (PROG (MVAL)
                                                       (WINDOWPROP IMAGE 'MENUPROCESS (THIS.PROCESS))
                                                       (WINDOWPROP IMAGE 'CLOSEFN 'CLOSE.PROCESS.MENU)
                                                       (WINDOWPROP IMAGE 'BUTTONEVENTFN 'WAKE.MY.PROCESS)
                                                 LP    (TOTOPW IMAGE)
                                                       (OR [NOT (EQ T (SETQ MVAL (BLOCK 200]
                                                           (RETURN NIL))
                                                       (GETMOUSESTATE)
                                                                 ; if mouse state is up, then someone came into the window with
                                                                 ; the mouse down.  Ignore it.
                                                       (OR (MOUSESTATE (OR LEFT RIGHT MIDDLE))
                                                           (GO LP))
                                                                 ; MVAL will be NIL only if the user clicked up outside the window
                                                       (OR (SETQ MVAL (MENU.HANDLER MENU DSP NIL T NESTEDFLG))
                                                           (GO LP))
                                                       (RETURN MVAL)))
                                    (T (MENU.HANDLER MENU DSP T T NESTEDFLG))))]
                                                                 ; evaluate menu form after image has been taken down.
          (RETURN (COND
                     (NESTEDFLG SELVAL)
                     (SELVAL (DOSELECTEDITEM MENU (CAR SELVAL)
                                    (CDR SELVAL])
```

(MENUTITLEFONT

```
  [LAMBDA (MENU SCREEN)                                       (* kbr%: " 2-Sep-85 14:35")
                                                              (* returns the title font for a menu.)
    (PROG (TITLEFONT)
          [COND
             ((NULL SCREEN)
              (COND
                 [(type? WINDOW (fetch (MENU IMAGE) of MENU))
                  (SETQ SCREEN (fetch (WINDOW SCREEN) of (fetch (MENU IMAGE) of MENU]
                 (T (SETQ SCREEN LASTSCREEN]
          (RETURN (COND
                     ((NULL (SETQ TITLEFONT (fetch (MENU MENUTITLEFONT) of MENU)))
                                                              (* use the window title font)
                      (DSPFONT NIL (fetch (SCREEN SCTITLEDS) of SCREEN)))
                     ((EQ TITLEFONT T)                        (* use the menu item font)
                      (fetch (MENU MENUFONT) of MENU))
                     ((FONTP (\COERCEFONTDESC TITLEFONT 'DISPLAY T)))
                     (T (DSPFONT NIL (fetch (SCREEN SCTITLEDS) of SCREEN])
```

(ADDMENU

```
  [LAMBDA (ADDEDMENU W POSITION DONTOPENFLG)                  (* kbr%: "24-Jan-86 18:00")

          (* adds a menu to a window. If W is not given, it is created; sized a necessary.)

    (OR (TYPENAMEP ADDEDMENU 'MENU)
        (\ILLEGAL.ARG ADDEDMENU))
    (PROG (IMAGEWIDTH IMAGEHEIGHT SCREEN)
          (SETQ IMAGEWIDTH (fetch (MENU IMAGEWIDTH) of ADDEDMENU))
          (SETQ IMAGEHEIGHT (fetch (MENU IMAGEHEIGHT) of ADDEDMENU))
                                                              (* put menu at POSITION if argument, otherwise its stored
                                                              position, otherwise at cursorposition)
          [COND
             ((POSITIONP POSITION))
             ((SETQ POSITION (fetch (MENU MENUPOSITION) of ADDEDMENU)))
             (W                                               (* if a window is given, put it in the lower left corner.)
                (SETQ POSITION (create POSITION
                                       XCOORD _ 0
                                       YCOORD _ 0)))
             (T (SETQ POSITION (create POSITION
                                       XCOORD _ LASTMOUSEX
                                       YCOORD _ LASTMOUSEY]
          [COND
             ((WINDOWP W)
```

```
        (* adding to an existing window. To avoid partial images when window is partly off the screen, this case could close window
        then blt to save region then reopen window.)
                                                                    (* locate menu grid in MENU.)
            (replace (REGION LEFT) of (fetch (MENU MENUGRID) of ADDEDMENU) with (IPLUS (fetch (POSITION XCOORD)
                                                                                         of POSITION)
                                                                                  (fetch (MENU
                                                                                             MENUOUTLINESIZE
                                                                                             )
                                                                                         of ADDEDMENU)))
            (replace (REGION BOTTOM) of (fetch (MENU MENUGRID) of ADDEDMENU) with (IPLUS (fetch (POSITION YCOORD)
                                                                                            of POSITION)
                                                                                     (fetch (MENU
                                                                                                MENUOUTLINESIZE
                                                                                                )
                                                                                            of ADDEDMENU)))
                                                                    (* Blt image into Window.)
            (BLTMENUIMAGE ADDEDMENU (WINDOWPROP W 'DSP)
                  DONTOPENFLG))
           (T                                                       (* have to create new window. Put position at Origin.)
              (SETQ SCREEN (COND
                             ((type? SCREEN W)
                              W)
                             (T LASTSCREEN)))
              (SETQ W (CREATEWFROMIMAGE (BITMAPCOPY (CHECK/MENU/IMAGE ADDEDMENU NIL SCREEN))
                            SCREEN))
              (MOVEW W (fetch (POSITION XCOORD) of POSITION)
                     (fetch (POSITION YCOORD) of POSITION)
                     SCREEN)
              (SHOWSHADEDITEMS ADDEDMENU W)
              (SETQ POSITION (create POSITION
                                     XCOORD _ 0
                                     YCOORD _ 0))                   (* locate menu grid in MENU.)
              (replace (REGION LEFT) of (fetch (MENU MENUGRID) of ADDEDMENU) with (fetch (MENU MENUOUTLINESIZE)
                                                                                          of ADDEDMENU))
              (replace (REGION BOTTOM) of (fetch (MENU MENUGRID) of ADDEDMENU) with (fetch (MENU MENUOUTLINESIZE)
                                                                                           of ADDEDMENU))
              (OR DONTOPENFLG (OPENW W]

        (* put MENUBUTTONFN in CURSORINFN so it will get called if button is down and moves into W.)

        (WINDOWPROP W 'CURSORINFN (FUNCTION MENUBUTTONFN))          (* Set ButtonEventFn to activate menu selection.)
        (WINDOWPROP W 'BUTTONEVENTFN (FUNCTION MENUBUTTONFN))
        (WINDOWPROP W 'CURSORMOVEDFN (FUNCTION MENUBUTTONFN))       (* put ADDEDMENU on USERDATA so MENUBUTTONFN can
    get at it.)
        (WINDOWADDPROP W 'MENU ADDEDMENU)
        (WINDOWADDPROP W 'REPAINTFN (FUNCTION MENUREPAINTFN))
        [COND
           ((NULL (fetch (MENU WHENSELECTEDFN) of ADDEDMENU))

        (* make the default selection function call EVAL.AS.PROCESS instead of EVAL so it won't tie up background.)

            (replace (MENU WHENSELECTEDFN) of ADDEDMENU with (FUNCTION BACKGROUNDWHENSELECTEDFN]
        [COND
           ((NOT (SUBREGIONP (DSPCLIPPINGREGION NIL W)
                             (MENUREGION ADDEDMENU)))               (* if the menu didn't fit, make it scrollable.)
            (WINDOWPROP W 'SCROLLFN (FUNCTION SCROLLBYREPAINTFN))
            (EXTENDEXTENT W (MENUREGION ADDEDMENU]
        (RETURN W])

(DELETEMENU
  [LAMBDA (MENU CLOSEFLG FROMWINDOW)                                (* rrb " 6-Apr-84 11:55")

        (* deletes a menu from its window. If it is the only menu in the window and CLOSEFLG is non-NIL, it closes the window.)

    (OR (TYPENAMEP MENU 'MENU)
        (\ILLEGAL.ARG MENU))
    (PROG ([W (COND
                ((type? WINDOW FROMWINDOW)
                 FROMWINDOW)
                (T (WFROMMENU MENU]
          OTHERMENUS)                                               (* see if menu is in a window.)
        (OR W (RETURN))                                             (* Blt image out of Window.)
        (ERASEMENUIMAGE MENU W)
        [COND
           [[NULL (CDR (SETQ OTHERMENUS (WINDOWPROP W 'MENU]        (* last menu)
            (OR (EQ MENU (CAR OTHERMENUS))
                (ERROR "MENU not correctly in W"))
            (WINDOWPROP W 'MENU NIL)
            (COND
               (CLOSEFLG (CLOSEW W]
           (T (WINDOWPROP W 'MENU (DREMOVE MENU OTHERMENUS]
        (COND
           ((EQ (fetch (MENU WHENSELECTEDFN) of MENU)
                (FUNCTION BACKGROUNDWHENSELECTEDFN))
```

```
        (* return the default selection function call EVAL instead of EVAL.AS.PROCESS so it will return the correct value.)

            (replace (MENU WHENSELECTEDFN) of MENU with NIL)))
        (RETURN W])
```

(MENUREGION

```
  [LAMBDA (MENU)                                                    (* rrb " 9-FEB-82 09:37")
                                                                    (* returns the region covered by the image of a MENU)
                                                                    (* calls IMAGEWIDTH first so that it will calculate an image if
                                                                    none exists yet.)
    (create REGION
           WIDTH _ (fetch (MENU IMAGEWIDTH) of MENU)
           HEIGHT _ (fetch (MENU IMAGEHEIGHT) of MENU)
           LEFT _ (fetch (MENU MENUREGIONLEFT) of MENU)
           BOTTOM _ (fetch (MENU MENUREGIONBOTTOM) of MENU])
```

(BLTMENUIMAGE

```
  [LAMBDA (MENU WIN DONTOPEN)                                       (* hdj "12-Apr-85 14:05")
                                                                    (* Displays a menu image at its position on DS.)
    (PROG ([SRC (COND
                ((WINDOWP (fetch IMAGE of MENU))
                 (fetch (WINDOW SAVE) of (fetch (MENU IMAGE) of MENU)))
                (T (fetch IMAGE of MENU]
           (DSTWIN (\INSUREWINDOW WIN)))
          (RETURN (COND
                    [(AND DONTOPEN (NOT (OPENWP DSTWIN)))       (* leave the window closed)
                     (PROG ((BORDER (WINDOWPROP DSTWIN 'BORDER))
                            (CR (DSPCLIPPINGREGION NIL DSTWIN)))
                           (RETURN (PROG1 (BITBLT SRC 0 0 (fetch (WINDOW SAVE) of DSTWIN)
                                                  (IPLUS BORDER (fetch (MENU MENUREGIONLEFT) of MENU))
                                                  (IPLUS BORDER (fetch (MENU MENUREGIONBOTTOM) of MENU))
                                                  (IMIN (BITMAPWIDTH SRC)
                                                        (fetch (REGION WIDTH) of CR))
                                                  (IMIN (BITMAPHEIGHT SRC)
                                                        (fetch (REGION HEIGHT) of CR)))
                                          (SHOWSHADEDITEMS MENU DSTWIN]
                    (T (PROG1 (BITBLT SRC NIL NIL DSTWIN (fetch (MENU MENUREGIONLEFT) of MENU)
                                      (fetch (MENU MENUREGIONBOTTOM) of MENU))
                              (SHOWSHADEDITEMS MENU DSTWIN])
```

(ERASEMENUIMAGE

```
  [LAMBDA (MENU W)                                                  (* rrb "19-MAR-82 10:26")
```

(* removes the menu image from a window by clearing the place it used to occupy.
Image may be different from stored image because user may have shaded an item.)

```
    (BITBLT NIL NIL NIL (WINDOWPROP W 'DSP)
           (IDIFFERENCE (fetch (REGION LEFT) of (fetch (MENU MENUGRID) of MENU))
                  (fetch MENUOUTLINESIZE of MENU))
           (IDIFFERENCE (fetch (REGION BOTTOM) of (fetch (MENU MENUGRID) of MENU))
                  (fetch MENUOUTLINESIZE of MENU))
           (fetch (MENU IMAGEWIDTH) of MENU)
           (fetch (MENU IMAGEHEIGHT) of MENU)
           'TEXTURE
           'REPLACE])
```

(DEFAULTMENUHELDFN

```
  [LAMBDA (ITEM)                                                    (* rrb "23-NOV-81 12:41")
    (COND
       ((AND (LISTP ITEM)
             (CADDR ITEM))
        (PROMPTPRINT (CADDR ITEM)))
       (T (PROMPTPRINT "Will select this item when you release the button."])
```

(DEFAULTWHENSELECTEDFN

```
  [LAMBDA (ITEM FROMMENU BUTTON)                                    (* rrb "24-Feb-84 15:01")
                                                                    (* default Menu handler)
    (COND
       ((AND (LISTP ITEM)
             (LISTP (CDR ITEM)))
        (STKEVAL (OR (STKPOS 'MENU)
                     'MENUBUTTONFN)
                 (CADR ITEM)
                 T))
       (T ITEM])
```

(BACKGROUNDWHENSELECTEDFN

```
  [LAMBDA (ITEM FROMMENU BUTTON)                                    (* rrb "27-AUG-82 10:17")
```

(* default Menu handler for fixed menus. It differs from DEFAULTWHENSELECTEDFN by calling EVAL.AS.PROCESS instead of EVAL.)

```
    (COND
       [(LISTP ITEM)
        (COND
           ((CDR ITEM)
            (EVAL.AS.PROCESS (CADR ITEM)))
           (T (CAR ITEM]
       (T ITEM])
```

(**GETMENUITEM**

```
  [LAMBDA (MENU XGRID YGRID)                                    (* rrb "31-JUL-81 07:31")
                                                                (* returns the menu item that is in grid location {XGRID,YGRID}.)
    (CAR (FNTH (fetch (MENU ITEMS) of MENU)
               (IPLUS (ITIMES (SUB1 (IDIFFERENCE (fetch MENUROWS of MENU)
                                                 YGRID))
                              (fetch MENUCOLUMNS of MENU))
                      XGRID 1])
```

(**MENUBUTTONFN**

```
  [LAMBDA (W)                                                   (* rrb "18-APR-83 17:19")
    (COND
       [(LASTMOUSESTATE (OR LEFT MIDDLE RIGHT))
        (TOTOPW W)
        (bind SELECTION for MENU in (WINDOWPROP W 'MENU) when [SETQ SELECTION (MENU.HANDLER MENU
                                                                                            (WINDOWPROP W 'DSP]
           do (DOSELECTEDITEM MENU (CAR SELECTION)
                              (CDR SELECTION]
       (T                                                       (* must have been button up or a cursor move event.)
          NIL])
```

(**MENU.HANDLER**

```
  [LAMBDA (MENU DSP KEEPCONTROLIFOUTFLG CHANGEOFFSETFLG NESTEDFLG)
                                                                ; Edited 28-Dec-93 12:56 by sybalsky
    (DECLARE (SPECVARS SUBMENU MOVEDLEFT))
    ;; handles details of watching mouse for menus.
    (RESETLST
        (RESETSAVE NIL (LIST '\SMASHMENUIMAGEONRESET MENU))
        [PROG (ITEM SUBITEMS SUBMENURESULT OLDBOXX OLDBOXY BOXX BOXY HELDSTATE (MOUSEDOWN (LASTMOUSESTATE
                                                                                           (NOT UP)))
                    (MOVEDLEFT "NESTED")
                    (LASTBUTTONSTATE LASTMOUSEBUTTONS)
                    (MGRIDSPEC (fetch (MENU MENUGRID) of MENU))
                    (HOLDTIMER (SETUPTIMER MENUHELDWAIT))
                    (HELDFN (fetch (MENU WHENHELDFN) of MENU))
                    (NROWS (fetch (MENU MENUROWS) of MENU))
                    (NCOLUMNS (fetch (MENU MENUCOLUMNS) of MENU))
                    SUBMENUWINDOW SUBMENU (LOCALMENUHELDWAIT (OR (FIXP MENUHELDWAIT)
                                                                 1200)))
          ;; SUBMENUWINDOW is used to hold the window of the submenu and to indicate if a submenu is up.  SUBMENU is to hold onto the
          ;; submenu so it can be passed to MENU if it is entered.
              [COND
                 ((AND MOUSEDOWN (STRICTLY/BETWEEN (SETQ BOXY (GRIDYCOORD (LASTMOUSEY DSP)
                                                                           MGRIDSPEC))
                                                   -1 NROWS)
                       (STRICTLY/BETWEEN (SETQ BOXX (GRIDXCOORD (LASTMOUSEX DSP)
                                                                MGRIDSPEC))
                                         -1 NCOLUMNS))
                  ;; make a special check for when the last state was down and save the information about which item that was over.
                  (SETQ SUBMENUWINDOW (\MENU.ITEM.SELECT (SETQ OLDBOXX BOXX)
                                                         (SETQ OLDBOXY BOXY)
                                                         MENU DSP]
              (RETURN (COND
                         ([SETQ ITEM
                           (ERSETQ (until (COND
                                             (MOUSEDOWN                 ; if mouse has been down, process it
                                                    (MOUSESTATE UP))
                                             ((MOUSESTATE (NOT UP))
                                                                        ; mouse hasn't been down but just went down.
                                              [COND
                                                 ((AND (NULL KEEPCONTROLIFOUTFLG)
                                                       (LASTMOUSESTATE RIGHT))
                                                  (DOWINDOWCOM (WHICHW LASTMOUSEX LASTMOUSEY)))
                                                 (T (SETQ MOUSEDOWN T)
                                                    (COND
                                                       (OLDBOXX     ; switch between boxing to flipping items.
                                                              (\BOXITEM OLDBOXX OLDBOXY MENU DSP)
                                                              (SETQ SUBMENUWINDOW (\MENU.ITEM.SELECT OLDBOXX
                                                                                                     OLDBOXY MENU DSP]
                                              NIL))
                                    do [COND
                                          [[OR (AND SUBMENUWINDOW (INSIDE? (fetch (WINDOW REG) of SUBMENUWINDOW
```

```
                                                    )
                                                LASTMOUSEX LASTMOUSEY))
                        (AND SUBMENU (EQ (GRIDYCOORD (LASTMOUSEY DSP)
                                                     MGRIDSPEC)
                                         OLDBOXY)
                             (IGEQ (GRIDXCOORD (LASTMOUSEX DSP)
                                               MGRIDSPEC)
                                   (COND
                                      (OLDBOXX (PLUS OLDBOXX 1))
                                      (T NCOLUMNS]
```

;; either the cursor moved into or already was inside of the submenu, or it rolled out the right side of
;; an item that has non-popup submenu items. It could already be inside if the submenu came up
;; over the menu. This can lead to funny interactions of submenus popping up and automatically
;; being selected when near the right edge of the screen but I can't think of anything better and this is
;; at least consistent.

```
                  (COND                    ; call submenu and process result.
                     ((EQ (SETQ SUBMENURESULT
                                (MENU SUBMENU (COND
                                                 (SUBMENUWINDOW 'INPLACE)
                                                 (T (NESTED.SUBMENU.POS MENU
                                                                        (GETMENUITEM MENU OLDBOXX
                                                                                     OLDBOXY)
                                                                        DSP)))
                                              NIL T))
                          MOVEDLEFT)  ; user moved back to left without selecting anything.
```

;; reopen the submenu which was closed by MENU on its way out. This would be cleaner to have
;; MENU not close it but this is hard to error set protect correctly.

```
                        (AND SUBMENUWINDOW (OPENW SUBMENUWINDOW))
                        (SETQ SUBMENURESULT NIL))
                       (T                  ; selected something from submenu
                          (COND
                             (MOUSEDOWN (\MENU.ITEM.DESELECT OLDBOXX OLDBOXY MENU DSP))
                             (T (\BOXITEM OLDBOXX OLDBOXY MENU DSP)))
                          (MENU.HELDSTATE.RESET OLDBOXX OLDBOXY)
                          (SETQ OLDBOXX)
                          (GO OUT]
                [(AND (STRICTLY/BETWEEN (SETQ BOXY (GRIDYCOORD (LASTMOUSEY DSP)
                                                               MGRIDSPEC))
                                        -1 NROWS)
                      (STRICTLY/BETWEEN (SETQ BOXX (GRIDXCOORD (LASTMOUSEX DSP)
                                                               MGRIDSPEC))
                                        -1 NCOLUMNS))
                                       ; BOXX and BOXY hold the number of the box pointed at.
                 (COND
                    ((OR (NEQ BOXX OLDBOXX)
                         (NEQ BOXY OLDBOXY))
                                       ; selected item has changed.
                                       ; deselect old item if there was one.
                     [COND
                        (OLDBOXX (COND
                                    (MOUSEDOWN (\MENU.ITEM.DESELECT OLDBOXX OLDBOXY
                                                                    MENU DSP))
                                    (T (\BOXITEM OLDBOXX OLDBOXY MENU DSP)))
                                 (MENU.HELDSTATE.RESET OLDBOXX OLDBOXY))
                        (T (SETQ HOLDTIMER (SETUPTIMER LOCALMENUHELDWAIT HOLDTIMER]
                                       ; invert new item
                     (COND
                        (MOUSEDOWN (SETQ SUBMENUWINDOW (\MENU.ITEM.SELECT BOXX BOXY
                                                                          MENU DSP)))
                        (T (\BOXITEM BOXX BOXY MENU DSP)))
                     (SETQ OLDBOXX BOXX)
                     (SETQ OLDBOXY BOXY))
                    ((AND HELDFN (NULL HELDSTATE)
                          (TIMEREXPIRED? HOLDTIMER))
                                       ; same button in same region for MENUHELDWAIT milliseconds.
                     (APPLY* HELDFN (GETMENUITEM MENU OLDBOXX OLDBOXY)
                             MENU
                             (\FDECODE/BUTTON LASTBUTTONSTATE))
                     (SETQ HELDSTATE T]
                (T                     ; cursor moved out of the menu, deselect any selected items
                   (COND
                      (OLDBOXX (COND
                                  (MOUSEDOWN (\MENU.ITEM.DESELECT OLDBOXX OLDBOXY
                                                                  MENU DSP))
                                  (T (\BOXITEM OLDBOXX OLDBOXY MENU DSP)))
                               (MENU.HELDSTATE.RESET OLDBOXX OLDBOXY)
                               (SETQ OLDBOXX)))
                   (COND
                      ((AND NESTEDFLG BOXX (IGREATERP 0 BOXX)
                            (ILESSP (LASTMOUSEX DSP)
                                    0))
```

;; make sure the mouse has moved all the way past the left including its border and outline size. We
;; know it has to be a popup menu that will have 0 as its left edge.

```
                                       ; if this is a nested call and the user moved to the left, return
                                       ; indicator of this.
```

```
                                        (RETURN MOVEDLEFT))
                                       ((NOT KEEPCONTROLIFOUTFLG)
                                        (RETURN]
                                 (COND
                                    ((NEQ LASTBUTTONSTATE (SETQ LASTBUTTONSTATE LASTMOUSEBUTTONS))
                                                                        ; reset held timer
                                     (MENU.HELDSTATE.RESET OLDBOXX OLDBOXX)))
                              finally                                   ; turn off inverse image.  and call whenunheldfn is necessary.
                                    OUT
                                    (COND
                                       (OLDBOXX (COND
                                                   (MOUSEDOWN (\MENU.ITEM.DESELECT OLDBOXX OLDBOXY MENU
                                                                                   DSP))
                                                   (T (\BOXITEM OLDBOXX OLDBOXY MENU DSP)))
                                                (MENU.HELDSTATE.RESET OLDBOXX OLDBOXY)))

                                    ;; if called for, change the menu offset so the menu will come up in the same place relative to
                                    ;; the cursor next time.

                                    [COND
                                       ((AND CHANGEOFFSETFLG OLDBOXX)
                                        (SELECTQ (fetch (MENU CHANGEOFFSETFLG) of MENU)
                                             ((Y NIL))
                                             (replace (POSITION XCOORD) of (fetch (MENU MENUOFFSET)
                                                                                  of MENU)
                                                with (LASTMOUSEX DSP)))
                                        (SELECTQ (fetch (MENU CHANGEOFFSETFLG) of MENU)
                                             ((X NIL))
                                             (replace (POSITION YCOORD) of (fetch (MENU MENUOFFSET)
                                                                                  of MENU)
                                                with (LASTMOUSEY DSP]
                                    (RETURN (COND
                                               (SUBMENURESULT)
                                               (OLDBOXX (CONS (GETMENUITEM MENU OLDBOXX OLDBOXY)
                                                              (\FDECODE/BUTTON LASTBUTTONSTATE]
                                                                        ; no error
                          (RETURN (CAR ITEM)))
                         (T                                             ; user ^E --- reset the menu selection.  ^d is handled by
                                                                        ; RESETLST.
                            [COND
                               (OLDBOXX (COND
                                           (MOUSEDOWN (\MENU.ITEM.DESELECT OLDBOXX OLDBOXY MENU DSP))
                                           (T (\BOXITEM OLDBOXX OLDBOXY MENU DSP]
                            (ERROR!])])
```

(DOSELECTEDITEM

```
[LAMBDA (MENU ITEM BUTTON)                                              ; Edited  9-Apr-94 00:43 by rmk:
                                                                        (* rrb "28-JAN-82 16:33")
  (CL:UNLESS [EQ '*DUMMYITEM* (CAR (LISTP (CDR (LISTP ITEM]
      (APPLY* (OR (fetch WHENSELECTEDFN of MENU)
                  (FUNCTION DEFAULTWHENSELECTEDFN))
              ITEM MENU BUTTON))])
```

(SHOWSHADEDITEMS

```
[LAMBDA (MENU DSP)                                                      (* edited%: "31-Dec-00 19:10")
```

(* shades a menu item with a background shade. DS/W if provided is the displaystream to use.)

```
  (PROG ((ALLITEMS (fetch (MENU ITEMS) of MENU))
         SHADE ITEM ITEMREGION ANYSUBITEMS)
        (SETQ ANYSUBITEMS (for ITEM in ALLITEMS thereis (\MENUSUBITEMS MENU ITEM)))
        (for ITEMDESCR in (fetch (MENU SHADEDITEMS) of MENU)
           do [SETQ ITEM (CAR (NTH ALLITEMS (CAR ITEMDESCR]
              (SETQ SHADE (CDR ITEMDESCR))
              (SETQ ITEMREGION (MENUITEMREGION ITEM MENU))
              (OR ITEMREGION (RETURN))                                  (* if the menu is not in a window don't do anything.)
              [COND
                 (ANYSUBITEMS (replace (REGION WIDTH) of ITEMREGION with (DIFFERENCE (fetch (REGION WIDTH)
                                                                                          of ITEMREGION)
                                                                                     (BITMAPWIDTH MENUSUBITEMMARK]
              (RESHADEITEM ITEM ITEMREGION MENU SHADE DSP])
```

(\AddShade

```
[LAMBDA (ITEM SHADE MENU)                                               ; Edited 29-Jul-87 14:56 by scp
  (PROG ((INDEX (\ItemNumber ITEM (fetch (MENU ITEMS) of MENU)))
         (SHADEDITEMS (fetch (MENU SHADEDITEMS) of MENU)))
        (if (NULL INDEX)
            then (RETURN))
        (for SHADEDITEM in SHADEDITEMS do (if (EQ (CAR SHADEDITEM)
                                                  INDEX)
                                              then (RPLACD SHADEDITEM SHADE)
                                                   (RETURN))
           finally (SETQ SHADEDITEMS (CONS (CONS INDEX SHADE)
                                           SHADEDITEMS)))
```

;; (if (EQ SHADE 0) then (* we take shade = 0 to mean 'unshade') (SETQ SHADEDITEMS (\DelShade INDEX SHADEDITEMS)) else (for

```
;; SHADEDITEM in SHADEDITEMS do (if (EQ (CAR SHADEDITEM) INDEX) then (RPLACD SHADEDITEM SHADE) (RETURN)) finally (SETQ
;; SHADEDITEMS (CONS (CONS INDEX SHADE) SHADEDITEMS))))
      (replace (MENU SHADEDITEMS) of MENU with SHADEDITEMS])
```

(\DelShade

```
[LAMBDA (KEY LIST)                                          (* hdj " 4-Sep-85 14:42")
  (COND
     ((NULL LIST)
      NIL)
     ((EQ KEY (CAAR LIST))
      (CDR LIST))
     (T (CONS (CAR LIST)
              (\DelShade KEY (CDR LIST])
```

(\FDECODE/BUTTON

```
[LAMBDA (BUTTONSTATE)                                       (* rrb " 9-JAN-82 13:59")
                                                            (* return RED BLUE or YELLOW from a button state.)
  (SELECTQ BUTTONSTATE
       (4 'LEFT)
       (2 'RIGHT)
       (1 'MIDDLE)
       NIL])
```

(MENUITEMREGION

```
[LAMBDA (ITEM IMENU)                                        ; Edited  8-Jul-93 19:26 by sybalskY:MV:ENVOS
                                                            ; returns the region for ITEM in IMENU.  NIL if ITEM isn't in
                                                            ; IMENU.
  (CHECK/MENU/IMAGE IMENU)                                  ; COMPUTE MENUCOLUMNS ETC
  (PROG (ITEMNUMBER (ITEMS (fetch (MENU ITEMS) of IMENU))
                (GRIDSPEC (fetch (MENU MENUGRID) of IMENU))
                (BORDER (fetch (MENU MENUBORDERSIZE) of IMENU)))
        [SETQ ITEMNUMBER (IDIFFERENCE (LENGTH ITEMS)
                                  (LENGTH (OR (FMEMB ITEM ITEMS)
                                              (for ITEMTAIL on ITEMS
                                                 when (EQ (CAR (LISTP (CAR ITEMTAIL)))
                                                          ITEM)
                                                 do (RETURN ITEMTAIL))
                                              (RETURN]
        (RETURN (create REGION
                        LEFT _ (IPLUS (fetch (REGION LEFT) of GRIDSPEC)
                                      (ITIMES (IREMAINDER ITEMNUMBER (fetch (MENU MENUCOLUMNS) of IMENU))
                                              (fetch (REGION WIDTH) of GRIDSPEC))
                                      BORDER)
                        BOTTOM _ (IPLUS (fetch (REGION BOTTOM) of GRIDSPEC)
                                        (ITIMES [SUB1 (IDIFFERENCE (fetch (MENU MENUROWS) of IMENU)
                                                                   (IQUOTIENT ITEMNUMBER (fetch (MENU MENUCOLUMNS)
                                                                                                of IMENU]
                                                                   (fetch (REGION HEIGHT) of GRIDSPEC))
                                        BORDER)
                        WIDTH _ (IDIFFERENCE (fetch (REGION WIDTH) of GRIDSPEC)
                                             (ITIMES 2 BORDER))
                        HEIGHT _ (IDIFFERENCE (fetch (REGION HEIGHT) of GRIDSPEC)
                                              (ITIMES 2 BORDER])
```

(\MENUITEMLABEL

```
[LAMBDA (ITEM)                                              (* rrb "21-AUG-81 08:13")
                                                            (* returns the item label of an item.)
  (COND
     ((LISTP ITEM)
      (CAR ITEM))
     (T ITEM])
```

(\MENUSUBITEMS

```
[LAMBDA (MENU ITEM)                                         (* rrb "29-Dec-83 09:54")
  (APPLY* (OR (fetch (MENU SUBITEMFN) of MENU)
              (FUNCTION DEFAULTSUBITEMFN))
          MENU ITEM])
```

(CHECK/MENU/IMAGE

```
[LAMBDA (MENU MAKEWINDOWFLG SCREEN)                         (* kbr%: " 5-Sep-85 20:31")

      (* returns menus image, creating one if necessary. The image field will be a WINDOW for popup menus.)

  (PROG (IMAGE DSP WINDOW)
        (OR (type? MENU MENU)
            (\ILLEGAL.ARG MENU))
        (SETQ IMAGE (fetch (MENU IMAGE) of MENU))
        [OR SCREEN (SETQ SCREEN (COND
                                  ((type? WINDOW IMAGE)
                                   (fetch (WINDOW SCREEN) of IMAGE))
                                  (T LASTSCREEN]
```

```
          [COND
             ((OR (NULL IMAGE)
                  (NOT (EQ (fetch (WINDOW SCREEN) of IMAGE)
                           SCREEN)))                              (* Switched screens. *)
              (UPDATE/MENU/IMAGE MENU SCREEN)
              (SETQ IMAGE (fetch (MENU IMAGE) of MENU]
          (COND
             (MAKEWINDOWFLG (COND
                                ((type? WINDOW IMAGE)
                                 (UPDATEWFROMIMAGE IMAGE))
                                (T (SETQ IMAGE (CREATEWFROMIMAGE IMAGE SCREEN))
                                   (replace (MENU IMAGE) of MENU with IMAGE)))
                            (SETQ DSP (fetch (WINDOW DSP) of IMAGE))      (* set the offset in the display stream to agree with the region.)
                            (DSPXOFFSET (fetch (WINDOW WBORDER) of IMAGE)
                                   DSP)
                            (DSPYOFFSET (fetch (WINDOW WBORDER) of IMAGE)
                                   DSP)))
          (RETURN (COND
                     ((type? BITMAP IMAGE)
                      IMAGE)
                     (T (fetch (WINDOW SAVE) of IMAGE])
```

(PPROMPT2

```
  [LAMBDA (ITEM)                                                  (* rrb "17-NOV-81 14:09")
                                                                  (* prints the second element of ITEM in the prompt window.)
    (COND
       ((AND (LISTP ITEM)
             (CADR ITEM))
        (PROMPTPRINT (CADR ITEM])
```

(UPDATE/MENU/IMAGE

```
  [LAMBDA (MNU SCREEN)                                            ; Edited 16-Jul-99 15:51 by rmk:
                                                                  ; Edited 10-Dec-93 16:01 by sybalsky
                                                                  ; recomputes the menu image from its labels.
    (PROG (NUMCOLS NUMROWS WIDTH HEIGHT DSP BLK COLWIDTH ROWHEIGHT BITSPERPIXEL MENUITEMS NITEMS BORDER OUTLINE
                 FONT TITLEFONT TITLEHEIGHT TITLEWIDTH WINDOW TITLE ANYSUBITEMS? CENTER?)
          [COND
             ((NULL SCREEN)
              (COND
                 [(type? WINDOW (fetch (MENU IMAGE) of MNU))
                  (SETQ SCREEN (fetch (WINDOW SCREEN) of (fetch (MENU IMAGE) of MNU]
                 (T (SETQ SCREEN LASTSCREEN]
          (SETQ MENUITEMS (fetch (MENU ITEMS) of MNU))
          (SETQ CENTER? (fetch (MENU CENTERFLG) of MNU))          ; check the font.
          (COND
             [(FONTP (SETQ FONT (AND (fetch (MENU MENUFONT) of MNU)
                                     (\COERCEFONTDESC (fetch (MENU MENUFONT) of MNU)
                                            'DISPLAY T]
             (T [SETQ FONT (COND
                              ((FONTP MENUFONT))
                              (T (SETQ MENUFONT (FONTCREATE 'HELVETICA 10]
                                                                  ; keep font in the menu
                (replace (MENU MENUFONT) of MNU with FONT)))
          (COND
             ((SETQ TITLE (fetch (MENU TITLE) of MNU))            ; set the title font
              (SETQ TITLEFONT (MENUTITLEFONT MNU SCREEN))
              (SETQ TITLEHEIGHT (FONTPROP TITLEFONT 'HEIGHT))
              (SETQ TITLEWIDTH (STRINGWIDTH TITLE TITLEFONT)))
             (T (SETQ TITLEHEIGHT 0)
                (SETQ TITLEWIDTH 0)))                             ; calculate the number of columns and rows
          (SETQ NITEMS (LENGTH MENUITEMS))
          (COND
             [(SETQ NUMCOLS (NUMBERP (fetch (MENU MENUCOLUMNS) of MNU)))
              (SETQ NUMROWS (COND
                              ((NUMBERP (fetch (MENU MENUROWS) of MNU)))
                              (T (ADD1 (IQUOTIENT (SUB1 NITEMS)
                                                  NUMCOLS]
             [(SETQ NUMROWS (NUMBERP (fetch (MENU MENUROWS) of MNU)))
              (SETQ NUMCOLS (ADD1 (IQUOTIENT (SUB1 NITEMS)
                                             NUMROWS]
             (T (SETQ NUMCOLS 1)
                (SETQ NUMROWS NITEMS)))
```

;; set BORDER to the size of the outline around each menu item and OUTLINE to the size of the outline around the whole menu.

```
          (SETQ BORDER (OR (FIXP (fetch (MENU MENUBORDERSIZE) of MNU))
                           (replace (MENU MENUBORDERSIZE) of MNU with 0)))
          [SETQ OUTLINE (OR (FIXP (fetch (MENU MENUOUTLINESIZE) of MNU))
                            (replace (MENU MENUOUTLINESIZE) of MNU with (IMAX BORDER 1]
          (SETQ ANYSUBITEMS? (for I in (fetch (MENU ITEMS) of MNU) when (\MENUSUBITEMS MNU I)
                                do (RETURN T)))
          (COND
             ((IGREATERP (SETQ COLWIDTH (fetch (MENU ITEMWIDTH) of MNU))
                     5000)
```

;; If ITEMWIDTH is greater than 5000, it was probably default clipping region. if no columnwidth is given {common case}, calculate it
;; from the items widths.

```
    [SETQ COLWIDTH (IPLUS (MAXMENUITEMWIDTH MNU T)
                          (ITIMES (ADD1 BORDER)
                                  2)
                          (COND
                             (ANYSUBITEMS? (BITMAPWIDTH MENUSUBITEMMARK))
                             (T 0]
    [COND
       ((IGREATERP (IPLUS TITLEWIDTH 2)
                   (ITIMES COLWIDTH NUMCOLS))                ; adjust column width to cover title.
        (SETQ COLWIDTH (IQUOTIENT (IPLUS TITLEWIDTH (SUB1 NUMCOLS))
                                  NUMCOLS]
    (replace (MENU ITEMWIDTH) of MNU with COLWIDTH)))
(COND
   ((ILESSP (SETQ ROWHEIGHT (fetch (MENU ITEMHEIGHT) of MNU))
            5000)
    ROWHEIGHT)
   (T (SETQ ROWHEIGHT (IPLUS (MAXMENUITEMHEIGHT MNU)
                             (ITIMES BORDER 2)))
      (replace (MENU ITEMHEIGHT) of MNU with ROWHEIGHT)))
(SETQ WIDTH (IPLUS (ITIMES COLWIDTH NUMCOLS)
                   (ITIMES OUTLINE 2)))
(SETQ HEIGHT (IPLUS (ITIMES NUMROWS ROWHEIGHT)
                    (ITIMES OUTLINE 2)
                    TITLEHEIGHT))
[COND
   [(AND (IGREATERP HEIGHT (fetch (SCREEN SCHEIGHT) of SCREEN))
         (NULL (fetch (MENU MENUCOLUMNS) of MNU))
         (NULL (fetch (MENU MENUROWS) of MNU)))
    ;; it is too large to fit on the screen and menu is defaulting the number of columns and rows If the user specified either the number of
    ;; rows or columns, assume they knew what they were doing.
    (PROG (NITEMSTOFIT)                                      ; menu is defaulting the number of columns
          (SETQ NITEMSTOFIT (IQUOTIENT (IDIFFERENCE (fetch (SCREEN SCHEIGHT) of SCREEN)
                                                    TITLEHEIGHT)
                                       ROWHEIGHT))
          (SETQ NUMCOLS (ADD1 (IQUOTIENT (SUB1 NITEMS)
                                         NITEMSTOFIT)))
          (SETQ NUMROWS (ADD1 (IQUOTIENT (SUB1 NITEMS)
                                         NUMCOLS)))
          (SETQ WIDTH (IPLUS (ITIMES COLWIDTH NUMCOLS)
                             (ITIMES OUTLINE 2)))
          (SETQ HEIGHT (IPLUS (ITIMES NUMROWS ROWHEIGHT)
                              (ITIMES OUTLINE 2)
                              TITLEHEIGHT))
    ;; changing the items field is suspect since conceivably the user might be depending upon it.  At least the fact that MENUCOLUMNS
    ;; is NIL keeps it from happening twice if it gets called again.
          (replace (MENU ITEMS) of MNU with (SETQ MENUITEMS (\MAKE.ITEMS.VERT.ORDER MENUITEMS NUMROWS
                                                                                    NUMCOLS]
   ((AND (NULL (fetch (MENU MENUCOLUMNS) of MNU))
         (fetch (MENU MENUROWS) of MNU))
    ;; user wants a certain number of rows but doesn't care about the columns, switch to vertical order so the blanks items appear in the
    ;; last row.
    (replace (MENU ITEMS) of MNU with (SETQ MENUITEMS (\MAKE.ITEMS.VERT.ORDER MENUITEMS NUMROWS NUMCOLS
                                                                              ]
(replace (MENU MENUCOLUMNS) of MNU with NUMCOLS)
(replace (MENU MENUROWS) of MNU with NUMROWS)
(SETQ BITSPERPIXEL (OR (fetch (SCREEN SCDEPTH) of SCREEN)
                       (fetch (SCREEN SCBITSPERPIXEL) of SCREEN)))
[SETQ BLK (COND
             ((AND [SETQ BLK (COND
                                ((type? BITMAP (SETQ BLK (fetch (MENU IMAGE) of MNU)))
                                 BLK)
                                ((type? WINDOW BLK)          ; if it is a window, make sure it is not active, then
                                 (CLOSEW BLK)
                                 (fetch (WINDOW SAVE) of BLK]
                   (EQ (fetch (BITMAP BITMAPWIDTH) of BLK)
                       WIDTH)
                   (EQ (fetch (BITMAP BITMAPHEIGHT) of BLK)
                       HEIGHT)
                   (EQ (fetch (BITMAP BITMAPBITSPERPIXEL) of BLK)
                       BITSPERPIXEL))                        ; reuse current image bitmap
              BLK)
             (T                                              ; create a new one
                (BITMAPCREATE WIDTH HEIGHT BITSPERPIXEL]
(BITBLT NIL NIL NIL BLK 0 0 WIDTH HEIGHT 'TEXTURE 'REPLACE BLACKSHADE)
                                                             ; Draw box by nested BitBlts
                                                             ; leave outline
(BITBLT NIL NIL NIL BLK OUTLINE OUTLINE (IDIFFERENCE WIDTH (ITIMES OUTLINE 2))
       (IDIFFERENCE HEIGHT (IPLUS TITLEHEIGHT (ITIMES OUTLINE 2)))
       'TEXTURE
       'REPLACE WHITESHADE)
(SETQ DSP (DSPCREATE BLK))
(DSPRIGHTMARGIN MAX.SMALLP DSP)
(DSPXOFFSET OUTLINE DSP)
```

```
          (DSPYOFFSET OUTLINE DSP)
          (replace (REGION LEFT) of (fetch (MENU MENUGRID) of MNU) with 0)
          (replace (REGION BOTTOM) of (fetch (MENU MENUGRID) of MNU) with 0)
          (GRID (fetch (MENU MENUGRID) of MNU)
                NUMCOLS NUMROWS BORDER DSP)
          (DSPOPERATION 'INVERT DSP)                              ; calculate the offset from the top of the item box to the base line
                                                                  ; of the printed item.
          [COND
             (TITLE                                               ; if there is a title, display it
                     (DSPFONT TITLEFONT DSP)
                     (\SHOWMENULABEL TITLE (create REGION
                                                   LEFT _ BORDER
                                                   BOTTOM _ (IDIFFERENCE (IPLUS HEIGHT BORDER)
                                                                         (IPLUS TITLEHEIGHT (ITIMES OUTLINE 2)))
                                                   WIDTH _ WIDTH
                                                   HEIGHT _ TITLEHEIGHT)
                              MNU DSP CENTER?)
                     (SETQ HEIGHT (IDIFFERENCE HEIGHT TITLEHEIGHT]
          [PROG (ITEMREGION MAJOR#)
                [SETQ ITEMREGION (create REGION
                                         LEFT _ BORDER
                                         BOTTOM _ (IDIFFERENCE (IPLUS HEIGHT BORDER)
                                                               (IPLUS ROWHEIGHT (ITIMES OUTLINE 2)))
                                         WIDTH _ (IDIFFERENCE (IDIFFERENCE (fetch (REGION WIDTH)
                                                                                  of (fetch (MENU MENUGRID)
                                                                                       of MNU))
                                                                           (ITIMES BORDER 2))
                                                              (COND
                                                                 (ANYSUBITEMS?
                                                                                  ; the subitem mark goes outside of the normal title space
                                                                  (BITMAPWIDTH MENUSUBITEMMARK))
                                                                 (T 0)))
                                         HEIGHT _ (IDIFFERENCE ROWHEIGHT (ITIMES BORDER 2]
                (SETQ MAJOR# 1)
                (DSPFONT FONT DSP)
           LP   (COND
                   (MENUITEMS (\SHOWMENULABEL (CAR MENUITEMS)
                                              ITEMREGION MNU DSP CENTER?)
                              (SETQ MENUITEMS (CDR MENUITEMS))
                              [COND
                                 ((EQ MAJOR# NUMCOLS)             ; advance to the next row
                                  (SETQ MAJOR# 1)
                                  (replace (REGION BOTTOM) of ITEMREGION with (IDIFFERENCE (fetch (REGION BOTTOM)
                                                                                                  of ITEMREGION)
                                                                                           ROWHEIGHT))
                                  (replace (REGION LEFT) of ITEMREGION with BORDER))
                                 (T (SETQ MAJOR# (ADD1 MAJOR#))
                                    (replace (REGION LEFT) of ITEMREGION with (IPLUS (fetch (REGION LEFT) of ITEMREGION)
                                                                                     COLWIDTH]
                              (GO LP]
          [COND
             ((NULL (fetch (MENU MENUOFFSET) of MNU))
              ;; set offset so cursor will be be in middle of the menu on first display if it is to move with the cursor.  If it is fixed offset, initialize it to 0
              (replace (MENU MENUOFFSET) of MNU with (COND
                                                        ((fetch (MENU CHANGEOFFSETFLG) of MNU)
                                                         (create POSITION
                                                                 XCOORD _ (IQUOTIENT WIDTH 2)
                                                                 YCOORD _ (IQUOTIENT HEIGHT 2)))
                                                        (T (create POSITION
                                                                   XCOORD _ 0
                                                                   YCOORD _ 0]
          [COND
             ((AND (type? WINDOW (SETQ WINDOW (fetch (MENU IMAGE) of MNU)))
                   (EQ (fetch (WINDOW SCREEN) of WINDOW)
                       SCREEN))                                   ; menu has a window, replace its save image.
              (replace (WINDOW SAVE) of WINDOW with BLK))
             (T (replace (MENU IMAGE) of MNU with (SETQ WINDOW (CREATEWFROMIMAGE BLK SCREEN]
                                                                  ; tell the window about its border
          (replace (WINDOW WBORDER) of WINDOW with OUTLINE)
          (ADVISEWDS WINDOW)                                      ; snap circular link between the display stream created for
                                                                  ; printing and its stream.
          (RETURN (fetch (WINDOW SAVE) of (fetch (MENU IMAGE) of MNU])
```

(\MAKE.ITEMS.VERT.ORDER

```
  [LAMBDA (ITEMS %#ROWS %#COLUMNS)                                ; Edited  9-Apr-94 00:42 by rmk:
                                                                  (* rrb " 3-Feb-86 14:46")
                                                                  (* changes the order of a list of elements to be by row.)

    (PROG ((ROWS (for I to %#ROWS collect (CONS)))
           (ITEM.POINTER ITEMS)
           (EMPTY.STRING ""))
          [for C to %#COLUMNS do (for R in ROWS do (TCONC R (COND
                                                              [(LISTP ITEM.POINTER)
                                                                  (* still items left)
                                                               (PROG1 (CAR ITEM.POINTER)
                                                                      (SETQ ITEM.POINTER (CDR ITEM.POINTER)))]
```

```
                                                        (T  (* use a dummy item)
                                                            (LIST EMPTY.STRING '*DUMMYITEM*]
        (RETURN (for ROW in ROWS join (CAR ROW])
```

(\SHOWMENULABEL

```
[LAMBDA (ITEM ITEMREGION MENU DSP CENTER?)                  (* edited%: "31-Dec-00 18:58")
```

;; displays the item label for ITEM in the region ITEMREGION on the stream DSP according to the formatting information from MENU.

```
(DECLARE (GLOBALVARS MENUSUBITEMMARK))
(LET ((LABEL (\MENUITEMLABEL ITEM)))
     [COND
        ((\MENUSUBITEMS MENU ITEM)

(* * ; "this item has subitems, put the mark in.")

           (BITBLT MENUSUBITEMMARK 0 0 DSP (IPLUS (fetch (REGION LEFT) of ITEMREGION)
                                                  (fetch (REGION WIDTH) of ITEMREGION))
                  (IPLUS (fetch (REGION BOTTOM) of ITEMREGION)
                         (FONTPROP (fetch (MENU MENUFONT) of MENU)
                                'DESCENT))
                  NIL NIL 'INPUT 'REPLACE NIL (CREATEREGION (fetch (REGION LEFT) of ITEMREGION)
                                                            (fetch (REGION BOTTOM) of ITEMREGION)
                                                            (IPLUS (fetch (REGION WIDTH) of ITEMREGION)
                                                                   (BITMAPWIDTH MENUSUBITEMMARK))
                                                            (fetch (REGION HEIGHT) of ITEMREGION]
     (COND
        [(BITMAPP LABEL)                                    ; bitblt the label using the default operation of the displaystream.
         (COND
            (CENTER? (BITBLT LABEL 0 0 DSP (IPLUS (fetch (REGION LEFT) of ITEMREGION)
                                                  (IQUOTIENT (IDIFFERENCE (fetch (REGION WIDTH) of ITEMREGION
                                                                                                  )
                                                                          (BITMAPWIDTH LABEL))
                                                             2))
                                    (IPLUS (fetch (REGION BOTTOM) of ITEMREGION)
                                           (IQUOTIENT (IDIFFERENCE (fetch (REGION HEIGHT) of ITEMREGION)
                                                                   (fetch (BITMAP BITMAPHEIGHT) of LABEL))
                                                      2))
                                    NIL NIL 'INPUT NIL NIL ITEMREGION))
            (T (BITBLT LABEL 0 0 DSP (fetch (REGION LEFT) of ITEMREGION)
                                     (fetch (REGION BOTTOM) of ITEMREGION)
                                     (fetch (REGION WIDTH) of ITEMREGION)
                                     (fetch (REGION HEIGHT) of ITEMREGION)
                                     'INPUT NIL NIL]
        (CENTER? (CENTERPRINTINREGION LABEL ITEMREGION DSP (MENU.PRIN2.FLG MENU)))
        (T (DSPXPOSITION (ADD1 (fetch (REGION LEFT) of ITEMREGION))
                  DSP)
           (DSPYPOSITION (IPLUS (fetch (REGION BOTTOM) of ITEMREGION)
                                (FONTDESCENT (DSPFONT NIL DSP)))
                  DSP)
           (CL:FUNCALL (if (MENU.PRIN2.FLG MENU)
                           then (FUNCTION PRIN4)
                        else (FUNCTION PRIN3))
                  LABEL DSP])
```

(\POSITION.MENU.IMAGE

```
[LAMBDA (MENU POSITION)                                     ; Edited 5-Jan-94 17:16 by nilsson
```

;; puts a menu image window in the right place on the screen. Subfunction to MENU

```
(PROG (SCREEN IMAGE MX MY)                                  ; make sure the image is a window
      (OR POSITION (SETQ POSITION (fetch (MENU MENUPOSITION) of MENU)))
      (COND
         ((type? SCREENPOSITION POSITION)
          (SETQ SCREEN (fetch (SCREENPOSITION SCREEN) of POSITION))
          (SETQ MX (fetch (SCREENPOSITION XCOORD) of POSITION))
          (SETQ MY (fetch (SCREENPOSITION YCOORD) of POSITION)))
         ((type? POSITION POSITION)
          (SETQ MX (fetch (POSITION XCOORD) of POSITION))
          (SETQ MY (fetch (POSITION YCOORD) of POSITION))
          (GETMOUSESTATE)
          (SETQ SCREEN LASTSCREEN))
         (T (GETMOUSESTATE)
            (SETQ MX LASTMOUSEX)
            (SETQ MY LASTMOUSEY)
            (SETQ SCREEN LASTSCREEN)))                      ; make sure the image is a window
      (CHECK/MENU/IMAGE MENU T SCREEN)
      (SETQ IMAGE (fetch (MENU IMAGE) of MENU))
      [SETQ MX (IDIFFERENCE MX (fetch (POSITION XCOORD) of (fetch (MENU MENUOFFSET) of MENU]
      [SETQ MY (IDIFFERENCE MY (fetch (POSITION YCOORD) of (fetch (MENU MENUOFFSET) of MENU]
                                                            ; Adjust the position so that the menu will be entirely on the
                                                            ; screen.
```

;; do left margin first so that if the menu is wider than the screen, the left most part of it will be shown

```
      (SETQ MX (IMAX (IMIN MX (IDIFFERENCE (fetch (SCREEN SCWIDTH) of SCREEN)
                                           (fetch (MENU IMAGEWIDTH) of MENU)))
                     0))
```

```
;; do the bottom margin first so that the top of the menu will show if the menu is higher than the a screen
      [SETQ MY (IMIN (IMAX MY 0)
                     (IDIFFERENCE (fetch (SCREEN SCHEIGHT) of SCREEN)
                                  (fetch (MENU IMAGEHEIGHT) of MENU]
      (SETQ IMAGE (fetch (MENU IMAGE) of MENU))
      (MOVEW IMAGE MX MY)
      (SHOWSHADEDITEMS MENU IMAGE)
      (RETURN IMAGE])
```

(\SMASHMENUIMAGEONRESET

```
[LAMBDA (MENU)                                          (* rrb " 9-Jan-84 19:23")

      (* sets the menu image field to NIL if RESETSTATE indicates that a ^D was typed.)

  (COND
     ((FMEMB RESETSTATE '(RESET HARDRESET))
      (replace (MENU IMAGE) of MENU with NIL])
```

(CLOSE.PROCESS.MENU

```
[LAMBDA (WINDOW)                                        (* dgb%: "15-DEC-83 19:18")
  (WAKE.PROCESS (WINDOWPROP WINDOW 'MENUPROCESS)
         T])
```

(DEFAULTSUBITEMFN

```
[LAMBDA (MENU ITEM)                                     (* rrb "17-Aug-84 17:24")
                                                        (* default subitemfn for menus. Checks the fourth element of the
                                                        (item for an expression of the form
                                                        (SUBITEMS a b c)))
  (AND (LISTP ITEM)
       (LISTP (SETQ ITEM (CDR ITEM)))
       (LISTP (SETQ ITEM (CDR ITEM)))
       (LISTP (SETQ ITEM (CDR ITEM)))
       (EQ [CAR (SETQ ITEM (LISTP (CAR ITEM]
           'SUBITEMS)
       (CDR ITEM])
```

(GETMENUPROP

```
[LAMBDA (MENU PROPERTY)                                 (* dgb%: "13-DEC-83 17:50")
  (LISTGET (fetch (MENU MENUUSERDATA) of MENU)
         PROPERTY])
```

(PUTMENUPROP

```
[LAMBDA (MENU PROPERTY VALUE)                           (* dgb%: "13-DEC-83 17:52")
  (PROG ((NOWDATA (fetch (MENU MENUUSERDATA) of MENU)))
        [COND
           (NOWDATA (LISTPUT NOWDATA PROPERTY VALUE))
           (T (replace (MENU MENUUSERDATA) of MENU with (LIST PROPERTY VALUE]
        (RETURN VALUE])
```

(WAKE.MY.PROCESS

```
[LAMBDA (WINDOW)                                        (* dgb%: "15-DEC-83 19:09")
  (WAKE.PROCESS (WINDOWPROP WINDOW 'MENUPROCESS)
         "ABC"])
```

(\INVERTITEM

```
[LAMBDA (COLUMN ROW MENU DSP)                           (* dgb%: "13-DEC-83 18:06")
                                                        (* inverts an item in a menu displayed in DSP.)
  (SHADEGRIDBOX COLUMN ROW BLACKSHADE 'INVERT (fetch (MENU MENUGRID) of MENU)
         (fetch (MENU MENUBORDERSIZE) of MENU)
         DSP])
```

(\MENU.ITEM.SELECT

```
[LAMBDA (COLUMN ROW MENU DSP)                           ; Edited  9-Apr-94 09:14 by rmk:
                                                        (* rrb "21-May-85 13:57")
  (DECLARE (USEDFREE SUBMENU))

      (* selects an item in a menu displayed in DSP. Looks for submenus and brings those up as well.
      Returns the image window of the submenu if it was brought up.)

  (PROG ((ITEM (GETMENUITEM MENU COLUMN ROW))
         SUBITEMS)
        (CL:UNLESS [EQ '*DUMMYITEM* (CAR (LISTP (CDR (LISTP ITEM]
            (\INVERTITEM COLUMN ROW MENU DSP)
            [RETURN (AND ITEM (SETQ SUBITEMS (\MENUSUBITEMS MENU ITEM))
                         (COND
                            [(EQ (CAR SUBITEMS)
                                 'POPUP)                (* if the first item is POPUP then bring up the menu.)
                             (SETQ SUBMENU (NESTED.SUBMENU MENU (CDR SUBITEMS)))
```

```
                    (OPENW (\POSITION.MENU.IMAGE SUBMENU (NESTED.SUBMENU.POS MENU ITEM DSP]
                   (T                                    (* otherwise just create it but don't bring it up.)
                      (SETQ SUBMENU (NESTED.SUBMENU MENU SUBITEMS))
                      NIL])])
```

(\MENU.ITEM.DESELECT

```
[LAMBDA (COLUMN ROW MENU DSP)                                  ; Edited  9-Apr-94 09:15 by rmk:
                                                               (* rrb "21-May-85 15:11")
  (DECLARE (USEDFREE SUBMENU SUBMENUWINDOW))

        (* deselects an item in a menu displayed in DSP. Also takes care of closing the submenu and resetting the variables that
        indicate that there is a submenu.)

  (CL:UNLESS [EQ '*DUMMYITEM* (CAR (LISTP (CDR (LISTP (GETMENUITEM MENU COLUMN ROW]
      (\INVERTITEM COLUMN ROW MENU DSP)
      (AND SUBMENUWINDOW (CLOSEW SUBMENUWINDOW))
      (SETQ SUBMENUWINDOW (SETQ SUBMENU NIL)))])
```

(\ItemNumber

```
[LAMBDA (ITEM ALLITEMS)                                        ; Edited  8-Jul-93 19:26 by sybalskY:MV:ENVOS
  ;; Walk thru the list of items in a menu, returning the relative item # of the menu item that matches ITEM.  Failing that, return NIL.
  (for SOMEITEM in ALLITEMS as ITEMNUM from 1 do (COND
                                                     ([OR (EQ SOMEITEM ITEM)
                                                          (EQ ITEM (CAR (LISTP SOMEITEM]
                                                      (RETURN ITEMNUM)))
     finally (RETURN NIL])
```

(\BOXITEM

```
[LAMBDA (COLUMN ROW MENU DSP)                                  ; Edited  9-Apr-94 09:39 by rmk:
                                                               (* rrb "28-Dec-83 17:34")
                                                               (* inverts an item in a menu displayed in DSP.)
  (CL:UNLESS [EQ '*DUMMYITEM* (CAR (LISTP (CDR (LISTP (GETMENUITEM MENU COLUMN ROW]
      (PROG ((BORDER (OR (FIXP (fetch (MENU MENUBORDERSIZE) of MENU))
                         0))
             (GRID (fetch (MENU MENUGRID) of MENU))
             LFT BTM WID HGHT)
            (BITBLT NIL NIL NIL DSP (SETQ LFT (IPLUS (LEFTOFGRIDCOORD COLUMN GRID)
                                                     BORDER))
                   (SETQ BTM (IPLUS (BOTTOMOFGRIDCOORD ROW GRID)
                                    BORDER))
                   (SETQ WID (IDIFFERENCE (fetch (REGION WIDTH) of GRID)
                                          (ITIMES BORDER 2)))
                   (SETQ HGHT (IDIFFERENCE (fetch (REGION HEIGHT) of GRID)
                                           (ITIMES BORDER 2)))
                   'TEXTURE
                   'INVERT BLACKSHADE)
            (BITBLT NIL NIL NIL DSP (ADD1 LFT)
                   (ADD1 BTM)
                   (IDIFFERENCE WID 2)
                   (IDIFFERENCE HGHT 2)
                   'TEXTURE
                   'INVERT BLACKSHADE)))])
```

(NESTED.SUBMENU

```
[LAMBDA (MENU SUBITEMS)                                        (* rrb "20-Jun-84 19:26")

        (* computes and returns the nested submenu for SUBITEMS. It maintains a cache on the MENUUSERDATA)

  (PROG [SUBMENU (SUBMENULST (GETMENUPROP MENU 'SUBMENUS]
        [COND
           ([NULL (SETQ SUBMENU (CDR (FASSOC SUBITEMS SUBMENULST]
                                                               (* Cache submenu on user data)
            (PUTMENUPROP MENU 'SUBMENUS (CONS [CONS SUBITEMS
                                                    (SETQ SUBMENU
                                                     (create MENU
                                                            ITEMS _ SUBITEMS
                                                            MENUOFFSET _
                                                            (create POSITION
                                                                   XCOORD _ 1
                                                                   YCOORD _ 5)
                                                            CHANGEOFFSETFLG _ 'Y
                                                            CENTERFLG _ (fetch (MENU CENTERFLG) of MENU)
                                                            MENUFONT _ (fetch (MENU MENUFONT) of MENU)
                                                            MENUBORDERSIZE _ (fetch (MENU MENUBORDERSIZE)
                                                                                    of MENU)
                                                            MENUOUTLINESIZE _ (IMAX (fetch (MENU
                                                                                           MENUOUTLINESIZE
                                                                                           )
                                                                                    of MENU)
                                                                                  1)
                                                            WHENHELDFN _ (fetch (MENU WHENHELDFN)
                                                                                of MENU)
```

```
                                                        WHENUNHELDFN _ (fetch (MENU WHENUNHELDFN)
                                                                          of MENU)
                                                        SUBITEMFN _ (fetch (MENU SUBITEMFN) of MENU]
                                                SUBMENULST]
        (RETURN SUBMENU])
```

(NESTED.SUBMENU.POS

```
  [LAMBDA (IMENU ITEM STREAM)                                   (* rrb "28-Dec-83 19:24")
                                                                (* return the position of a nested submenu should have.)
    (PROG (ITEMNUMBER (ITEMS (fetch (MENU ITEMS) of IMENU))
           (GRIDSPEC (fetch (MENU MENUGRID) of IMENU))
           (BORDER (fetch (MENU MENUBORDERSIZE) of IMENU))
           (DD (\GETDISPLAYDATA STREAM)))
          [SETQ ITEMNUMBER (IDIFFERENCE (LENGTH ITEMS)
                                        (LENGTH (OR (FMEMB ITEM ITEMS)
                                                    (for ITEMTAIL on ITEMS when (EQ (CAAR ITEMTAIL)
                                                                                    ITEM)
                                                       do (RETURN ITEMTAIL))
                                                    (RETURN]
          (RETURN (create POSITION
                          XCOORD _ (\DSPTRANSFORMX (IPLUS (fetch (REGION LEFT) of GRIDSPEC)
                                                          (ITIMES (IREMAINDER ITEMNUMBER (fetch (MENU MENUCOLUMNS
                                                                                                )
                                                                                          of IMENU))
                                                                  (fetch (REGION WIDTH) of GRIDSPEC))
                                                          (IDIFFERENCE (fetch (REGION WIDTH) of GRIDSPEC)
                                                                  (ITIMES 2 BORDER)))
                                                   DD)
                          YCOORD _ (\DSPTRANSFORMY (IPLUS (ITIMES -2 BORDER)
                                                          (fetch (REGION BOTTOM) of GRIDSPEC)
                                                          (ITIMES [SUB1 (IDIFFERENCE (fetch (MENU MENUROWS)
                                                                                            of IMENU)
                                                                                     (IQUOTIENT ITEMNUMBER
                                                                                            (fetch (MENU MENUCOLUMNS)
                                                                                              of IMENU]
                                                                  (fetch (REGION HEIGHT) of GRIDSPEC)))
                                                   DD])
```

(WFROMMENU

```
  [LAMBDA (MENU)                                                (* kbr%: " 3-Apr-85 11:38")
                                                                (* finds the window that menu is in if any.)
    (for WINDOW in (OPENWINDOWS T) thereis (MEMB MENU (WINDOWPROP WINDOW 'MENU])

)

(RPAQQ MENUSUBITEMMARK )

(RPAQ? MENUFONT (FONTCREATE 'HELVETICA 10))

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS MENU.HELDSTATE.RESET MACRO ((BX BY)
                                      [COND
                                         (HELDSTATE (COND
                                                       ((SETQ HELDSTATE (fetch (MENU WHENUNHELDFN)
                                                                           of MENU))
                                                        (APPLY* HELDSTATE (GETMENUITEM MENU BX BY)
                                                               MENU
                                                                (\FDECODE/BUTTON LASTBUTTONSTATE))
                                                        (SETQ HELDSTATE NIL]
                                      (SETQ HOLDTIMER (SETUPTIMER MENUHELDWAIT HOLDTIMER))))

(PUTPROPS MENU.PRIN2.FLG MACRO ((MNU)
                                (LISTGET (fetch (MENU MENUUSERDATA) of MNU)
                                       :ESCAPE)))
)
)

;; scrolling menu functions and utilities

(DEFINEQ

(MENUREPAINTFN
  [LAMBDA (WINDOW REG)                                          (* hdj "11-Apr-85 12:11")
                                                                (* repaints the menus in a window.)
    (PROG [(DSP (WINDOWPROP WINDOW 'DSP]                        (* stuff new images over old)
          (for MENU in (REVERSE (WINDOWPROP WINDOW 'MENU)) do (BLTMENUIMAGE MENU DSP])

)

;; misc utility fns.
```

```
(DEFINEQ
```

(MAXSTRINGWIDTH

```
  [LAMBDA (L FONT PRIN2FLG RDTBL)                                    (* rrb " 9-JAN-82 13:49")
    (bind (M _ 0) for I in L do (SETQ M (IMAX M (STRINGWIDTH I FONT PRIN2FLG RDTBL))) finally (RETURN M])
```

(CENTEREDPRIN1

```
  [LAMBDA (EXP DS LEFT WIDTH Y)                                      (* rrb "17-DEC-82 15:53")
                                                                     (* prints an expression in the middle of a width.)
    (MOVETO (IPLUS LEFT (IQUOTIENT (IDIFFERENCE WIDTH (STRINGWIDTH EXP DS))
                                   2))
           Y DS)
    (PRIN3 EXP DS])
```

(CENTERPRINTINREGION

```
  [LAMBDA (EXP REGION STREAM PRIN2FLG)                               (* bvm%: "14-Oct-86 12:55")
    ;; prints an expression in the middle of a region
    (OR (type? REGION REGION)
        (SETQ REGION (DSPCLIPPINGREGION NIL STREAM)))
    (CENTERPRINTINAREA EXP (fetch (REGION LEFT) of REGION)
           (fetch (REGION BOTTOM) of REGION)
           (fetch (REGION WIDTH) of REGION)
           (fetch (REGION HEIGHT) of REGION)
           STREAM PRIN2FLG])
```

(CENTERPRINTINAREA

```
  [LAMBDA (EXP X Y WIDTH HEIGHT STREAM PRIN2FLG)                     (* bvm%: "14-Oct-86 12:54")
    ;; prints expression EXP in a box defined by remaining args.  If PRIN2FLG is true uses PRIN2-pnames
    (SETQ STREAM (\OUTSTREAMARG STREAM))
    (PROG ((STRWIDTH (STRINGWIDTH EXP STREAM PRIN2FLG))
           (PFN (if PRIN2FLG
                    then (FUNCTION PRIN4)
                  else (FUNCTION PRIN3)))
           XPOS)
          (MOVETO (SETQ XPOS (IPLUS X (IQUOTIENT (ADD1 (IDIFFERENCE WIDTH STRWIDTH))
                                                 2)))
                  (IPLUS Y (IQUOTIENT (IPLUS (IDIFFERENCE HEIGHT (FONTPROP STREAM 'ASCENT))
                                             (FONTPROP STREAM 'DESCENT))
                                      2))
                  STREAM)
          (COND
             ((IGREATERP (IPLUS XPOS STRWIDTH)
                         (DSPRIGHTMARGIN NIL STREAM))
              ;; if string would cause a CR to be inserted, change the right margin to avoid it.  When PRIN3 is fixed so that it never inserts CR, this
              ;; can be removed.
              (RESETLST
                  (RESETSAVE NIL (LIST 'DSPRIGHTMARGIN (DSPRIGHTMARGIN (IPLUS XPOS STRWIDTH)
                                                                       STREAM)
                                       STREAM))
                  (CL:FUNCALL PFN EXP STREAM)))
             (T (CL:FUNCALL PFN EXP STREAM])
```

(STRICTLY/BETWEEN

```
  [LAMBDA (VAL LOWER HIGHER)                                         (* rrb "30-JUL-81 14:53")
                                                                     (* returns T if VAL is strictly between LOWER and HIGHER)
    (AND (IGREATERP VAL LOWER)
         (IGREATERP HIGHER VAL])

)

;; examples of use.

(DEFINEQ
```

(UNREADITEM

```
  [LAMBDA (ITEM MENU BUTTON)                                         (* rrb "31-JUL-81 17:37")
    (BKSYSBUF (CONCAT (MKSTRING (COND
                                  ((LISTP ITEM)
                                   (EVAL (CADR ITEM)))
                                  (T ITEM)))
                      "
                      "])
```

(TYPEINMENU

```
  [LAMBDA (LST)                                                      (* rrb "17-NOV-81 14:04")
    (create MENU
           ITEMS _ LST
           WHENSELECTEDFN _ (FUNCTION UNREADITEM])
```

(SHADEITEM

```
  [LAMBDA (ITEM MENU SHADE DS/W)                                    (* lmm "16-Nov-86 01:01")
                                                                    ; shades a menu item with a background shade.  DS/W if
                                                                    ; provided is the displaystream to use.

    (PROG ((NEWSHADE (OR SHADE WHITESHADE))
           DSP ITEMREGION)
          (SETQ ITEMREGION (MENUITEMREGION ITEM MENU))              ; if the item isn't in MENU don't do anything.
          (OR ITEMREGION (RETURN))
          (\AddShade ITEM NEWSHADE MENU)                            ; if the menu is not in a window mark it as shaded but don't try to
                                                                    ; display
          (COND
             ([SETQ DSP (COND
                           [(NULL DS/W)
                            (COND
                               ((SETQ DSP (WFROMMENU MENU))
                                (WINDOWPROP DSP 'DSP]
                           ((DISPLAYSTREAMP (GETSTREAM DS/W 'OUTPUT]
              [COND
                 ((for ITEM in (fetch (MENU ITEMS) of MENU) thereis (\MENUSUBITEMS MENU ITEM))
                  (replace (REGION WIDTH) of ITEMREGION with (DIFFERENCE (fetch (REGION WIDTH) of ITEMREGION)
                                                                         (BITMAPWIDTH MENUSUBITEMMARK]
              (RESHADEITEM ITEM ITEMREGION MENU NEWSHADE DSP])
```

(RESHADEITEM

```
  [LAMBDA (ITEM ITEMREGION MENU NEWSHADE DSP)                       (* edited%: "31-Dec-00 19:16")
    (RESETLST
        (if (FONTP NEWSHADE)
            then (DSPFILL ITEMREGION BLACKSHADE 'ERASE DSP)
                 (RESETSAVE NIL (LIST 'DSPFONT (DSPFONT NEWSHADE DSP)
                                      DSP))
          else (DSPFILL ITEMREGION NEWSHADE 'REPLACE DSP)
               (RESETSAVE NIL (LIST 'DSPOPERATION (DSPOPERATION (MOST/VISIBLE/OPERATION NEWSHADE)
                                                                DSP)
                                    DSP))
               (RESETSAVE NIL (LIST 'DSPFONT (DSPFONT (fetch (MENU MENUFONT) of MENU)
                                                      DSP)
                                    DSP)))
        (RESETSAVE NIL (LIST 'DSPRIGHTMARGIN (DSPRIGHTMARGIN 64000 DSP)
                             DSP))
        (\SHOWMENULABEL ITEM ITEMREGION MENU DSP (fetch (MENU CENTERFLG) of MENU)))])
```

(MOST/VISIBLE/OPERATION

```
  [LAMBDA (SHADE)

          (* chooses the operation that is most visible way of putting characters on a SHADE background.)

    (COND
       ((IGREATERP (%#BITSON SHADE)
                   8)
        'ERASE)
       (T 'PAINT])
```

(%#BITSON

```
  [LAMBDA (N)                                                       (* rrb "16-AUG-81 18:35")
                                                                    (* determines the number of bits that are on.)
    (PROG ((MASK 1)
           (I 1)
           NBITS)
          (COND
             ((NOT (ZEROP (LOGAND N 1)))
              (SETQ NBITS 1))
             (T (SETQ NBITS 0)))
      LP  (COND
             ((EQ I BITSPERSHADE)
              (RETURN NBITS)))
          (SETQ MASK (LLSH MASK 1))
          (SETQ I (ADD1 I))
          [COND
             ((NOT (ZEROP (LOGAND N MASK)))
              (SETQ NBITS (ADD1 NBITS]
          (GO LP])
```

(BUTTONPANEL

```
  [LAMBDA (LABELLST)                                                (* rrb "17-NOV-81 14:09")
                                                                    (* make items which have second element that marks whether
                                                                    or not they are selected.)
    (create MENU
            ITEMS _ (for LABEL in LABELLST collect (LIST LABEL "Release the button to select this item." NIL))
            CENTERFLG _ T
            WHENSELECTEDFN _ (FUNCTION BUTTONPANEL/SELECTION/FN)
            WHENHELDFN _ (FUNCTION PPROMPT2])
```

(BUTTONPANEL/SELECTION/FN

```
  [LAMBDA (ITEM MENU BUTTON WINDOW)                          (* rrb "10-NOV-81 09:25")
                                                             (* flips the selection and shades the background.)
    (SHADEITEM ITEM MENU (COND
                           ((CADDR ITEM)
                            WHITESHADE)
                           (T MENUSELECTSHADE))
           WINDOW)
    (RPLACA (CDDR ITEM)
           (NOT (CADDR ITEM])
```

(GETSELECTEDITEMS

```
  [LAMBDA (WMENU)                                            (* rrb "10-NOV-81 09:26")
    (for ITEM in (fetch ITEMS of WMENU) when (CADDR ITEM) collect (CAR ITEM])

)

(RPAQQ EDITCMDS ("P" "PP" ("LF" "%
")
                     0 1 -1 2 3 "BK" "EF" "EVAL"))

(RPAQQ MENUHELDWAIT 1200)

(DECLARE%: EVAL@COMPILE

(RPAQQ BITSPERSHADE 16)

(CONSTANTS (BITSPERSHADE 16))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS MENUSELECTSHADE)
)

(RPAQQ MENUSELECTSHADE 32800)

(DEFINEQ
```

(MENUDESELECT

```
  [LAMBDA (ITEM MENU)                                        (* deselects a menu item)
    (SHADEITEM ITEM MENU WHITESHADE)
    (replace (MENU MENUUSERDATA) of MENU with NIL])
```

(MENUSELECT

```
  [LAMBDA (ITEM MENU)                                        (* rrb "23-SEP-81 15:26")
                                                             (* selects a menu item)
    (SHADEITEM ITEM MENU MENUSELECTSHADE)
    (replace (MENU MENUUSERDATA) of MENU with ITEM])

)

(DECLARE%: DOCOPY DONTEVAL@LOAD

(RPAQQ MENUFONT NIL)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS MENUFONT MENUHELDWAIT)
)

(DECLARE%: EVAL@COMPILE

[DATATYPE MENU
       (IMAGE SAVEIMAGE ITEMS MENUROWS MENUCOLUMNS MENUGRID CENTERFLG CHANGEOFFSETFLG MENUFONT TITLE MENUOFFSET
              WHENSELECTEDFN MENUBORDERSIZE MENUOUTLINESIZE WHENHELDFN MENUPOSITION WHENUNHELDFN MENUUSERDATA
              MENUTITLEFONT SUBITEMFN MENUFEEDBACKFLG SHADEDITEMS)
       MENUGRID _ (create REGION
                         LEFT _ 0
                         BOTTOM _ 0)
       WHENHELDFN _ 'DEFAULTMENUHELDFN WHENUNHELDFN _ 'CLRPROMPT
       (ACCESSFNS ((ITEMWIDTH (fetch (REGION WIDTH) of (fetch (MENU MENUGRID) of DATUM))
                              (replace (REGION WIDTH) of (fetch (MENU MENUGRID) of DATUM) with NEWVALUE))
                   (ITEMHEIGHT (fetch (REGION HEIGHT) of (fetch (MENU MENUGRID) of DATUM))
                              (replace (REGION HEIGHT) of (fetch (MENU MENUGRID) of DATUM) with NEWVALUE))
                   (IMAGEWIDTH (BITMAPWIDTH (CHECK/MENU/IMAGE DATUM)))
                   (IMAGEHEIGHT (BITMAPHEIGHT (CHECK/MENU/IMAGE DATUM)))
                   (MENUREGIONLEFT (IDIFFERENCE (fetch (REGION LEFT) of (fetch (MENU MENUGRID) of DATUM))
                                                (fetch (MENU MENUOUTLINESIZE) of DATUM)))
                   (MENUREGIONBOTTOM (IDIFFERENCE (fetch (REGION BOTTOM) of (fetch (MENU MENUGRID) of DATUM))
                                                  (fetch (MENU MENUOUTLINESIZE) of DATUM]
)
```

```
(/DECLAREDATATYPE 'MENU
       '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
              POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER)
       ;; ---field descriptor list elided by lister---
       '44)

(PUTPROPS MENU COPYRIGHT ("Venue & Xerox Corporation" 1982 1983 1984 1985 1986 1987 1988 1990 1991 1993 1994
                                  1999))
```

FUNCTION INDEX

VARIABLE INDEX

MACRO INDEX

RECORD INDEX

CONSTANT INDEX