File created: 8-Nov-88 17:04:43 **{ERINYES}<LISPUSERS>MEDLEY>TMENU.;2**

changes to: (FNS ReShapeMenu)

previous date: 16-Feb-87 16:33:23 {ERINYES}<LISPUSERS>MEDLEY>TMENU.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;


```
(RPAQQ TMENUCOMS
       [(* Copyright (c)
          1982 by Xerox Corporation.)
        (* * Functions to support editable menus that insert items into the TTY stream. Written in 1981 by Mark
          Stefik, Danny Bobrow, and Christopher Tong.)
        (FNS * TMENUFNS)
        (* * Fns to support WindowShade feature.)
        (FNS * WINDOWSHADEFNS)
        (* * These fns would probably not be called by a user.)
        (FNS * InternalTMENUFNS)
        (VARS YellowButtonItems (YellowButtonMenu NIL)
              (firstCallFlgTmenu T)
              (lastDeletedItem NIL)
              (menuedFiles NIL))
        (* Display Utility Functions)
        (FNS SELECTW FLIPREGION)
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                                (NLAML)
                                                                                (LAMA CPROMPT PROMPT])
```

(* * Copyright (c) 1982 by Xerox Corporation.)

(* * Functions to support editable menus that insert items into the TTY stream.
Written in 1981 by Mark Stefik, Danny Bobrow, and Christopher Tong.)

```
(RPAQQ TMENUFNS (MakeFileMenus TMenu PROMPT CPROMPT CloseFileMenus))

(DEFINEQ
```

(MakeFileMenus

```
  [LAMBDA (fileName)                                          (* cht%: " 8-SEP-82 23:58")
```

(* * Make menus for the FNS and CLASSES in file.)

```
    (PROG (title (coms (PACK* fileName 'COMS))
                 comsList relevantComs relevantComsName windows)
```

(* * Make menus for the FNS and CLASSES in the file.)

```
          (SETQ comsList (EVAL coms))
          (for com in comsList do (SELECTQ (CAR com)
                                           (VARS [COND
                                                     ((EQ (CADR com)
                                                          '*)
                                                      (SETQ title (CONCAT (CAR com)
                                                                          '%
                                                                          (CADDR com)))
                                                      (SETQ windows (CONS (TMenu (LIST 'FetchNames (CADDR com))
                                                                                 title T 'WindowShade)
                                                                          windows))
                                                      (SETQ relevantComs (CONS (CADDR com)
                                                                               relevantComs])
                                           ((FNS CLASSES)
                                               [COND
                                                  ((EQ (CADR com)
                                                       '*)
                                                   (SETQ title (CONCAT (CAR com)
                                                                       '%
                                                                       (CADDR com)))
                                                   (SETQ windows (CONS (TMenu (CADDR com)
                                                                              title T 'WindowShade)
                                                                       windows))
                                                   (SETQ relevantComs (CONS (CADDR com)
                                                                            relevantComs)))
                                                  (T (SETQ title (CONCAT (CAR com)
                                                                         '%  fileName))
                                                     (SETQ windows (CONS (TMenu (LIST 'QUOTE (CDR com))
                                                                                title T 'WindowShade)
                                                                         windows])
                                           NIL))
```

```
(* * Make menu of the coms for which menus were just created.)

[SETQ relevantComs (SORT (CONS fileName (CONS coms relevantComs]
(SETQ relevantComsName (PACK* 'RELEVANT%  coms))
(SET relevantComsName relevantComs)
(SETQ windows (CONS (TMenu relevantComsName relevantComsName T 'WindowShade)
                    windows))
(SETQ menuedFiles (CONS (LIST fileName windows)
                        menuedFiles])
```

(TMenu

```
  [LAMBDA (itemExpr title displaySpec windowShadeFlg buttonFn defaultTrailerString)
                                                                  (* mjs%: " 7-DEC-82 13:17")

(* * Creates a menu of items. Buttons work as follows -
LEFT. When items are selected using the LEFT mouse button, they are placed in the terminal input buffer.
RIGHT. The RIGHT button gives the usual window commands, including a SHAPE command tailored for these menus.
MIDDLE. The MIDDLE button gives menu commands for adding or deleting an item, or for sorting the list of items.
-
The optional argument displaySpec has the following interpretations -
REGION. The menu window is placed in that region. -
NUMBER. The number is used as the number of columns in the menu.
A window is allocated of the minimal sufficient size. The user gets to place the window.
-
T. The number of columns is computed by assuming a maximum of 15 rows per column.
Otherwise like previous case. NIL. The user is prompted for a bounding box for a window.
-
-
The optinal argument buttonFn is the name of the function to be used as the BUTTONEVENTFN for the window;
it defaults to MenuButtonFn. -
-
The optional argument defaultTrailerString is the string to be inserted after the item in the TTY buffer.
If not specified, then a space is used.)

    (PROG (window menu items dsp (font (FONTCREATE 'GACHA 10)))
          [COND
             ((NLISTP itemExpr)
              (SETQ items (EVAL itemExpr)))
             ((FGETD (CAR itemExpr))
              (SETQ items (EVAL itemExpr)))
             (T (SETQ items itemExpr)
                (SETQ itemExpr (KWOTE itemExpr]
          [COND
             ((EQ T displaySpec)                                  (* Here if Caller wants standard shape.)
              (SETQ displaySpec (CEILING (FQUOTIENT (LENGTH items)
                                                    15]
          [SETQ window (COND
                         ((NUMBERP displaySpec)                   (* Here if Caller specifies number of menu columns.)
                          (SETQ menu (create MENU
                                             ITEMS _ items
                                             WHENSELECTEDFN _ (FUNCTION UnreadExpr)
                                             MENUFONT _ font
                                             MENUCOLUMNS _ displaySpec
                                             CENTERFLG _ T
                                             MENUOUTLINESIZE _ 0))
                          (SETQ dsp (DSPCREATE))
                          (DSPCLIPPINGREGION (create REGION
                                                     LEFT _ 0
                                                     BOTTOM _ 0
                                                     WIDTH _ (fetch IMAGEWIDTH of menu)
                                                     HEIGHT _ (fetch IMAGEHEIGHT of menu))
                                 dsp)
                          (SETQ window (CREATEW dsp title))
                          (ADDMENU menu window (create POSITION
                                                       XCOORD _ 0
                                                       YCOORD _ 0))
                          (MOVEW window)
                          window)
                         (T (ReShapeMenu (CREATEW displaySpec title)
                                   NIL NIL (SETQ menu (create MENU
                                                              ITEMS _ items
                                                              WHENSELECTEDFN _ (FUNCTION UnreadExpr)
                                                              MENUFONT _ font
                                                              MENUCOLUMNS _ 1
                                                              CENTERFLG _ T]
          (WINDOWPROP window 'MENUEXPR itemExpr)
          (WINDOWPROP window 'RESHAPEFN (FUNCTION ReShapeMenu))
          [WINDOWPROP window 'BUTTONEVENTFN (APPLY 'FUNCTION (LIST (OR buttonFn 'MenuButtonFn]
          (COND
             (defaultTrailerString (WINDOWPROP window 'DEFAULTTRAILERSTRING defaultTrailerString)))
          (COND
             (windowShadeFlg (MakeWindowShade window)))
          (RETURN window])
```

(PROMPT

```
  [LAMBDA nargs                                                    (* ct%: " 8-DEC-81 14:14")
```

(* Print an arbitrary number of arguments in the prompt window, after first clearing the window.)

```
    (DSPFILL NIL BLACKSHADE 'INPUT PROMPTWINDOW)
    (DSPRESET PROMPTWINDOW)
    (PROG ((I 0))
          (while (ILESSP I nargs) do (PRIN1 (ARG nargs (SETQ I (ADD1 I)))
                                            PROMPTWINDOW])
```

(CPROMPT

```
  [LAMBDA nargs                                                    (* ct%: " 8-DEC-81 14:14")
```

(* Print an arbitrary number of arguments centered in the prompt window, after first clearing the window.
A call with no arguments simply clears the window.)

```
    (PROG (MSG (I 0))
          (DSPFILL NIL BLACKSHADE 'INPUT PROMPTWINDOW)
          (DSPRESET PROMPTWINDOW)
          [SETQ MSG (while (ILESSP I nargs) collect (ARG nargs (SETQ I (ADD1 I]
          (COND
             (MSG (CENTERPRINTINREGION MSG (DSPCLIPPINGREGION NIL PROMPTWINDOW)
                         PROMPTWINDOW])
```

(CloseFileMenus

```
  [LAMBDA (fileName)                                               (* cht%: " 8-SEP-82 23:53")
```

(* * Closes all windows used for displaying the COMS of fileName.)

```
    (PROG (fileInfo)
          (SETQ fileInfo (for fileInfo in menuedFiles thereis (EQ (CAR fileInfo)
                                                                  fileName)))
          (for window in (CADR fileInfo) do (CLOSEW window))
          (SETQ menuedFiles (REMOVE fileInfo menuedFiles])

)
```

(* * Fns to support WindowShade feature.)

```
(RPAQQ WINDOWSHADEFNS (MakeWindowShade MoveShadeFn ReshapeShadeFn UnMakeWindowShade WindowShadeButtonFn))

(DEFINEQ
```

(MakeWindowShade

```
  [LAMBDA (window)                                                 (* dgb%: "11-AUG-82 11:24")
```

(* Create an special window with only a title -- taken from the main window.
When this small window is buttoned, it opens the main window and runs its BUTTONFN.
The region for the small window contains the title of the given window plus a small white space below.)

```
    [COND
       ((NULL window)                                              (* If window not given, use the one at the cursor)
        (SETQ window (WHICHW)))
       ((EQ window T)
        (SETQ window (SELECTW]
    (COND
       ((NOT (WINDOWP window))
        (ERROR window "Not a window")))
    (PROG [iconWindow (wregion (WINDOWPROP window 'REGION))
                  (title (WINDOWPROP window 'TITLE]
          (SETQ iconWindow (CREATEW (create REGION
                                            LEFT _ (fetch LEFT of wregion)
                                            BOTTOM _ (IPLUS (fetch BOTTOM of wregion)
                                                            (fetch HEIGHT of wregion)
                                                            -20)
                                            HEIGHT _ 20
                                            WIDTH _ (fetch WIDTH of wregion))
                                    title))
          (WINDOWPROP window 'IconWindow iconWindow)
          (WINDOWPROP iconWindow 'IconFor window)
          (WINDOWPROP iconWindow 'BUTTONEVENTFN 'WindowShadeButtonFn)
          (WINDOWPROP iconWindow 'MOVEFN 'MoveShadeFn)
          (WINDOWPROP iconWindow 'RESHAPEFN 'ReshapeShadeFn)
          (WINDOWPROP iconWindow 'CLOSEFN 'UnMakeWindowShade)
          (CLOSEW window])
```

(MoveShadeFn

```
  [LAMBDA (iconWindow pos)                                         (* dgb%: " 5-JUN-83 22:06")
                                                                   (* Makes sure that mainWindow moves right along with
                                                                   windowShade)
    (PROG [mainR (wr (WINDOWPROP iconWindow 'REGION))
                 (mainW (WINDOWPROP iconWindow 'IconFor]
```

```
          (SETQ mainR (WINDOWPROP mainW 'REGION))
          (MOVEW mainW (IPLUS (fetch LEFT of mainR)
                              (IDIFFERENCE (fetch XCOORD of pos)
                                           (fetch LEFT of wr)))
                 (IPLUS (fetch BOTTOM of mainR)
                        (IDIFFERENCE (fetch YCOORD of pos)
                                     (fetch BOTTOM of wr])
```

(ReshapeShadeFn

```
  [LAMBDA (shadeWindow bitMap region)                    (* dgb%: " 2-AUG-82 00:32")
                                                         (* Makes sure that mainWindow is reshaped insead of the
                                                         windowShade)

    (PROG [(r (WINDOWPROP shadeWindow 'REGION))
           (w (WINDOWPROP shadeWindow 'IconFor]           (* Open code of close of IconWindow to avoid interactions)
          (CLOSEW shadeWindow)
          (SHAPEW w r)
          (MakeWindowShade w])
```

(UnMakeWindowShade

```
  [LAMBDA (shade)                                        (* dgb%: " 2-AUG-82 02:05")
                                                         (* Closefn for MenuShades)
    (OR shade (SETQ shade (WHICHW)))
    (PROG [(w (WINDOWPROP shade 'IconFor]
          (OPENW w)
          (WINDOWPROP w 'CLOSEFN NIL)
          (WINDOWPROP w 'IconWindow NIL)                 (* Open window. Break forward link, and back link)
          (WINDOWPROP shade 'IconFor NIL)
          (RETURN w])
```

(WindowShadeButtonFn

```
  [LAMBDA (windowShade)                                  (* dgb%: " 4-JUN-82 07:14")
                                                         (* Open the main window, run its button fn, close it and return)
    (PROG [(mainWindow (OPENW (WINDOWPROP windowShade 'IconFor]
          (ADJUSTCURSORPOSITION 0 -20)                   (* Move cursor down so it is sure to be in the mainWindow)
          (APPLY* (WINDOWPROP mainWindow 'BUTTONEVENTFN)
                 mainWindow)
          (CLOSEW mainWindow])

)


          (* * These fns would probably not be called by a user.)

(RPAQQ InternalTMENUFNS (AddItem CEILING ComputeMenuItems DInsert DeleteItem FetchNames InsertItem MenuButtonFn
                               NewMenuExpr PrintMenuExpr ReShapeMenu SetUpDeleteItem SetUpInsertItem SortItems
                               UnreadExpr))

(DEFINEQ
```

(AddItem

```
  [LAMBDA (window menu)                                  (* ct%: "14-JUN-82 07:22")

          (* * Adds an item from the terminal stream to the menu in the window.)

    (PROG (item)
          (CLEARBUF)
          (PROMPT "Please type in new menu item." "(Either an atom or a list of form: (printThis evalThis
                 comment trailerStr))")
          (SETQ item (OR (READ)
                         lastDeletedItem))
          (CLEARBUF)
          (CPROMPT)
          (replace ITEMS of menu with (NCONC1 (fetch ITEMS of menu)
                                              item))
          (ReShapeMenu window])
```

(CEILING

```
  [LAMBDA (fnum)                                         (* ct%: " 8-DEC-81 14:15")

          (* * Returns the minimum integer greater or equal to a number.)

    (PROG (num)
          (SETQ num (FIX fnum))
          (RETURN (COND
                     ((LESSP num fnum)
                      (ADD1 num))
                     (T num])
```

(ComputeMenuItems

```
  [LAMBDA (window menu)                                  (* ct%: "13-JUN-82 00:36")

          (* * Use the expression associated with a menu to recompute the list of items.)
```

```
    (PROG (expr items)
          (SETQ expr (WINDOWPROP window 'MENUEXPR))
          (COND
             [expr (SETQ items (EVAL expr))
                   (COND
                      ((OR (LISTP items)
                           (NULL items))
                       (replace ITEMS of menu with items)
                       (ReShapeMenu window NIL NIL menu))
                      (T (CPROMPT "Menu expression returns non-list."]
             (T (CPROMPT "No expression for this menu."])
```

(DInsert

```
  [LAMBDA (newItem oldItem List)                                    (* ct%: " 8-DEC-81 14:15")
```

(* * Destructively inserts newItem before alloccurences of oldItem in List.)

```
    (PROG (TempList)
          (SETQ TempList (LSUBST (LIST newItem oldItem)
                                 oldItem List))
          (RPLACA List (CAR TempList))
          (RPLACD List (CDR TempList))
          (RETURN List])
```

(DeleteItem

```
  [LAMBDA (item menu button)                                        (* mjs%: "11-FEB-82 17:33")
```

(* * Used to delete an item from a menu.)

```
    (PROG (items itemExpr)
          (SETQ lastDeletedItem item)
          (replace ITEMS of menu with (DREMOVE item (fetch ITEMS of menu)))
          (CPROMPT)
          (ReShapeMenu (WFROMMENU menu)
                 NIL NIL menu)
          (replace WHENSELECTEDFN of menu with (FUNCTION UnreadExpr])
```

(FetchNames

```
  [LAMBDA (lst)                                                     (* hgb%: " 4-JUN-82 13:22")
    (for x in lst collect (COND
                            ((LISTP x)
                             (CAR x))
                            (T x])
```

(InsertItem

```
  [LAMBDA (listItem menu button)                                    (* ct%: "14-JUN-82 07:24")
```

(* * Inserts a newItem from the terminal stream to the menu just before the selected listItem.
Assumes list has no duplicates.)

```
    (PROG (newItem newList)
          (CLEARBUF)
          (PROMPT "Please type in new item." "(Either an atom, or a list of form: (printThis evalThis comment
                 trailerStr))")
          (SETQ newItem (OR (READ)
                            lastDeletedItem))
          (CLEARBUF)
          (CPROMPT)
          (replace ITEMS of menu with (DInsert newItem listItem (fetch ITEMS of menu)))
          (ReShapeMenu (WFROMMENU menu)
                 NIL NIL menu)
          (replace WHENSELECTEDFN of menu with (FUNCTION UnreadExpr])
```

(MenuButtonFn

```
  [LAMBDA (window)                                                  (* mjs%: " 7-DEC-82 13:26")
```

(* * Called when LEFT or MIDDLE button depressed inside a window.
Routes action to MENU.HANDLER for LEFT and to MenuActionFn if MIDDLE.)

```
    (PROG (menu selection)
          [SETQ menu (CAR (WINDOWPROP window 'MENU]
          (TOTOPW window)
          (COND
             [(LASTMOUSESTATE LEFT)                                 (* Here to select item. Use standard menu package functions.)
              (COND
                 ([SETQ selection (MENU.HANDLER menu (WINDOWPROP window 'DSP]
                  (DOSELECTEDITEM menu (CAR selection)
                         (CDR selection]
             ((LASTMOUSESTATE MIDDLE)                               (* Here to process AddItem or DeleteItem action.)
              (SELECTQ (MENU (SETQ YellowButtonMenu (create MENU
                                                            ITEMS _ YellowButtonItems)))
```

```
                  (AddItem (AddItem window menu))
                  (DeleteItem (SetUpDeleteItem window menu))
                  (SortItems (SortItems window menu))
                  (InsertItem (SetUpInsertItem window menu))
                  (UseExpr (ComputeMenuItems window menu))
                  (NewExpr (NewMenuExpr window menu))
                  (PrintExpr (PrintMenuExpr window menu))
                  NIL])
```

(NewMenuExpr

```
  [LAMBDA (window menu)                                    (* ct%: " 8-DEC-81 14:15")
```

(* * Set a new expression for computing the menu items.)

```
    (PROG (EXPR)
          (CPROMPT "Enter New Expression for computing Menu items.")
          (SETQ EXPR (READ))
          (WINDOWPROP window 'MENUEXPR EXPR)
          (CPROMPT)
          (ComputeMenuItems window menu])
```

(PrintMenuExpr

```
  [LAMBDA (window menu)                                    (* ct%: " 8-DEC-81 14:15")
```

(* * Print the expression for computing the items of this menu.)

```
    (PROG (EXPR)
          (SETQ EXPR (WINDOWPROP window 'MENUEXPR))
          (COND
             (EXPR (PRINT EXPR))
             (T (CPROMPT "No Expression Set for this Menu")))
          (CLEARBUF])
```

(ReShapeMenu

```
  [LAMBDA (window oldImageBm oldRegion menu)               ; Edited  8-Nov-88 17:04 by jtm:
```

(* * Used to reshape menus created by TMenu. Tries to choose menuRows and menuColumns appropriately, and to adjust itemHeight and itemWidth so that the menu fits nicely in the window.)

```
    (PROG (menuColumns menuRows width height items itemWidth itemHeight font numItems clipRegion
                 oldButtonEventFn)
          [COND
             ((NULL menu)
              (SETQ menu (CAR (WINDOWPROP window 'MENU]
          (SETQ clipRegion (DSPCLIPPINGREGION NIL window))
          (SETQ width (fetch WIDTH of clipRegion))
          (SETQ height (fetch HEIGHT of clipRegion))
```

(* * Compute itemWidth to be the widest printing item in the menu.
Allow 2 points extra spacing.)

```
          (SETQ font (fetch (MENU MENUFONT) of menu))
          [SETQ items (for item in (fetch ITEMS of menu) collect (COND
                                                                    ((NLISTP item)
                                                                     item)
                                                                    (T (CAR item]
          (SETQ itemWidth (IPLUS (MAXSTRINGWIDTH items font)
                                 2))
```

(* * Compute menuColumns to be the ratio of the window WIDTH to the itemWidth, but no more than the number of items in the menu and at least one.)

```
          (SETQ numItems (FLENGTH items))
          (SETQ menuColumns (MAX (MIN (IQUOTIENT width itemWidth)
                                      numItems)
                                 1))
```

(* * Given menuColumns, adjust itemWidth so that the items will exactly fill the window.)

```
          (SETQ itemWidth (IQUOTIENT width menuColumns))
          (SETQ menuRows (CEILING (FQUOTIENT numItems menuColumns)))
```

(* * Compute itemHeight to be the ratio of the window height to the number of rows, but at least the height of the font.)

```
          [SETQ itemHeight (IMAX (IQUOTIENT height menuRows)
                                 (FONTPROP font 'HEIGHT]          (* Recompute menuRows in case they won't fit.)
          (SETQ menuRows (CEILING (FQUOTIENT height itemHeight)))
          (COND
             ((AND (EQP menuRows 2)
                   (IGREATERP numItems 3)
                   (IGREATERP (IQUOTIENT numItems 2)
                          (IDIFFERENCE numItems menuColumns)))
```

(* * Bias row and column configuration to prevent widow rows. Only for 2 row menus having fewer than half the items in the second row.)

```
            (SETQ menuColumns (CEILING (FQUOTIENT numItems 2)))
            (SETQ itemWidth (IQUOTIENT width menuColumns)))
           ((AND [IGREATERP height (ITIMES numItems (FONTPROP font 'HEIGHT]
                 (OR (GREATERP (QUOTIENT itemHeight itemWidth)
                               numItems)
                     (EQ menuRows 1)))
```

(* * Bias choice to vertical column menu if it is feasible.)

```
            (SETQ menuColumns 1)
            (SETQ menuRows numItems)
            (SETQ itemHeight (IQUOTIENT height numItems))
            (SETQ itemWidth width)))
```

(* * Smash values in the menu record.)

```
(replace MENUCOLUMNS of menu with menuColumns)
(replace ITEMHEIGHT of menu with itemHeight)
(replace ITEMWIDTH of menu with itemWidth)
(replace MENUROWS of menu with NIL)
(replace MENUOUTLINESIZE of menu with 0)
```

(* * This hack forces the menu package to remove the window, observe the changed parameters in the menu record, and re-display the menu. It also compensates for some window properties smashed by ADDMENU%: RESHAPEFN and BUTTONEVENTFN.)

```
(SETQ oldButtonEventFn (WINDOWPROP window 'BUTTONEVENTFN))
(DELETEMENU menu)
(UPDATE/MENU/IMAGE menu)                                    (* Make menu scrollable if needed.)
(WINDOWPROP window 'SCROLLFN NIL)
(ADDMENU menu window (create POSITION
                             XCOORD _ 0
                             YCOORD _ 0)
       (IGREATERP numItems (ITIMES menuRows menuColumns)))
(WINDOWPROP window 'RESHAPEFN (FUNCTION ReShapeMenu))
(WINDOWPROP window 'BUTTONEVENTFN (APPLY 'FUNCTION (LIST oldButtonEventFn)))
(RETURN window])
```

(SetUpDeleteItem

```
  [LAMBDA (window menu)                                     (* ct%: " 8-DEC-81 14:15")
```

(* * Temporarily replaces WHENSELECTEDFN so that selection causes deletion of item from menu.)

```
    (replace WHENSELECTEDFN of menu with (FUNCTION DeleteItem))
    (CPROMPT "Please select menu item to be deleted."])
```

(SetUpInsertItem

```
  [LAMBDA (window menu)                                     (* ct%: " 8-DEC-81 14:15")
```

(* * Temporarily replaces WHENSELECTEDFN so that selection causes insertion of new item just before selected item.)

```
    (replace WHENSELECTEDFN of menu with (FUNCTION InsertItem))
    (CPROMPT "Select item to insert new item before."])
```

(SortItems

```
  [LAMBDA (window menu)                                     (* ct%: " 8-DEC-81 14:15")
                                                            (* Sort the items in the menu.)
    (replace ITEMS of menu with (SORT (fetch ITEMS of menu)))
    (ReShapeMenu window NIL NIL menu])
```

(UnreadExpr

```
  [LAMBDA (exp menu)                                        (* mjs%: " 7-DEC-82 13:36")
```

(* * Items have the following fields%: (printx evalx commentx trailx) where -
printx is what appears in the menu. -
evalx is an expression to be evaluated if item is selected. -
commentx appears if item is red buttoned for a minimum time. -
trailx is the character printed after the item in the print stream. -
every field except print is optional. Default trail the defaultTrailingString argument specified when the TMenu was created if specified. Otherwise it is a space if print is atom, and the empty string otherwise.)

```
    (PROG (printx evalx trailx)
          (SETQ trailx (WINDOWPROP (WFROMMENU menu)
                                   'DEFAULTTRAILERSTRING))
          [COND
             ((NLISTP exp)                                  (* Here item is just an atom.)
              (SETQ printx exp)
              (SETQ trailx (OR trailx " ")))
             (T                                             (* Here when item is a list.)
                (SETQ printx (CAR exp))
                [COND
                   ((SETQ evalx (CADR exp))
```

```
                    (SETQ printx (EVAL evalx]
                 (COND
                    ((CADDDR exp)
                     (SETQ trailx (CADDDR exp)))
                    ((NLISTP printx)
                     (SETQ trailx (OR trailx " "]
           (BKSYSBUF printx)
           (COND
              (trailx (BKSYSBUF trailx])

)

(RPAQQ YellowButtonItems
       ((AddItem 'AddItem "Add item to menu")
        (DeleteItem 'DeleteItem "Delete item from menu")
        (InsertItem 'InsertItem "Insert item in menu")
        (SortItems 'SortItems "Sort items in menu")
        (UseExpr 'UseExpr "Use itemExpr to recompute item list for menu")
        (NewExpr 'NewExpr "Used to enter a new expression for computing items on menu")
        (PrintExpr 'PrintExpr "Prints the current itemExpr")))

(RPAQQ YellowButtonMenu NIL)

(RPAQQ firstCallFlgTmenu T)

(RPAQQ lastDeletedItem NIL)

(RPAQQ menuedFiles NIL)


          (* * Display Utility Functions)

(DEFINEQ

(SELECTW
  [LAMBDA NIL                                                    (* dgb%: " 7-FEB-83 14:17")
    (PROG NIL
          (PROMPT "Move mouse to desired window.
                 then press down the CTRL key or click mouse")
      LP  [COND
             ((OR (KEYDOWNP 'CTRL)
                  (NOT (MOUSESTATE UP)))
              (GETMOUSESTATE)
              (PROMPT)
              (RETURN (WHICHW]
          (GO LP])

(FLIPREGION
  [LAMBDA (DSP REGION)                                           (* dgb%: "24-JUN-82 00:01")

          (* Complement bits in region in DSP. If only DSP is given, complement the window or the DSP)

    [COND
       ((NULL REGION)
        (SETQ REGION (DSPCLIPPINGREGION NIL DSP]
    (BITBLT NIL NIL NIL DSP (fetch LEFT of REGION)
           (fetch BOTTOM of REGION)
           (fetch WIDTH of REGION)
           (fetch HEIGHT of REGION)
           'TEXTURE
           'INVERT BLACKSHADE])

)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA CPROMPT PROMPT)
)

(PUTPROPS TMENU COPYRIGHT ("Xerox Corporation" 1987 1988))
```

FUNCTION INDEX

VARIABLE INDEX