File created: 1-Feb-2022 17:09:01 **{DSK}<Users>kaplan>Local>medley3.5>my-medley>lispusers>FM-CREATOR.;2**

changes to: (FNS FMC-EDIT.INFO)

previous date: 18-Aug-88 14:32:54 {DSK}<Users>kaplan>Local>medley3.5>my-medley>lispusers>FM-CREATOR.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
;;
;; Copyright (c) 1988 by Rank Xerox France. Author Andre BLAVIER.

(RPAQQ FM-CREATORCOMS
       ((PROP MAKEFILE-ENVIRONMENT FM-CREATOR)
        (* * FMC items record)
        (RECORDS FMC-ITEM)
        (* * FMC creation functions)
        (FNS FMC-CREATE FMC-INSTALL.IP.WINDOW FMC-INSTALL.GP.WINDOW FMC-CREATE.SHADE.MENU FMC-CREATE.SHADE.ITEM)
        (* * FMC windows functions)
        (FNS FMC-BUTTONEVENTFN FMC-CLOSEFN FMC-COPYBUTTONEVENTFN FMC-COPYINSERTFN FMC-CURSORMOVEDFN
             FMC-CURSOROUTFN FMC-EXPANDFN FMC-ICONFN FMC-SHRINKFN FMC-WINDOWENTRYFN)
        (* * FMC macros)
        (FUNCTIONS FM-GET.ITEM.LABEL FM-GET.ITEM.STATE FMC-CLEAR.REGION FMC-GET.ITEM FMC-GROUP?
               FMC-MARK.AS.CHANGED FMC-PUT.ITEM GET.REGION.BOTTOM GET.REGION.HEIGHT GET.REGION.LEFT
               GET.REGION.WIDTH NULLSTR PUT.REGION.BOTTOM PUT.REGION.HEIGHT PUT.REGION.LEFT PUT.REGION.WIDTH)
        (* * Right menu functions)
        (FNS FMC-FIXRIGHTMENU FMC-DORIGHTSELECTION)
        (* * Selection functions)
        (FNS FMC-SELECT.ITEM FMC-SELECT.LIST FMC-SELECT.LIST.ITEM FMC-SELECTALL FMC-GET.SELECTION FMC-DESELECT
             FMC-DESELECT.ITEM FMC-DESELECT.LIST FMC-DESELECT.LIST.ITEM)
        (* * Property windows functions)
        (FNS FMC-APPLY FMC-SHOW.ITEM FMC-SHOW.GROUP FMC-NEWITEM FMC-UPDATE.ITEM FMC-UPDATE.GROUP FMC-GET.LABEL
             FMC-CHOOSE.ITEM.BOXSHADE FMC-CHOOSE.ITEM.BG FMC-GET.MENUPROPS FMC-GET.INITSTATE FMC-EDIT.FN
             FMC-LINKS)
        (* * Property windows descriptions)
        (VARS FMC-IP-DESC FMC-GP-DESC)
        (* * Creating bitmaps)
        (FNS FMC-MAKEBITMAP FMC-COMPOUND.BITMAP FMC-SNAPBM)
        (* * Moving items)
        (FNS FMC-MOVE.SELECTION FMC-MOVE.BITMAP FMC-TRACK.NEW.ITEM FMC-UPDATE.BM.POSITION FMC-UPDATE.REGION)
        (* * Shaping items)
        (FNS FMC-COMPUTE.SHAPE.REGS FMC-SHAPE FMC-BOX.NEWREGIONFN FMC-NOBOX.NEWREGIONFN)
        (* * Redrawing items)
        (FNS FMC-REDRAW FMC-REDRAW.ITEM)
        (* * Deleting and undeleting items)
        (FNS FMC-DELETE FMC-UNDELETE)
        (* * GROUPing and UNGROUPing)
        (FNS FMC-GROUP FMC-UNGROUP)
        (* * Align and Center functions)
        (FNS FMC-ALIGN FMC-HCENTER FMC-VCENTER FMC-REL.MOVE)
        (* * File saving and loading)
        (FNS FMC-GET FMC-GET.ONE.OBJECT FMC-PUT FMC-PUT.OBJECT)
        (* * Creating a summary)
        (FNS FMC-EDIT.INFO FMC-EDIT.INFO.ITEM)
        (* * Hardcopy functions)
        (FNS FMC-HARDCOPY FMC-HARDCOPY.ITEM)
        (* * Creating the desription list)
        (FNS FMC-COMPUTE FMC-COMPUTE.OBJECT)
        (* * Miscellaneous)
        (FNS FMC-CREATE.ITEM.FROM.LIST FMC-DRAW.BOX FMC-CHOOSE.WINDOW.BG FMC-DISPLAY.GRID FMC-SET.GRIDSIZE
             FMC-FONT->LIST FMC-LIST->FONT FMC-SORT.ITEM.LIST FMC-IMPORT FMC-PROMPTPRINT)
        (* * Icon stuff)
        [COMS (BITMAPS FMC-ICON FMC-ICON.MASK)
              (INITVARS (FMC-ICON.TEMPLATE (create TITLEDICON ICON _ FMC-ICON MASK _ FMC-ICON.MASK TITLEREG _
                                                   (CREATEREGION 2 2 70 28]
        (* *)
        (COMS (P [OR (SASSOC 'FMCreator BackgroundMenuCommands)
                     (NCONC1 BackgroundMenuCommands '(FMCreator '(FMC-CREATE)
                                                                 "Opens a Free Menu Creator window for use"]
                 (SETQ BackgroundMenu NIL)))
        (CURSORS MOVINGCURSOR)))

(PUTPROPS FM-CREATOR MAKEFILE-ENVIRONMENT (:READTABLE "INTERLISP" :PACKAGE "INTERLISP"))
```

(* * FMC items record)

```
(DECLARE%: EVAL@COMPILE

(RECORD FMC-ITEM
       (TYPE LABEL FONT ID COLLECTION DESELECT MESSAGE INITSTATE BOX BOXSHADE BACKGROUND LINKS OBJECTLIST MENU
             CHANGESTATE SELECTEDFN DOWNFN HELDFN MOVEDFN REGION BITMAP XBMOFFSET YBMOFFSET INFINITEWIDTH
```

```
            USERDATA)
       XBMOFFSET _ 0 YBMOFFSET _ 0)
)


          (* * FMC creation functions)

(DEFINEQ

(FMC-CREATE
  [LAMBDA NIL                                                    ; Edited 18-Aug-88 12:21 by A.BLAVIER
    ;; Create a Free-Menu Creator window.
    (LET ((Window (CREATEW (LIST LASTMOUSEX LASTMOUSEY 400 200)
                      "Free Menu Creator" 4 T))
          (RightMenu (create MENU
                          ITEMS _
                          `((Redraw 'REDRAW "Redraw the FMC window")
                            [Grid 'GRID "Grid"
                                  (SUBITEMS ("No Grid" 'NOGRID)
                                         (Size 'SIZE "Specify a grid size"
                                               (SUBITEMS 2 3 4 5 6 7 8 9 10))
                                         ("Display Grid" 'DISPLAYGRID)
                                         ("Remove Grid Display" 'REMOVEGRIDDISPLAY]
                            ("Delete" 'DELETE "Delete selected item(s)" (SUBITEMS ("Forget save list"
                                                                                  'FORGET.SAVE.LIST
                                                                                  "Forget the list of deleted
                                                                                  items")))
                            ("Undelete" 'UNDELETE.LAST "Undelete last item" (SUBITEMS (Last 'UNDELETE.LAST
                                                                                           "Undelete last
                                                                                           item")
                                                                                  (All 'UNDELETE.ALL
                                                                                       "Undelete all deleted
                                                                                       item(s)")
                                                                                  (List 'UNDELETE.LIST
                                                                                        "Undelete one item"))
                                   )
                            (Group 'GROUP "Group the selected items")
                            (Ungroup 'UNGROUP "Ungroup the selection")
                            [Align 'ALIGN "Align selected items" (SUBITEMS ("Left sides" 'ALIGNLEFT)
                                                                           ("Right sides" 'ALIGNRIGHT)
                                                                           ("Tops" 'ALIGNTOP)
                                                                           ("Bottoms" 'ALIGNBOTTOM]
                            [Center 'CENTER "Center items" (SUBITEMS (Horizontally 'HCENTER)
                                                                     (Vertically 'VCENTER]
                            ("Select All" 'SELECTALL "Select all the items")
                            ("Background" 'WBACKGROUND "Set the window's background")
                            ("Summary" 'SUMMARY "Create a summary of the items")
                            (Import 'IMPORT "Import items from a Free Menu")
                            (Compute 'COMPUTE "Create the Free Menu description")
                            (Get 'GET "Get items from a file")
                            (Put 'PUT "Put items onto a file")
                            ("Fixed Menu" 'FIXRIGHTMENU "Attach Right Menu"))
                          CENTERFLG _ T))
          FMC.PROMPTWINDOW)
         (PRINTOUT PROMPTWINDOW "FREE MENU CREATOR" T "Copyright (c) 1988 by Rank Xerox France. Author Andre
                BLAVIER")
         (SETCURSOR WAITINGCURSOR)
         (PUTMENUPROP RightMenu 'FMC.WINDOW Window)
    ;; Initialize window properties
         (WINDOWPROP Window 'WINDOWENTRYFN (FUNCTION FMC-WINDOWENTRYFN))
         (WINDOWPROP Window 'BUTTONEVENTFN (FUNCTION FMC-BUTTONEVENTFN))
         (WINDOWPROP Window 'RIGHTBUTTONFN (FUNCTION FMC-BUTTONEVENTFN))
         (WINDOWPROP Window 'CURSORMOVEDFN (FUNCTION FMC-CURSORMOVEDFN))
         (WINDOWPROP Window 'CURSOROUTFN (FUNCTION FMC-CURSOROUTFN))
         (WINDOWPROP Window 'REPAINTFN (FUNCTION FMC-REDRAW))
         (WINDOWPROP Window 'RESHAPEFN (LIST (FUNCTION REPOSITIONATTACHEDWINDOWS)
                                             (FUNCTION FMC-REDRAW)))
         (WINDOWPROP Window 'HARDCOPYFN (FUNCTION FMC-HARDCOPY))
         (WINDOWPROP Window 'COPYBUTTONEVENTFN (FUNCTION FMC-COPYBUTTONEVENTFN))
         (WINDOWPROP Window 'COPYINSERTFN (FUNCTION FMC-COPYINSERTFN))
         (WINDOWPROP Window 'CLOSEFN (FUNCTION FMC-CLOSEFN))
         (WINDOWPROP Window 'SHRINKFN (FUNCTION FMC-SHRINKFN))
         (WINDOWPROP Window 'ICONFN (FUNCTION FMC-ICONFN))
         (WINDOWPROP Window 'EXPANDFN (FUNCTION FMC-EXPANDFN))
         (WINDOWPROP Window 'RIGHTMENU RightMenu)
         (WINDOWPROP Window 'FMC.CHANGED NIL)                    ; this flag will be set when any item changing occurs - it is
                                                                 ; checked before window closing
    ;; Create a prompt window
         (SETQ FMC.PROMPTWINDOW (CREATEW [CREATEREGION 0 0 100 (HEIGHTIFWINDOW (FONTPROP (DEFAULTFONT
                                                                                          'DISPLAY)
                                                                                  'HEIGHT]
                                         NIL NIL T))
         (WINDOWPROP FMC.PROMPTWINDOW 'PAGEFULLFN (FUNCTION NILL))
         [WINDOWPROP FMC.PROMPTWINDOW 'MAXSIZE (CONS MAX.SMALLP (GET.REGION.HEIGHT (WINDOWPROP FMC.PROMPTWINDOW
```

```
                                                                                    'REGION]
        (DSPSCROLL 'ON FMC.PROMPTWINDOW)
        (ATTACHWINDOW FMC.PROMPTWINDOW Window 'TOP)
        (WINDOWPROP FMC.PROMPTWINDOW 'PASSTOMAINCOMS '(MOVEW SHAPEW SHRINKW BURYW))
        (WINDOWPROP Window 'FMC.PROMPTWINDOW FMC.PROMPTWINDOW)
        ;;
        (MOVEW Window)
        ;; Create and attach the Item and Group Properties windows
        (SETCURSOR WAITINGCURSOR)
        (WINDOWPROP Window 'FMC.IP.WINDOW (FMC-INSTALL.IP.WINDOW Window))
        (WINDOWPROP Window 'FMC.GP.WINDOW (FMC-INSTALL.GP.WINDOW Window))
        ;; Spawn a process
        (ADD.PROCESS `(PROGN (TTYDISPLAYSTREAM ,Window)
                             (UNTIL (WINDOWPROP ,Window 'CANCLOSE) DO (BLOCK)))
               'NAME
               'FMCreator
               'WINDOW Window)
        ;;
        (SETCURSOR DEFAULTCURSOR])
```

(FMC-INSTALL.IP.WINDOW

```
  [LAMBDA (WINDOW)                                          ; Edited 16-Aug-88 14:17 by A.BLAVIER
    ;; Create and attach the Item Properties window
    (PROG ((FMC.IP.Window (FREEMENU FMC-IP-DESC "ITEM PROPERTIES"))
           (DefaultFont (DEFAULTFONT 'DISPLAY))
           FontList ShadeMenu BlackBitmap WhiteBitmap)
     ;; Initialize Font to DEFAULTFONT
          (SETQ FontList (FMC-FONT->LIST DefaultFont))
          (FM.CHANGESTATE 'FAMILY (CAR FontList)
                 FMC.IP.Window)
          (FM.CHANGESTATE 'SIZE (CADR FontList)
                 FMC.IP.Window)
          (FM.CHANGESTATE 'FACE (CADDR FontList)
                 FMC.IP.Window)
     ;; the LABEL and MESSAGE edit fields are initially empty
          (FM.CHANGELABEL 'LABELLINK "" FMC.IP.Window)
          (FM.CHANGELABEL 'MESSAGELINK "" FMC.IP.Window)
     ;; Create the shade menu
          (SETQ ShadeMenu (FMC-CREATE.SHADE.MENU))
          (SETQ BlackBitmap (CAAR (fetch (MENU ITEMS) of ShadeMenu)))
                                                            ; the blackshade bitmap is the first menu item
          [SETQ WhiteBitmap (CAAR (NTH (fetch (MENU ITEMS) of ShadeMenu)
                                       (SUB1 (LENGTH (fetch (MENU ITEMS) of ShadeMenu]
                                                            ; the whiteshade BM is the one before last
          (FM.CHANGESTATE 'BOXSHADE BlackBitmap FMC.IP.Window)
          (FM.CHANGESTATE 'BACKGROUND WhiteBitmap FMC.IP.Window)
     ;; initialize window properties
          (WINDOWPROP FMC.IP.Window 'SHADE.MENU ShadeMenu)
          (WINDOWPROP FMC.IP.Window 'FMC.ITEM.BOXSHADE BLACKSHADE)
          (WINDOWPROP FMC.IP.Window 'FMC.ITEM.BACKGROUND WHITESHADE)
          [for PROP in '(CHANGESTATE SELECTEDFN DOWNFN HELDFN MOVEDFN)
             do (WINDOWPROP FMC.IP.Window (PACK* 'FMC.ITEM. PROP)
                        '(FUNCTION NILL]
          (WINDOWPROP FMC.IP.Window 'FMC.ITEM.MENUPROPS '(NIL))
     ;; attach the IP window
          (ATTACHWINDOW FMC.IP.Window WINDOW 'TOP 'LEFT)
          (WINDOWPROP FMC.IP.Window 'MAIN.WINDOW WINDOW)
          (WINDOWPROP FMC.IP.Window 'RESHAPEFN 'DON'T)
          (RETURN FMC.IP.Window])
```

(FMC-INSTALL.GP.WINDOW

```
  [LAMBDA (WINDOW)                                          ; Edited 16-Aug-88 14:19 by A.BLAVIER
    ;; Create and attach the Group Properties window
    (PROG ((FMC.GP.Window (FREEMENU FMC-GP-DESC "GROUP PROPERTIES"))
           (FMC.IP.Window (WINDOWPROP WINDOW 'FMC.IP.WINDOW))
           ShadeMenu BlackBitmap WhiteBitmap)
     ;; see comments of FMC-INSTALL.IP.WINDOW
          (SETQ ShadeMenu (WINDOWPROP FMC.IP.Window 'SHADE.MENU))
          (SETQ BlackBitmap (CAAR (fetch (MENU ITEMS) of ShadeMenu)))
          [SETQ WhiteBitmap (CAAR (NTH (fetch (MENU ITEMS) of ShadeMenu)
                                       (SUB1 (LENGTH (fetch (MENU ITEMS) of ShadeMenu]
          (FM.CHANGESTATE 'BOXSHADE BlackBitmap FMC.GP.Window)
          (FM.CHANGESTATE 'BACKGROUND WhiteBitmap FMC.GP.Window)
```

```
          (WINDOWPROP FMC.GP.Window 'FMC.ITEM.BOXSHADE BLACKSHADE)
          (WINDOWPROP FMC.GP.Window 'FMC.ITEM.BACKGROUND WHITESHADE)
          (WINDOWPROP FMC.GP.Window 'SHADE.MENU ShadeMenu)
          (ATTACHWINDOW FMC.GP.Window FMC.IP.Window 'RIGHT 'BOTTOM)
          (WINDOWPROP FMC.GP.Window 'MAIN.WINDOW WINDOW)
          (WINDOWPROP FMC.GP.Window 'RESHAPEFN 'DON'T)
          (RETURN FMC.GP.Window])
```

(FMC-CREATE.SHADE.MENU

```
  [LAMBDA NIL                                          ; Edited 16-Aug-88 14:21 by A.BLAVIER
    ;; Create a shade menu built out of the bitmaps corresponding to Shades.
    (LET ((Shades '(65535 43605 38505 52224 61713 5160 32768 0))
          ShadeItems)
         (SETQ ShadeItems (for s in Shades collect (FMC-CREATE.SHADE.ITEM s)))
         [NCONC1 ShadeItems '("OTHER" (CADADR (FMC-CREATE.SHADE.ITEM (EDITSHADE]
         (create MENU
                 ITEMS _ ShadeItems
                 ITEMWIDTH _ 60
                 CENTERFLG _ T
                 TITLE _ "SHADE"])
```

(FMC-CREATE.SHADE.ITEM

```
  [LAMBDA (SHADE)                                      ; Edited 16-Aug-88 14:24 by A.BLAVIER
    ;; Compute a shade menu item of the form : (BITMAP (BITMAP . SHADE))
    (PROG (Bitmap Stream bm)
          (SETQ Bitmap (BITMAPCREATE 60 12))
          (SETQ Stream (DSPCREATE Bitmap))
          (DSPFILL '(0 0 59 12)
                 SHADE
                 'REPLACE Stream)
          (DRAWLINE 0 0 0 11 1 'REPLACE Stream)
          (DRAWLINE 0 11 59 11 1 'REPLACE Stream)
          (DRAWLINE 59 11 59 0 1 'REPLACE Stream)
          (DRAWLINE 59 0 0 0 1 'REPLACE Stream)
          (SETQ bm (BITMAPCOPY Bitmap))
          (RETURN (LIST bm (KWOTE (CONS bm SHADE])

)


          (* * FMC windows functions)

(DEFINEQ
```

(FMC-BUTTONEVENTFN

```
  [LAMBDA (WINDOW)                                     ; Edited 16-Aug-88 14:33 by A.BLAVIER
    ;; This is the BUTTONEVENTFN attached to FMC windows.
    ;; In any case bring the window to top
    (TOTOPW WINDOW)
    (LET ((SelectedItem (WINDOWPROP WINDOW 'SELECTED.ITEM))
          (SelectionList (WINDOWPROP WINDOW 'SELECTION.LIST))
          Item Timer Reg)
         ;;
         ;; Deal with RIGHT button : pops up either FMC right menu or standard window menu
         (if (LASTMOUSESTATE RIGHT)
             then (if [OR (WINDOWPROP WINDOW 'FIXEDRIGHTMENU)
                          (IGREATERP (fetch (POSITION YCOORD) of (CURSORPOSITION NIL WINDOW))
                                     (WINDOWPROP WINDOW 'HEIGHT]
                      then (DOWINDOWCOM WINDOW)
                    else (FMC-DORIGHTSELECTION (MENU (WINDOWPROP WINDOW 'RIGHTMENU))
                                               WINDOW)))
         ;;
         ;; Deal with LEFT Button
         [if [AND (MOUSESTATE (AND LEFT (NOT MIDDLE)))
                  (ILESSP (fetch (POSITION YCOORD) of (CURSORPOSITION NIL WINDOW))
                          (WINDOWPROP WINDOW 'HEIGHT]
             then (if [SETQ Item (for i in (WINDOWPROP WINDOW 'ITEMLIST) thereis (INSIDEP (FMC-GET.ITEM i REGION)
                                                                                          (CURSORPOSITION NIL WINDOW]
                      then ;; user clicked inside an Item
                           [if (SHIFTDOWNP 'META)
                               then ;; user Meta-clicked
                                    (if (NOT (FMEMB Item SelectionList))
                                        then ;; if Item is not yet part of a multiple selection add it to the selection
                                             (WINDOWADDPROP WINDOW 'SELECTION.LIST Item)
                                             (FMC-SELECT.LIST.ITEM Item WINDOW)
                                             (if SelectedItem
```

```
                                        then ;; Item was the unique selection : make it the only item of a multiple selection
                                             (WINDOWADDPROP WINDOW 'SELECTION.LIST SelectedItem T)
                                             (FMC-DESELECT.ITEM WINDOW)
                                             (FMC-SELECT.LIST.ITEM SelectedItem WINDOW))
                                   else ;; remove Item from the multiple selection
                                        (FMC-DESELECT.LIST.ITEM Item WINDOW))
                              else (COND
                                     ((EQUAL Item SelectedItem)    ; Item is the unique selected item : move it
                                      (FMC-MOVE.SELECTION WINDOW 'UNIQUE)
                                      (FMC-COMPUTE.SHAPE.REGS WINDOW))
                                     ((FMEMB Item SelectionList)   ; Item is a member of a multiple selection : move the selection
                                      (FMC-MOVE.SELECTION WINDOW 'MULTIPLE))
                                     (T                            ; Item is not yet selected : make it the unique selection
                                        (FMC-DESELECT WINDOW)
                                        (SETQ Timer (SETUPTIMER 20000 NIL 'TICKS 'TICKS))
                                        (WINDOWPROP WINDOW 'SELECTED.ITEM Item)
                                        (FMC-SELECT.ITEM WINDOW)
                                        [until (OR (MOUSESTATE (NOT LEFT))
                                                   (TIMEREXPIRED? Timer 'TICKS]
                                        (if (MOUSESTATE LEFT)
                                            then (FMC-MOVE.SELECTION WINDOW 'UNIQUE))
                                        (FMC-COMPUTE.SHAPE.REGS WINDOW]
                         else ;; user clicked outside any item
                              (if (SHIFTDOWNP 'META)
                                  then ;; if Meta : select items inside a ghost region
                                       (FMC-GET.SELECTION WINDOW)
                                else ;; remove any selection
                                     (FMC-DESELECT WINDOW]
    ;;
    ;; Deal with MIDDLE Button
    [if [AND (MOUSESTATE (AND MIDDLE (NOT LEFT)))
             (ILESSP (fetch (POSITION YCOORD) of (CURSORPOSITION NIL WINDOW))
                     (WINDOWPROP WINDOW 'HEIGHT]
         then (if [SETQ Item (for i in (WINDOWPROP WINDOW 'ITEMLIST) thereis (INSIDEP (FMC-GET.ITEM i REGION)
                                                                                      (CURSORPOSITION NIL WINDOW]
                  then ;; user clicked inside an Item
                       (if (EQUAL Item SelectedItem)
                           then ;; shape the Item
                                (if [SETQ Reg (for reg in (WINDOWPROP WINDOW 'SHAPE.REGS)
                                                   thereis (INSIDEP (CAR reg)
                                                                    (CURSORPOSITION NIL WINDOW]
                                    then (SETCURSOR (CADDDR Reg))
                                         (APPLY (CADR Reg)
                                                (LIST (CADDR Reg)
                                                      WINDOW)))
                           else (SETCURSOR DEFAULTCURSOR]
    ;;
    ;; The following statement handles the case where the user released the left button in a selected item : adjust the cursor appropriately
    (if (MOUSESTATE (AND (NOT LEFT)
                         (NOT MIDDLE)
                         (NOT RIGHT)))
        then (FMC-CURSORMOVEDFN WINDOW])
```

(FMC-CLOSEFN

```
[LAMBDA (W)                                                    ; Edited 17-Aug-88 10:23 by A.BLAVIER
    ;; Attempt to close a FMC window.
    ;; Check the FMC.CHANGED property
    (if [AND (WINDOWPROP W 'FMC.CHANGED)
             (NOT (MOUSECONFIRM "Warning ! Not saved yet. LEFT to quit anyway" "" (WINDOWPROP W
                                                                                              'FMC.PROMPTWINDOW]
        then 'DON'T
      else ;; if OK then set the CANCLOSE prop, which is continually checked by the process's main loop
           (WINDOWPROP W 'CANCLOSE T])
```

(FMC-COPYBUTTONEVENTFN

```
[LAMBDA (W)                                                    ; Edited 16-Aug-88 14:39 by A.BLAVIER
    ;; Copy selected item(s) to another FMC window.
    ;; Check that the destination window (the window that currently has the TTY) is a FMC window
    (if (WINDOWPROP (PROCESSPROP (TTY.PROCESS)
                                 'WINDOW)
                    'FMC.PROMPTWINDOW)
```

```
        then ;; pass a list of the form : (FROM-FMC (selected items ...))
             (COPYINSERT (LIST 'FROM-FMC (COND
                                           [(WINDOWPROP W 'SELECTED.ITEM)
                                            (LIST (COPYALL (WINDOWPROP W 'SELECTED.ITEM]
                                           ((WINDOWPROP W 'SELECTION.LIST)
                                            (COPYALL (WINDOWPROP W 'SELECTION.LIST])
```

(FMC-COPYINSERTFN

```
[LAMBDA (OBJ WINDOW)                                    ; Edited 16-Aug-88 14:44 by A.BLAVIER
  ;; User is trying to insert some OBJect into the FMC window, using the COPY key.
  (COND
     ((IMAGEOBJP OBJ)
      ;; if OBJect is an IMAGEOBJ -> make LABEL be its bitmap
      (LET (ImageBox Bitmap Stream)
           (SETQ ImageBox (APPLY (IMAGEOBJPROP OBJ 'IMAGEBOXFN)
                                 (LIST OBJ WINDOW)))
           (SETQ Bitmap (BITMAPCREATE (fetch (IMAGEBOX XSIZE) of ImageBox)
                                      (fetch (IMAGEBOX YSIZE) of ImageBox)))
           (SETQ Stream (DSPCREATE Bitmap))
           (APPLY (IMAGEOBJPROP OBJ 'DISPLAYFN)
                  (LIST OBJ Stream 'DISPLAY))
           (FM.CHANGELABEL 'LABELLINK Bitmap (WINDOWPROP WINDOW 'FMC.IP.WINDOW)
                  T)))
     ((EQ (CAR OBJ)
          'FROM-FMC)
      ;;  if OBJect is a list of FMC item(s) (coming from another FMC window) -> add them to the window's ITEMLIST
      (for item in (CADR OBJ) do (FMC-REDRAW.ITEM item WINDOW)
                                 (WINDOWADDPROP WINDOW 'ITEMLIST item))
      (if (WINDOWPROP WINDOW 'SELECTED.ITEM)
          then (FMC-DESELECT.ITEM WINDOW))
      (if (WINDOWPROP WINDOW 'SELECTION.LIST)
          then (FMC-DESELECT.LIST WINDOW))
      (WINDOWPROP WINDOW 'SELECTION.LIST (CADR OBJ))
      (FMC-SELECT.LIST WINDOW)
      (FMC-MARK.AS.CHANGED WINDOW])
```

(FMC-CURSORMOVEDFN

```
[LAMBDA (WINDOW)                                        ; Edited 16-Aug-88 15:30 by A.BLAVIER
  ;; If the cursor is inside a selected item make it the MOVINGCURSOR, else make it the DEFAULTCURSOR.
  (LET (Item)
       (COND
          ((AND (SETQ Item (WINDOWPROP WINDOW 'SELECTED.ITEM))
                (INSIDEP (FMC-GET.ITEM Item REGION)
                         (CURSORPOSITION NIL WINDOW)))
           (SETCURSOR MOVINGCURSOR))
          ((for i in (WINDOWPROP WINDOW 'SELECTION.LIST) thereis (INSIDEP (FMC-GET.ITEM i REGION)
                                                                          (CURSORPOSITION NIL WINDOW)))
           (SETCURSOR MOVINGCURSOR))
          (T (SETCURSOR DEFAULTCURSOR])
```

(FMC-CURSOROUTFN

```
[LAMBDA (W)                                             ; Edited 16-Aug-88 14:06 by A.BLAVIER
  (SETCURSOR DEFAULTCURSOR])
```

(FMC-EXPANDFN

```
[LAMBDA (W)                                             ; Edited 16-Aug-88 15:44 by A.BLAVIER
  (if (WINDOWPROP W 'PROCESS)
      then ;; grab the TTY
           (TTY.PROCESS (WINDOWPROP W 'PROCESS])
```

(FMC-ICONFN

```
[LAMBDA (W)                                             ; Edited 16-Aug-88 15:42 by A.BLAVIER
  ;; Return a titled icon when shrinking a FMC window.
  ;; This function is extracted from the TEdit ICONFN.
  (PROG [(Icon (WINDOWPROP W 'ICON))
         (IconTitle (WINDOWPROP W 'FMC.ICON.TITLE))
         (WindowTitle (WINDOWPROP W 'TITLE]
        [COND
          ((OR (AND IconTitle (EQUAL IconTitle WindowTitle))
               (AND (NOT IconTitle)
                    Icon))
           NIL)
          (Icon (WINDOWPROP W 'FMC.ICON.TITLE (SETQ IconTitle WindowTitle))
                (ICONTITLE IconTitle NIL NIL Icon))
          (T (WINDOWPROP W 'FMC.ICON.TITLE (SETQ IconTitle WindowTitle))
```

```
                (WINDOWPROP W 'ICON (TITLEDICONW FMC-ICON.TEMPLATE IconTitle '(HELVETICA 8 STANDARD)
                                                 NIL T NIL 'FILE]
          (RETURN (WINDOWPROP W 'ICON])
```

(**FMC-SHRINKFN**

```
  [LAMBDA (W)                                                    ; Edited 16-Aug-88 15:45 by A.BLAVIER
    (if (AND (EQ (WINDOWPROP W 'PROCESS)
                 (TTY.PROCESS)))
        then ;; abandon the TTY
             (TTY.PROCESS T])
```

(**FMC-WINDOWENTRYFN**

```
  [LAMBDA (W)                                                    ; Edited 17-Aug-88 10:24 by A.BLAVIER
    ;; Grab the TTY when the mouse clicks in the window, and process the BUTTONEVENTFN.
    (if [AND [NOT (OR (SHIFTDOWNP 'SHIFT)
                      (SHIFTDOWNP 'META)
                      (KEYDOWNP 'MOVE)
                      (KEYDOWNP 'COPY]
             (PROCESSP (WINDOWPROP W 'PROCESS]
        then (TTY.PROCESS (WINDOWPROP W 'PROCESS))
             (FMC-BUTTONEVENTFN W])

)


          (* * FMC macros)

(DEFMACRO FM-GET.ITEM.LABEL (ID.OR.LABEL WINDOW)
   (LIST 'FM.ITEMPROP (LIST 'FM.GETITEM ID.OR.LABEL NIL WINDOW)
         ''LABEL))

(DEFMACRO FM-GET.ITEM.STATE (ID.OR.LABEL WINDOW)
   (LIST 'FM.ITEMPROP (LIST 'FM.GETITEM ID.OR.LABEL NIL WINDOW)
         ''STATE))

(DEFMACRO FMC-CLEAR.REGION (REGION WINDOW)
   (LIST 'DSPFILL REGION 'WHITESHADE ''REPLACE WINDOW))

(DEFMACRO FMC-GET.ITEM (ITEM FIELD)
   (LIST 'fetch `(FMC-ITEM ,FIELD)
         'of ITEM))

(DEFMACRO FMC-GROUP? (OBJECT)
   (LIST 'EQ `(FMC-GET.ITEM ,OBJECT TYPE)
         ''GROUP))

(DEFMACRO FMC-MARK.AS.CHANGED (W)
   (LIST 'WINDOWPROP W ''FMC.CHANGED T))

(DEFMACRO FMC-PUT.ITEM (ITEM FIELD VALUE)
   (LIST 'replace `(FMC-ITEM ,FIELD)
         'of ITEM 'with VALUE))

(DEFMACRO GET.REGION.BOTTOM (REGION)
   (LIST 'fetch '(REGION BOTTOM)
         'of REGION))

(DEFMACRO GET.REGION.HEIGHT (REGION)
   (LIST 'fetch '(REGION HEIGHT)
         'of REGION))

(DEFMACRO GET.REGION.LEFT (REGION)
   (LIST 'fetch '(REGION LEFT)
         'of REGION))

(DEFMACRO GET.REGION.WIDTH (REGION)
   (LIST 'fetch '(REGION WIDTH)
         'of REGION))

(DEFMACRO NULLSTR (STR)
   (LIST 'STREQUAL STR ""))
```

```
(DEFMACRO PUT.REGION.BOTTOM (REGION VALUE)
   (LIST 'replace '(REGION BOTTOM)
         'of REGION 'with VALUE))

(DEFMACRO PUT.REGION.HEIGHT (REGION VALUE)
   (LIST 'replace '(REGION HEIGHT)
         'of REGION 'with VALUE))

(DEFMACRO PUT.REGION.LEFT (REGION VALUE)
   (LIST 'replace '(REGION LEFT)
         'of REGION 'with VALUE))

(DEFMACRO PUT.REGION.WIDTH (REGION VALUE)
   (LIST 'replace '(REGION WIDTH)
         'of REGION 'with VALUE))
```

(* * Right menu functions)

```
(DEFINEQ

(FMC-FIXRIGHTMENU
  [LAMBDA (WINDOW)                                          ; Edited 25-Jul-88 11:16 by A.BLAVIER
    ;; Fix the right menu if not yet attached.
    (COND
       ((NOT (WINDOWPROP WINDOW 'FIXEDRIGHTMENU))
        (replace (MENU WHENSELECTEDFN) of (WINDOWPROP WINDOW 'RIGHTMENU) with (FUNCTION FMC-DORIGHTSELECTION))
        (LET [(MenuWindow (ATTACHMENU (WINDOWPROP WINDOW 'RIGHTMENU)
                                      WINDOW
                                      'RIGHT
                                      'TOP]
             (WINDOWPROP MenuWindow 'ATTACHEDTO WINDOW)
             ;; Don't pass CLOSEW to the main window !
             (WINDOWPROP MenuWindow 'PASSTOMAINCOMS '(MOVEW SHAPEW SHRINKW BURYW))
             [WINDOWPROP MenuWindow 'CLOSEFN (FUNCTION (LAMBDA (MW)
                                                         (WINDOWPROP (WINDOWPROP MW 'ATTACHEDTO)
                                                                     'FIXEDRIGHTMENU NIL)
                                                         (replace (MENU WHENSELECTEDFN)
                                                            of (WINDOWPROP (WINDOWPROP MW 'ATTACHEDTO)
                                                                           'RIGHTMENU)
                                                            with (FUNCTION DEFAULTWHENSELECTEDFN))
                                                         (DETACHWINDOW MW]
             (WINDOWPROP WINDOW 'FIXEDRIGHTMENU MenuWindow])

(FMC-DORIGHTSELECTION
  [LAMBDA (ITEM MENU.OR.WINDOW BUTTON)                      ; Edited 17-Aug-88 10:38 by A.BLAVIER
    ;; Handles right menu selection either pop-up or fixed.
    (LET ([WINDOW (if (WINDOWP MENU.OR.WINDOW)
                      then MENU.OR.WINDOW
                    else (GETMENUPROP MENU.OR.WINDOW 'FMC.WINDOW]
          (Selection (if (LISTP ITEM)
                         then (EVAL (CADR ITEM))
                       else ITEM)))
         (SELECTQ Selection
              (FIXRIGHTMENU (FMC-FIXRIGHTMENU WINDOW))
              (REDRAW (FMC-REDRAW WINDOW))
              (GRID (if (WINDOWPROP WINDOW 'GRIDSIZE)
                        then (FMC-PROMPTPRINT (CONCAT "Grid size is " (WINDOWPROP WINDOW 'GRIDSIZE))
                                              WINDOW)
                      else (FMC-PROMPTPRINT "No grid" WINDOW)))
              ((NOGRID 2 3 4 5 6 7 8 9 10)
                   (FMC-SET.GRIDSIZE Selection WINDOW))
              (DISPLAYGRID (FMC-DISPLAY.GRID WINDOW))
              (REMOVEGRIDDISPLAY
                   (if (WINDOWPROP WINDOW 'DISPLAYGRID)
                       then (WINDOWPROP WINDOW 'DISPLAYGRID NIL)
                            (FMC-REDRAW WINDOW)))
              (DELETE (FMC-DELETE WINDOW))
              (FORGET.SAVE.LIST
                   (WINDOWPROP WINDOW 'DELETED.ITEMS NIL))
              (UNDELETE.LAST
                   (FMC-UNDELETE 'LAST WINDOW))
              (UNDELETE.ALL (FMC-UNDELETE 'ALL WINDOW))
              (UNDELETE.LIST
                   (FMC-UNDELETE 'LIST WINDOW))
              (GROUP (FMC-GROUP WINDOW))
              (UNGROUP (FMC-UNGROUP WINDOW))
              ((ALIGNLEFT ALIGNRIGHT ALIGNTOP ALIGNBOTTOM)
```

```
                (FMC-ALIGN Selection WINDOW))
              (HCENTER (FMC-HCENTER WINDOW))
              (VCENTER (FMC-VCENTER WINDOW))
              (SELECTALL (FMC-SELECTALL WINDOW))
              (WBACKGROUND (FMC-CHOOSE.WINDOW.BG WINDOW))
              (SUMMARY (FMC-EDIT.INFO WINDOW))
              (IMPORT (FMC-IMPORT WINDOW))
              (COMPUTE (FMC-COMPUTE WINDOW))
              (GET (FMC-GET WINDOW))
              (PUT (FMC-PUT WINDOW))
             NIL])

)

          (* * Selection functions)

(DEFINEQ

(FMC-SELECT.ITEM
  [LAMBDA (WINDOW)                                          ; Edited 17-Aug-88 10:50 by A.BLAVIER
    ;; Highlight the unique selected item of WINDOW.
    (LET ((Item (WINDOWPROP WINDOW 'SELECTED.ITEM))
          Reg RLeft RBottom RRight RTop)
         (SETQ Reg (FMC-GET.ITEM Item REGION))
         (SETQ RLeft (IDIFFERENCE (GET.REGION.LEFT Reg)
                                  2))
         (SETQ RBottom (IDIFFERENCE (GET.REGION.BOTTOM Reg)
                                    2))
         (SETQ RRight (IPLUS (fetch (REGION RIGHT) of Reg)
                             2))
         (SETQ RTop (IPLUS (fetch (REGION TOP) of Reg)
                           2))
        ;;
         (DRAWLINE RLeft RBottom RRight RBottom 1 'REPLACE WINDOW NIL '(1 1))
         (DRAWLINE RRight RBottom RRight RTop 1 'REPLACE WINDOW NIL '(1 1))
         (DRAWLINE RRight RTop RLeft RTop 1 'REPLACE WINDOW NIL '(1 1))
         (DRAWLINE RLeft RTop RLeft RBottom 1 'REPLACE WINDOW NIL '(1 1))
         (FMC-COMPUTE.SHAPE.REGS WINDOW])

(FMC-SELECT.LIST
  [LAMBDA (WINDOW)                                          ; Edited 17-Aug-88 11:00 by A.BLAVIER
    ;; Highlight all the items of a multiple selection.
    (for item in (WINDOWPROP WINDOW 'SELECTION.LIST) do (FMC-SELECT.LIST.ITEM item WINDOW])

(FMC-SELECT.LIST.ITEM
  [LAMBDA (ITEM WINDOW)                                     ; Edited 17-Aug-88 10:56 by A.BLAVIER
    ;; Highlight ITEM, member of the selection list, as a part of a multiple selection.
    (LET* ((Reg (FMC-GET.ITEM ITEM REGION))
           RLeft RBottom RRight RTop)
          (SETQ RLeft (IDIFFERENCE (GET.REGION.LEFT Reg)
                                   2))
          (SETQ RBottom (IDIFFERENCE (GET.REGION.BOTTOM Reg)
                                     2))
          (SETQ RRight (IPLUS (fetch (REGION RIGHT) of Reg)
                              2))
          (SETQ RTop (IPLUS (fetch (REGION TOP) of Reg)
                            2))
         ;;
          (DRAWLINE RLeft RBottom RRight RBottom 1 'REPLACE WINDOW NIL '(1 3))
          (DRAWLINE RRight RBottom RRight RTop 1 'REPLACE WINDOW NIL '(1 3))
          (DRAWLINE RRight RTop RLeft RTop 1 'REPLACE WINDOW NIL '(1 3))
          (DRAWLINE RLeft RTop RLeft RBottom 1 'REPLACE WINDOW NIL '(1 3])

(FMC-SELECTALL
  [LAMBDA (WINDOW)                                          ; Edited 17-Aug-88 10:58 by A.BLAVIER
    ;; Select all the items of WINDOW.
    (if (WINDOWPROP WINDOW 'SELECTED.ITEM)
        then (FMC-DESELECT.ITEM WINDOW))
    (WINDOWPROP WINDOW 'SELECTION.LIST (WINDOWPROP WINDOW 'ITEMLIST))
    (FMC-SELECT.LIST WINDOW])

(FMC-GET.SELECTION
  [LAMBDA (WINDOW)                                          ; Edited 17-Aug-88 11:23 by A.BLAVIER
    ;; META + LEFT Button outside any item --> let the user select a region and add the enclosed items to the selection list.
    (LET ((SelectedItem (WINDOWPROP WINDOW 'SELECTED.ITEM))
```

```
          (SelectionRegion (GETREGION))
          (WLeft (IPLUS (GET.REGION.LEFT (WINDOWPROP WINDOW 'REGION))
                        3))
          (WBottom (IPLUS (GET.REGION.BOTTOM (WINDOWPROP WINDOW 'REGION))
                          3)))
        ;; convert SelectionRegion from Screen to Window coordinates
        (PUT.REGION.LEFT SelectionRegion (IDIFFERENCE (GET.REGION.LEFT SelectionRegion)
                                                      WLeft))
        (PUT.REGION.BOTTOM SelectionRegion (IDIFFERENCE (GET.REGION.BOTTOM SelectionRegion)
                                                        WBottom))
        (for item in (WINDOWPROP WINDOW 'ITEMLIST) do (if (SUBREGIONP SelectionRegion (FMC-GET.ITEM item REGION))
                                                          then (WINDOWADDPROP WINDOW 'SELECTION.LIST item)
                                                               (FMC-SELECT.LIST.ITEM item WINDOW)
                                                               (if SelectedItem
                                                                   then ;; if there was a unique selection then add it to the
                                                                        ;; multiple selection
                                                                        (WINDOWADDPROP WINDOW 'SELECTION.LIST
                                                                               SelectedItem T)
                                                                        (FMC-DESELECT.ITEM WINDOW)
                                                                        (FMC-SELECT.LIST.ITEM SelectedItem WINDOW)
                                                                        (SETQ SelectedItem NIL])
```

(FMC-DESELECT

```
  [LAMBDA (WINDOW)                                          ; Edited 17-Aug-88 11:24 by A.BLAVIER
    ;; Lowlight any selection.
    (if (WINDOWPROP WINDOW 'SELECTED.ITEM)
        then (FMC-DESELECT.ITEM WINDOW))
    (if (WINDOWPROP WINDOW 'SELECTION.LIST)
        then (FMC-DESELECT.LIST WINDOW])
```

(FMC-DESELECT.ITEM

```
  [LAMBDA (WINDOW)                                          ; Edited 17-Aug-88 11:25 by A.BLAVIER
    ;; Lowlight the unique selected item of WINDOW.
    (LET ((Item (WINDOWPROP WINDOW 'SELECTED.ITEM))
          Region Rleft Rbottom Rright Rtop)
         (SETQ Region (FMC-GET.ITEM Item REGION))
         (SETQ Rleft (IDIFFERENCE (GET.REGION.LEFT Region)
                                  2))
         (SETQ Rbottom (IDIFFERENCE (GET.REGION.BOTTOM Region)
                                    2))
         (SETQ Rright (IPLUS (fetch (REGION RIGHT) of Region)
                             2))
         (SETQ Rtop (IPLUS (fetch (REGION TOP) of Region)
                           2))
        ;;
         (DRAWLINE Rleft Rbottom Rright Rbottom 1 'ERASE WINDOW)
         (DRAWLINE Rright Rbottom Rright Rtop 1 'ERASE WINDOW)
         (DRAWLINE Rright Rtop Rleft Rtop 1 'ERASE WINDOW)
         (DRAWLINE Rleft Rtop Rleft Rbottom 1 'ERASE WINDOW)
         (WINDOWPROP WINDOW 'SELECTED.ITEM NIL])
```

(FMC-DESELECT.LIST

```
  [LAMBDA (WINDOW)                                          ; Edited 17-Aug-88 11:25 by A.BLAVIER
    ;; Lowlight all the items of a multiple selection.
    (for item in (WINDOWPROP WINDOW 'SELECTION.LIST) do (FMC-DESELECT.LIST.ITEM item WINDOW))
    (WINDOWPROP WINDOW 'SELECTION.LIST NIL])
```

(FMC-DESELECT.LIST.ITEM

```
  [LAMBDA (ITEM WINDOW)                                     ; Edited 17-Aug-88 11:29 by A.BLAVIER
    ;; Lowlight one item, member of a multiple selection.
    (LET ((Reg (fetch (FMC-ITEM REGION) of ITEM))
          RLeft RBottom RRight RTop)
         (SETQ RLeft (IDIFFERENCE (GET.REGION.LEFT Reg)
                                  2))
         (SETQ RBottom (IDIFFERENCE (GET.REGION.BOTTOM Reg)
                                    2))
         (SETQ RRight (IPLUS (fetch (REGION RIGHT) of Reg)
                             2))
         (SETQ RTop (IPLUS (fetch (REGION TOP) of Reg)
                           2))
        ;;
         (DRAWLINE RLeft RBottom RRight RBottom 1 'ERASE WINDOW)
         (DRAWLINE RRight RBottom RRight RTop 1 'ERASE WINDOW)
         (DRAWLINE RRight RTop RLeft RTop 1 'ERASE WINDOW)
         (DRAWLINE RLeft RTop RLeft RBottom 1 'ERASE WINDOW)
```

```
        (WINDOWDELPROP WINDOW 'SELECTION.LIST ITEM])

)


          (* * Property windows functions)

(DEFINEQ
```

(FMC-APPLY

```
  [LAMBDA (ITEM PROP.WINDOW BUTTON)                    ; Edited 17-Aug-88 11:51 by A.BLAVIER
    ;; User clicked in the APPLY item of the Item or Group Properties Free Menu : apply property(ies) to selected item(s).
    (LET ((MainWindow (WINDOWPROP PROP.WINDOW 'MAIN.WINDOW))
          (Prop T)
          Unique? ItemList)
         ;;
         (SETQ Unique? (WINDOWPROP MainWindow 'SELECTED.ITEM))
         [if Unique?
             then (WINDOWPROP MainWindow 'SELECTION.LIST (LIST (WINDOWPROP MainWindow 'SELECTED.ITEM]
         (SETQ ItemList (WINDOWPROP MainWindow 'SELECTION.LIST))
         ;;
         (if (STREQUAL (WINDOWPROP PROP.WINDOW 'TITLE)
                       "ITEM PROPERTIES")
             then ;; Request comes from the ITEM Properties window
                  ;; Check that no GROUP is in the selection
                  [if (for i in ItemList thereis (FMC-GROUP? i))
                      then (FMC-PROMPTPRINT "Can't apply to GROUPs" MainWindow)
                    else [if (IGREATERP (LENGTH ItemList)
                                        1)
                             then ;; if there is more than one selected item then apply only ONE property to each item
                                  (SETQ Prop
                                   (MENU (create MENU
                                                 ITEMS _
                                                 '(TYPE LABEL BOX BOXSHADE BACKGROUND FONT CHANGESTATE SELECTEDFN
                                                        DOWNFN HELDFN MOVEDFN)
                                                 TITLE _ "Apply which Property ?"]
                                (if Prop
                                    then (FMC-DESELECT.LIST MainWindow)
                                         (for item in ItemList do (DSPFILL (FMC-GET.ITEM item REGION)
                                                                           WHITESHADE
                                                                           'REPLACE MainWindow)
                                                                  (if Unique?
                                                                      then (FMC-UPDATE.ITEM item 'ALL PROP.WINDOW)
                                                                    else (FMC-UPDATE.ITEM item Prop PROP.WINDOW))
                                                                  (FMC-REDRAW.ITEM item MainWindow))
                                         (if Unique?
                                             then (WINDOWPROP MainWindow 'SELECTED.ITEM (CAR ItemList))
                                                  (WINDOWPROP MainWindow 'SELECTION.LIST NIL)
                                                  (FMC-SELECT.ITEM MainWindow)
                                           else (WINDOWPROP MainWindow 'SELECTION.LIST ItemList)
                                                (FMC-SELECT.LIST MainWindow]
           else ;; Request comes from the GROUP Properties window
                ;; Check that the selection is unique and that it's a GROUP
                (if (AND (EQP (LENGTH ItemList)
                              1)
                         (FMC-GROUP? (CAR ItemList)))
                    then (FMC-DESELECT.LIST MainWindow)
                         (DSPFILL (FMC-GET.ITEM (CAR ItemList)
                                                REGION)
                                  WHITESHADE
                                  'REPLACE MainWindow)
                         (FMC-UPDATE.GROUP (CAR ItemList)
                                           PROP.WINDOW)
                         (FMC-REDRAW.ITEM (CAR ItemList)
                                          MainWindow)
                         (WINDOWPROP MainWindow 'SELECTED.ITEM (CAR ItemList))
                         (WINDOWPROP MainWindow 'SELECTION.LIST NIL)
                         (FMC-SELECT.ITEM MainWindow)
                  else (if [AND (IGREATERP (LENGTH ItemList)
                                           0)
                                (NOT (FMC-GROUP? (CAR ItemList]
                           then (FMC-PROMPTPRINT "Can't apply to simple ITEMs" MainWindow])
```

(FMC-SHOW.ITEM

```
  [LAMBDA (IPW.ITEM IP.WINDOW BUTTON)                  ; Edited 17-Aug-88 11:47 by A.BLAVIER
    ;; User clicked in the SHOW item of the Item Properties Free Menu : update the Free Menu according to the first selected item.
    (LET ((MainWindow (WINDOWPROP IP.WINDOW 'MAIN.WINDOW))
          Item ItemBox ItemFont)
```

```
        [SETQ Item (OR (WINDOWPROP MainWindow 'SELECTED.ITEM)
                       (CAR (WINDOWPROP MainWindow 'SELECTION.LIST]
        (SETQ ItemBox (FMC-GET.ITEM Item BOX))
        (SETQ ItemFont (FMC-GET.ITEM Item FONT))
        (if Item
            then (if (FMC-GROUP? Item)
                     then (FMC-PROMPTPRINT "Can't show GROUPs" MainWindow)
                   else (FM.CHANGESTATE 'TYPE (FMC-GET.ITEM Item TYPE)
                               IP.WINDOW)
                        (FM.CHANGELABEL 'LABELLINK (FMC-GET.ITEM Item LABEL)
                               IP.WINDOW)
                        (FM.CHANGELABEL 'IDLINK (FMC-GET.ITEM Item ID)
                               IP.WINDOW)
                        (FM.CHANGESTATE 'BOX (if ItemBox
                                                 then ItemBox
                                               else 0)
                               IP.WINDOW)
                        (FM.CHANGESTATE 'BOXSHADE [CAR (CADADR (FMC-CREATE.SHADE.ITEM (FMC-GET.ITEM Item
                                                                                              BOXSHADE]
                               IP.WINDOW)
                        (WINDOWPROP IP.WINDOW 'FMC.ITEM.BOXSHADE (FMC-GET.ITEM Item BOXSHADE))
                        (FM.CHANGESTATE 'BACKGROUND [CAR (CADADR (FMC-CREATE.SHADE.ITEM (FMC-GET.ITEM Item
                                                                                                BACKGROUND]
                               IP.WINDOW)
                        (WINDOWPROP IP.WINDOW 'FMC.ITEM.BACKGROUND (FMC-GET.ITEM Item BACKGROUND))
                        (FM.CHANGESTATE 'FAMILY (CAR ItemFont)
                               IP.WINDOW)
                        (FM.CHANGESTATE 'SIZE (CADR ItemFont)
                               IP.WINDOW)
                        (FM.CHANGESTATE 'FACE (CADDR ItemFont)
                               IP.WINDOW)
                        (FM.CHANGELABEL 'MESSAGELINK (FMC-GET.ITEM Item MESSAGE)
                               IP.WINDOW)
                        (for prop in '(CHANGESTATE SELECTEDFN DOWNFN HELDFN MOVEDFN) bind Def
                           do (SETQ Def (SELECTQ prop
                                                 ((CHANGESTATE)
                                                      (FMC-GET.ITEM Item CHANGESTATE))
                                                 ((SELECTEDFN)
                                                      (FMC-GET.ITEM Item SELECTEDFN))
                                                 ((DOWNFN)
                                                      (FMC-GET.ITEM Item DOWNFN))
                                                 ((HELDFN)
                                                      (FMC-GET.ITEM Item HELDFN))
                                                 ((MOVEDFN)
                                                      (FMC-GET.ITEM Item MOVEDFN))
                                                T))
                              (FM.CHANGESTATE prop (if (ATOM Def)
                                                       then (MKATOM (LIST 'FUNCTION Def))
                                                     else (MKATOM Def))
                                     IP.WINDOW)
                              (WINDOWPROP IP.WINDOW (PACK* 'FMC.ITEM. prop)
                                     (if (ATOM Def)
                                         then (LIST 'FUNCTION Def)
                                       else Def)))
                        (FM.CHANGESTATE 'INFINITEWIDTH (FMC-GET.ITEM Item INFINITEWIDTH)
                               IP.WINDOW)
                        (FM.CHANGESTATE 'MENU (MKATOM (FMC-GET.ITEM Item MENU))
                               IP.WINDOW)
                        (WINDOWPROP IP.WINDOW 'FMC.ITEM.MENUPROPS (FMC-GET.ITEM Item MENU))
                        (FM.CHANGESTATE 'LINKS (MKATOM (FMC-GET.ITEM Item LINKS))
                               IP.WINDOW)
                        (WINDOWPROP IP.WINDOW 'FMC.ITEM.LINKS (FMC-GET.ITEM Item LINKS))
                        (FM.CHANGESTATE 'INITSTATE (FMC-GET.ITEM Item INITSTATE)
                               IP.WINDOW])
```

(FMC-SHOW.GROUP

```
  [LAMBDA (GPW.ITEM GP.WINDOW BUTTON)                    ; Edited 17-Aug-88 11:52 by A.BLAVIER
```

;; User clicked in the SHOW item of the Group Properties Free Menu : update the Free Menu according to the unique selected item.

```
    (LET ((MainWindow (WINDOWPROP GP.WINDOW 'MAIN.WINDOW))
          Item)
         (SETQ Item (WINDOWPROP MainWindow 'SELECTED.ITEM))
         (if Item
             then (if (FMC-GROUP? Item)
                      then (FM.CHANGELABEL 'IDLINK (FMC-GET.ITEM Item ID)
                                  GP.WINDOW)
                           (FM.CHANGELABEL 'COLLECTIONLINK (if (FMC-GET.ITEM Item COLLECTION)
                                                               then (FMC-GET.ITEM Item COLLECTION)
                                                             else "")
                                  GP.WINDOW)
                           (FM.CHANGESTATE 'DESELECT (FMC-GET.ITEM Item DESELECT)
                                  GP.WINDOW)
                           (FM.CHANGESTATE 'BOX (FMC-GET.ITEM Item BOX)
                                  GP.WINDOW)
                           (FM.CHANGESTATE 'BOXSHADE [CAR (CADADR (FMC-CREATE.SHADE.ITEM (FMC-GET.ITEM Item
                                                                                                 BOXSHADE]
```

```
                                GP.WINDOW)
                          (WINDOWPROP GP.WINDOW 'FMC.ITEM.BOXSHADE (FMC-GET.ITEM Item BOXSHADE))
                          (FM.CHANGESTATE 'BACKGROUND [CAR (CADADR (FMC-CREATE.SHADE.ITEM (FMC-GET.ITEM Item
                                                                                                BACKGROUND]
                                GP.WINDOW)
                          (WINDOWPROP GP.WINDOW 'FMC.ITEM.BACKGROUND (FMC-GET.ITEM Item BACKGROUND))
                    else (FMC-PROMPTPRINT "Can't show simple items" MainWindow])
```

(FMC-NEWITEM

```
[LAMBDA (IPW.ITEM IP.WINDOW BUTTON)                          ; Edited 17-Aug-88 12:00 by A.BLAVIER
```

;; User clicked in the NEW item of the Item Properties window : create a new FMC item.

```
(LET ((MainWindow (WINDOWPROP IP.WINDOW 'MAIN.WINDOW))
      (NewItem (create FMC-ITEM)))
```

;; set this new item's properties according to the IP window

```
(FMC-UPDATE.ITEM NewItem 'ALL IP.WINDOW)
```

;; if the cursor is currently outside the main window, move it inside

```
(if (NOT (INSIDEP (DSPCLIPPINGREGION NIL MainWindow)
                  (CURSORPOSITION NIL MainWindow)))
    then (CURSORPOSITION '(5 . 5)
                  MainWindow))
```

;; remove any previous selection

```
(if (WINDOWPROP MainWindow 'SELECTED.ITEM)
    then (FMC-DESELECT.ITEM MainWindow))
(if (WINDOWPROP MainWindow 'SELECTION.LIST)
    then (FMC-DESELECT.LIST MainWindow))
```

;; let the user place the item where he wants

```
(FMC-TRACK.NEW.ITEM NewItem MainWindow)
(WINDOWADDPROP MainWindow 'ITEMLIST NewItem)
```

;; make the new item the unique selection

```
(WINDOWPROP MainWindow 'SELECTED.ITEM NewItem)
(FMC-REDRAW.ITEM NewItem MainWindow)
(FMC-SELECT.ITEM MainWindow)
(FMC-COMPUTE.SHAPE.REGS MainWindow])
```

(FMC-UPDATE.ITEM

```
[LAMBDA (ITEM PROP IP.WINDOW)                                ; Edited 11-Aug-88 14:29 by A.BLAVIER
```

;; Update ITEM's fields according to the Item Properties Free Menu.

;; PROP is either ALL or one of LABEL, BOX, BOXSHADE, BACKGROUND, FONT, CHANGESTATE, SELECTEDFN, DOWNFN, HELDFN, MOVEDFN

```
(LET (ItemLabel ItemBox ItemBitmap)
     (FMC-MARK.AS.CHANGED (WINDOWPROP IP.WINDOW 'MAIN.WINDOW))
```

;; update LABEL

```
     (if (FMEMB PROP '(ALL LABEL))
         then (SETQ ItemLabel (FM-GET.ITEM.LABEL 'LABELLINK IP.WINDOW))
              (if (NULLSTR ItemLabel)
                  then (SETQ ItemLabel "*NOLABEL*"))
              (FMC-PUT.ITEM ITEM LABEL ItemLabel)
       else (SETQ ItemLabel (FMC-GET.ITEM ITEM LABEL)))
```

;; update BACKGROUND

```
     [if (FMEMB PROP '(ALL BACKGROUND))
         then (FMC-PUT.ITEM ITEM BACKGROUND (WINDOWPROP IP.WINDOW 'FMC.ITEM.BACKGROUND]
```

;; update BOX if required. Anyway keep the value of BOX : we'll need it later

```
     (if (FMEMB PROP '(ALL BOX))
         then (SETQ ItemBox (FM-GET.ITEM.STATE 'BOX IP.WINDOW))
              (if (EQP ItemBox 0)
                  then (SETQ ItemBox NIL))
              (FMC-PUT.ITEM ITEM BOX ItemBox)
       else (SETQ ItemBox (FMC-GET.ITEM ITEM BOX)))
```

;; update BOXSHADE

```
     (if (FMEMB PROP '(ALL BOXSHADE))
         then (FMC-PUT.ITEM ITEM BOXSHADE (if ItemBox
                                              then (WINDOWPROP IP.WINDOW 'FMC.ITEM.BOXSHADE)
                                            else NIL)))
```

;; update TYPE

```
     (if (FMEMB PROP '(ALL TYPE))
         then (FMC-PUT.ITEM ITEM TYPE (FM-GET.ITEM.STATE 'TYPE IP.WINDOW)))
```

;; update FONT

```
     [if (FMEMB PROP '(ALL FONT))
         then (FMC-PUT.ITEM ITEM FONT (LIST (FM-GET.ITEM.STATE 'FAMILY IP.WINDOW)
                                            (FM-GET.ITEM.STATE 'SIZE IP.WINDOW)
                                            (FM-GET.ITEM.STATE 'FACE IP.WINDOW]
```

```
;; update CHANGESTATE, SELECTEDFN, DOWNFN, HELDFN, MOVEDFN
(SELECTQ PROP
     ((ALL)
          (FMC-PUT.ITEM ITEM CHANGESTATE (WINDOWPROP IP.WINDOW 'FMC.ITEM.CHANGESTATE))
          (FMC-PUT.ITEM ITEM SELECTEDFN (WINDOWPROP IP.WINDOW 'FMC.ITEM.SELECTEDFN))
          (FMC-PUT.ITEM ITEM DOWNFN (WINDOWPROP IP.WINDOW 'FMC.ITEM.DOWNFN))
          (FMC-PUT.ITEM ITEM HELDFN (WINDOWPROP IP.WINDOW 'FMC.ITEM.HELDFN))
          (FMC-PUT.ITEM ITEM MOVEDFN (WINDOWPROP IP.WINDOW 'FMC.ITEM.MOVEDFN)))
     ((CHANGESTATE)
          (FMC-PUT.ITEM ITEM CHANGESTATE (WINDOWPROP IP.WINDOW 'FMC.ITEM.CHANGESTATE)))
     ((SELECTEDFN)
          (FMC-PUT.ITEM ITEM SELECTEDFN (WINDOWPROP IP.WINDOW 'FMC.ITEM.SELECTEDFN)))
     ((DOWNFN)
          (FMC-PUT.ITEM ITEM DOWNFN (WINDOWPROP IP.WINDOW 'FMC.ITEM.DOWNFN)))
     ((HELDFN)
          (FMC-PUT.ITEM ITEM HELDFN (WINDOWPROP IP.WINDOW 'FMC.ITEM.HELDFN)))
     ((MOVEDFN)
          (FMC-PUT.ITEM ITEM MOVEDFN (WINDOWPROP IP.WINDOW 'FMC.ITEM.MOVEDFN)))
     T)
;; update ID, MESSAGE, MENU, LINKS, INITSTATE, INFINITEWIDTH
(if (EQ PROP 'ALL)
    then [FMC-PUT.ITEM ITEM ID (if (NULLSTR (FM-GET.ITEM.LABEL 'IDLINK IP.WINDOW))
                                   then NIL
                               else (MKATOM (FM-GET.ITEM.LABEL 'IDLINK IP.WINDOW]
         (FMC-PUT.ITEM ITEM MESSAGE (FM-GET.ITEM.LABEL 'MESSAGELINK IP.WINDOW))
         (FMC-PUT.ITEM ITEM MENU (WINDOWPROP IP.WINDOW 'FMC.ITEM.MENUPROPS))
         (FMC-PUT.ITEM ITEM LINKS (WINDOWPROP IP.WINDOW 'FMC.ITEM.LINKS))
         (FMC-PUT.ITEM ITEM INITSTATE (FM-GET.ITEM.STATE 'INITSTATE IP.WINDOW))
         (FMC-PUT.ITEM ITEM INFINITEWIDTH (FM-GET.ITEM.STATE 'INFINITEWIDTH IP.WINDOW)))
;; recompute the bitmap
[SETQ ItemBitmap (FMC-MAKEBITMAP ItemLabel (FMC-LIST->FONT (FMC-GET.ITEM ITEM FONT]
(FMC-PUT.ITEM ITEM BITMAP ItemBitmap)
;; if ITEM has no region yet, then it's a new item --> create the region
[if (NOT (FMC-GET.ITEM ITEM REGION))
    then (FMC-PUT.ITEM ITEM REGION (CREATEREGION 5 5 (BITMAPWIDTH ItemBitmap)
                                                     (BITMAPHEIGHT ItemBitmap]
;; update XBMOFFSET and YBMOFFSET
(if (NOT ItemBox)
    then (FMC-PUT.ITEM ITEM XBMOFFSET 0)
         (FMC-PUT.ITEM ITEM YBMOFFSET 0))
(if (AND ItemBox (EQP (FMC-GET.ITEM ITEM YBMOFFSET)
                      0))
    then (FMC-PUT.ITEM ITEM XBMOFFSET 2)
         (FMC-PUT.ITEM ITEM YBMOFFSET 2))
;; update REGION according to BITMAP and BOXSPACE
(FMC-PUT.ITEM ITEM REGION (CREATEREGION (GET.REGION.LEFT (FMC-GET.ITEM ITEM REGION))
                                        (GET.REGION.BOTTOM (FMC-GET.ITEM ITEM REGION))
                                        (IPLUS (BITMAPWIDTH ItemBitmap)
                                               (FMC-GET.ITEM ITEM XBMOFFSET)
                                               (FMC-GET.ITEM ITEM XBMOFFSET))
                                        (IPLUS (BITMAPHEIGHT ItemBitmap)
                                               (FMC-GET.ITEM ITEM YBMOFFSET)
                                               (FMC-GET.ITEM ITEM YBMOFFSET])
```

(FMC-UPDATE.GROUP

```
[LAMBDA (GROUP GP.WINDOW)                                   ; Edited 11-Aug-88 16:45 by A.BLAVIER
;; Update GROUP's fields according to the Group Properties Free Menu.
(FMC-MARK.AS.CHANGED (WINDOWPROP GP.WINDOW 'MAIN.WINDOW))
;; update ID
[FMC-PUT.ITEM GROUP ID (if (NULLSTR (FM-GET.ITEM.LABEL 'IDLINK GP.WINDOW))
                           then NIL
                       else (MKATOM (FM-GET.ITEM.LABEL 'IDLINK GP.WINDOW]
;; update COLLECTION. Check that all the subitems are NWAY.
[if (NULLSTR (FM-GET.ITEM.LABEL 'COLLECTIONLINK GP.WINDOW))
    then (FMC-PUT.ITEM GROUP COLLECTION NIL)
  else (if (for i in (FMC-GET.ITEM GROUP OBJECTLIST) thereis (NEQ (FMC-GET.ITEM i TYPE)
                                                                  'NWAY))
           then (FMC-PROMPTPRINT "All items in GROUP should be NWAY" (WINDOWPROP GP.WINDOW 'MAIN.WINDOW))
       else (FMC-PUT.ITEM GROUP COLLECTION (MKATOM (FM-GET.ITEM.LABEL 'COLLECTIONLINK GP.WINDOW]
;; update DESELECT
(FMC-PUT.ITEM GROUP DESELECT (FM-GET.ITEM.STATE 'DESELECT GP.WINDOW))
;; update BOX
(FMC-PUT.ITEM GROUP BOX (FM-GET.ITEM.STATE 'BOX GP.WINDOW))
;; update BOXSHADE
(FMC-PUT.ITEM GROUP BOXSHADE (WINDOWPROP GP.WINDOW 'FMC.ITEM.BOXSHADE))
```

```
    ;; update BACKGROUND
    (FMC-PUT.ITEM GROUP BACKGROUND (WINDOWPROP GP.WINDOW 'FMC.ITEM.BACKGROUND])
```

(FMC-GET.LABEL

```
  [LAMBDA (IPW.ITEM IP.WINDOW BUTTON)                    ; Edited 17-Aug-88 12:08 by A.BLAVIER
    ;; User clicked in the LABEL item of the IP window.
    (COND
       ((EQUAL BUTTON '(LEFT))
        ;; he LEFT buttoned : edit the field
        (FM.EDITITEM 'LABELLINK IP.WINDOW))
       ((EQUAL BUTTON '(MIDDLE))
        ;; he MIDDLE buttoned : grab a bitmap from screen and set LABEL to it
        (FM.CHANGELABEL 'LABELLINK (FMC-SNAPBM)
               IP.WINDOW T))
       ((EQUAL BUTTON '(RIGHT))
        ;; he RIGHT buttoned : edit the field, clearing it first
        (FM.EDITITEM 'LABELLINK IP.WINDOW T])
```

(FMC-CHOOSE.ITEM.BOXSHADE

```
  [LAMBDA (IPW.ITEM IP.WINDOW BUTTON)                    ; Edited 17-Aug-88 12:10 by A.BLAVIER
    ;; User clicked in the BOXSHADE item of the Item or Group Prop window : let him choose a shade from a menu.
    (LET ((ShadeMenu (WINDOWPROP IP.WINDOW 'SHADE.MENU))
          Shade)
         (SETQ Shade (MENU ShadeMenu))
         (if Shade
             then (FM.CHANGESTATE 'BOXSHADE (CAR Shade)
                                  IP.WINDOW)
                  (WINDOWPROP IP.WINDOW 'FMC.ITEM.BOXSHADE (CDR Shade])
```

(FMC-CHOOSE.ITEM.BG

```
  [LAMBDA (IPW.ITEM IP.WINDOW BUTTON)                    ; Edited 17-Aug-88 12:12 by A.BLAVIER
    ;; User clicked in the BACKGROUND item of the Item or Group Prop window : let him choose a background from a menu.
    (LET ((ShadeMenu (WINDOWPROP IP.WINDOW 'SHADE.MENU))
          Shade)
         (SETQ Shade (MENU ShadeMenu))
         (if Shade
             then (FM.CHANGESTATE 'BACKGROUND (CAR Shade)
                                  IP.WINDOW)
                  (WINDOWPROP IP.WINDOW 'FMC.ITEM.BACKGROUND (CDR Shade])
```

(FMC-GET.MENUPROPS

```
  [LAMBDA (IPW.ITEM IP.WINDOW BUTTON)                    ; Edited 17-Aug-88 12:15 by A.BLAVIER
    ;; User clicked in the MENU button of the IP window : edit an EXPR where CAR is a list of menu items, CADR a font description list, CADDR a title
    ;; for the menu.
    (LET (Expr)
         [if (CAR (WINDOWPROP IP.WINDOW 'FMC.ITEM.MENUPROPS))
             then [SETQ Expr (EDITE (COPYALL (WINDOWPROP IP.WINDOW 'FMC.ITEM.MENUPROPS]
           else (SETQ Expr (EDITE (COPYALL '(("ITEMS")
                                             "[FONT]" "[TITLE]"]
         ;; a FM label can't be a list, so make it an atom
         (FM.CHANGESTATE 'MENU (MKATOM Expr)
                IP.WINDOW)
         ;; but keep the list in mind
         (WINDOWPROP IP.WINDOW 'FMC.ITEM.MENUPROPS Expr])
```

(FMC-GET.INITSTATE

```
  [LAMBDA (IPW.ITEM IP.WINDOW BUTTON)                    ; Edited 17-Aug-88 12:28 by A.BLAVIER
    ;; User clicked in the INITSTATE button of the IP window.
    (LET ((MainWindow (WINDOWPROP IP.WINDOW 'MAIN.WINDOW))
          (ItemType (FM-GET.ITEM.STATE 'TYPE IP.WINDOW))
          (MenuProps (WINDOWPROP IP.WINDOW 'FMC.ITEM.MENUPROPS))
          InitstateMenuItems Choice PromptWindow)
         ;; The INITSTATE property is relevant only for TOGGLE, 3STATE and STATE items.
         (if (FMEMB ItemType '(TOGGLE 3STATE STATE))
             then ;; pop up a menu with the possible values for the INITSTATE, depending on the item type
                  (SETQ InitstateMenuItems (SELECTQ ItemType
                                                    (TOGGLE '("NIL" T))
                                                    (3STATE '(OFF "NIL" T))
                                                    (STATE (if (CAR MenuProps)
```

```
                                                    then (APPEND '("NIL" T)
                                                                 (CAR MenuProps)
                                                                 '(OTHER))
                                                  else '("NIL" T OTHER)))
                                           T))
                    (SETQ Choice (MENU (create MENU
                                               ITEMS _ InitstateMenuItems)))
                    ;; nothing is appropriate : let him specify a value
                    [if (EQ Choice 'OTHER)
                        then (SETQ PromptWindow (WINDOWPROP MainWindow 'FMC.PROMPTWINDOW))
                             (DSPRESET PromptWindow)
                             (TTY.PROCESS (THIS.PROCESS))
                             (SETQ Choice (PROMPTFORWORD "INITSTATE :" NIL "Type in an unevaluated expression"
                                                 PromptWindow NIL NIL (CHARCODE (EOL ESCAPE LF TAB]
                  else (SETQ Choice NIL)
                       (FM.CHANGESTATE 'INITSTATE Choice IP.WINDOW))
                (if Choice
                    then (if (STREQUAL Choice "NIL")
                             then (SETQ Choice NIL))
                         (FM.CHANGESTATE 'INITSTATE Choice IP.WINDOW])
```

(FMC-EDIT.FN

```
  [LAMBDA (IPW.ITEM IP.WINDOW BUTTON)                    ; Edited 17-Aug-88 13:36 by A.BLAVIER
    ;; User wants to specify a function property : open a SEdit window.
    ;; Expr should be of the form (FUNCTION FNname) or a LAMDA expression
    (LET ((Prop (FM.ITEMPROP IPW.ITEM 'ID))
          Expr)
         [SETQ Expr (EDITE (COPYALL (WINDOWPROP IP.WINDOW (PACK* 'FMC.ITEM. Prop]
         (FM.CHANGESTATE Prop (MKATOM Expr)
                IP.WINDOW)
         (WINDOWPROP IP.WINDOW (PACK* 'FMC.ITEM. Prop)
                Expr])
```

(FMC-LINKS

```
  [LAMBDA (IPW.ITEM IP.WINDOW BUTTON)                    ; Edited 17-Aug-88 13:44 by A.BLAVIER
    ;; User clicked in the LINKS button of the IP window : let him add or remove a link.
    (LET ((MainWindow (WINDOWPROP IP.WINDOW 'MAIN.WINDOW))
          [LinkMenu (create MENU
                            ITEMS _ '(("Add Link" 'ADD)
                                      ("Remove Link" 'REMOVE]
          Item Id Link)
         (SELECTQ (MENU LinkMenu)
              (ADD (FMC-PROMPTPRINT "Click on the item you want to link to" MainWindow)
                   (until (MOUSESTATE LEFT))
                   (if [SETQ Item (for i in (WINDOWPROP MainWindow 'ITEMLIST) thereis (INSIDEP (FMC-GET.ITEM i
                                                                                                    REGION)
                                                                                              (CURSORPOSITION NIL
                                                                                                     MainWindow]
                       then (SETQ Id (FMC-GET.ITEM Item ID))
                            (if Id
                                then (SETQ Link (LIST (FMC-GET.ITEM Item TYPE)
                                                      (MKATOM Id)))
                                     (WINDOWPROP IP.WINDOW 'FMC.ITEM.LINKS Link)
                                     (FM.CHANGESTATE 'LINKS (MKATOM Link)
                                            IP.WINDOW)
                              else (FMC-PROMPTPRINT "This item has no ID - can't link" MainWindow)))
                   (while (MOUSESTATE LEFT)))
              (REMOVE (WINDOWPROP IP.WINDOW 'FMC.ITEM.LINKS '(NIL))
                      (FM.CHANGESTATE 'LINKS "(NIL)" IP.WINDOW))
              T])

)


          (* * Property windows descriptions)

(RPAQQ FMC-IP-DESC
       ((PROPS FORMAT EXPLICIT)
        (LABEL APPLY TYPE MOMENTARY LEFT 0 BOTTOM 195 BOX 1 BOXSHADE 65535 BOXSPACE 1 FONT (MODERN 14 BOLDITALIC
                                                                                                    )
               SELECTEDFN FMC-APPLY)
        (LABEL SHOW TYPE MOMENTARY LEFT 49 BOTTOM 195 BOX 1 BOXSHADE 65535 BOXSPACE 1 FONT (MODERN 14 BOLDITALIC
                                                                                                    )
               SELECTEDFN FMC-SHOW.ITEM)
        (LABEL NEW TYPE MOMENTARY LEFT 111 BOTTOM 195 BOX 1 BOXSHADE 65535 BOXSPACE 1 FONT (MODERN 14 BOLDITALIC
                                                                                                    )
               SELECTEDFN FMC-NEWITEM)
        (LABEL TYPE TYPE STATE LEFT -1 BOTTOM 178 FONT (MODERN 12 BOLD)
               MENUITEMS
               (MOMENTARY TOGGLE 3STATE STATE NWAY EDIT NUMBER EDITSTART DISPLAY)
               LINKS
               (DISPLAY TYPELINK)
```

```
        INITSTATE MOMENTARY)
    (LABEL MOMENTARY TYPE DISPLAY LEFT 31 BOTTOM 177 ID TYPELINK FONT (MODERN 12 STANDARD))
    (LABEL LABEL TYPE MOMENTARY LEFT 113 BOTTOM 178 FONT (MODERN 12 BOLD)
        SELECTEDFN FMC-GET.LABEL LINKS (EDIT LABELLINK))
    (LABEL "" TYPE EDIT LEFT 151 BOTTOM 177 ID LABELLINK FONT (MODERN 12 STANDARD)
        INITSTATE "IIIIMMMMMMMMMMMMMM")
    (LABEL ID TYPE EDITSTART LEFT -1 BOTTOM 162 FONT (MODERN 12 BOLD)
        LINKS
        (EDIT IDLINK))
    (LABEL "" TYPE EDIT LEFT 14 BOTTOM 161 ID IDLINK FONT (MODERN 12 STANDARD)
        INITSTATE "")
    (LABEL FONT TYPE DISPLAY LEFT -1 BOTTOM 146 FONT (MODERN 12 ITALIC))
    (LABEL FAMILY TYPE STATE LEFT 32 BOTTOM 146 ID FAMILY FONT (MODERN 12 BOLD)
        MENUITEMS
        (CLASSIC MODERN TERMINAL TITAN GACHA HELVETICA TIMESROMAN)
        LINKS
        (DISPLAY FAMILYLINK)
        INITSTATE GACHA)
    (LABEL GACHA TYPE DISPLAY LEFT 77 BOTTOM 145 ID FAMILYLINK FONT (MODERN 12 STANDARD))
    (LABEL SIZE TYPE STATE LEFT 161 BOTTOM 146 ID SIZE FONT (MODERN 12 BOLD)
        MENUITEMS
        (6 7 8 9 10 11 12 14 18 24 30 36)
        LINKS
        (DISPLAY SIZELINK)
        INITSTATE 12)
    (LABEL 10 TYPE DISPLAY LEFT 191 BOTTOM 145 ID SIZELINK FONT (MODERN 12 STANDARD))
    (LABEL FACE TYPE STATE LEFT 210 BOTTOM 146 ID FACE FONT (MODERN 12 BOLD)
        MENUITEMS
        (REGULAR ITALIC BOLD BOLDITALIC)
        LINKS
        (DISPLAY FACELINK)
        INITSTATE BOLDITALIC)
    (LABEL REGULAR TYPE DISPLAY LEFT 241 BOTTOM 145 ID FACELINK FONT (MODERN 12 STANDARD))
    (LABEL BOX TYPE STATE LEFT -1 BOTTOM 130 FONT (MODERN 12 BOLD)
        MENUITEMS
        (0 1 2 3 4 5 6 7 8 9 10)
        LINKS
        (DISPLAY BOXLINK)
        INITSTATE 0)
    (LABEL 0 TYPE DISPLAY LEFT 26 BOTTOM 129 ID BOXLINK FONT (MODERN 12 STANDARD))
    (LABEL BOXSHADE TYPE STATE LEFT 40 BOTTOM 130 FONT (MODERN 12 BOLD)
        SELECTEDFN FMC-CHOOSE.ITEM.BOXSHADE LINKS (DISPLAY BOXSHADELINK))
    (LABEL "" TYPE DISPLAY LEFT 107 BOTTOM 130 ID BOXSHADELINK FONT (MODERN 12 STANDARD)
        MAXWIDTH 60)
    (LABEL BACKGROUND TYPE STATE LEFT 176 BOTTOM 130 ID BACKGROUND FONT (MODERN 12 BOLD)
        SELECTEDFN FMC-CHOOSE.ITEM.BG LINKS (DISPLAY BACKGROUNDLINK))
    (LABEL "" TYPE DISPLAY LEFT 262 BOTTOM 130 ID BACKGROUNDLINK FONT (MODERN 12 STANDARD)
        MAXWIDTH 60)
    (LABEL MENU TYPE STATE LEFT -1 BOTTOM 114 FONT (MODERN 12 BOLD)
        SELECTEDFN FMC-GET.MENUPROPS LINKS (DISPLAY MENULINK)
        INITSTATE "(NIL)")
    (LABEL "(NIL)" TYPE DISPLAY LEFT 39 BOTTOM 113 ID MENULINK FONT (MODERN 12 STANDARD))
    (LABEL INITSTATE TYPE STATE LEFT 195 BOTTOM 114 ID INITSTATE FONT (MODERN 12 BOLD)
        SELECTEDFN FMC-GET.INITSTATE LINKS (DISPLAY INITSTATELINK))
    (LABEL "#NOLABEL#" TYPE DISPLAY LEFT 257 BOTTOM 113 ID INITSTATELINK FONT (MODERN 12 STANDARD))
    (LABEL CHANGESTATE TYPE STATE LEFT -1 BOTTOM 98 ID CHANGESTATE FONT (MODERN 12 BOLD)
        SELECTEDFN FMC-EDIT.FN LINKS (DISPLAY CHANGESTATELINK)
        INITSTATE "(FUNCTION NILL)")
    (LABEL "(FUNCTION NILL)" TYPE DISPLAY LEFT 89 BOTTOM 97 ID CHANGESTATELINK FONT (MODERN 12 STANDARD))
    (LABEL SELECTEDFN TYPE STATE LEFT -1 BOTTOM 81 ID SELECTEDFN FONT (MODERN 12 BOLD)
        SELECTEDFN FMC-EDIT.FN LINKS (DISPLAY SELECTEDFNLINK)
        INITSTATE "(FUNCTION NILL)")
    (LABEL "(FUNCTION NILL)" TYPE DISPLAY LEFT 75 BOTTOM 80 ID SELECTEDFNLINK FONT (MODERN 12 STANDARD))
    (LABEL "DOWNFN" TYPE STATE LEFT -1 BOTTOM 65 ID DOWNFN FONT (MODERN 12 BOLD)
        SELECTEDFN FMC-EDIT.FN LINKS (DISPLAY DOWNFNLINK)
        INITSTATE "(FUNCTION NILL)")
    (LABEL "(FUNCTION NILL)" TYPE DISPLAY LEFT 57 BOTTOM 64 ID DOWNFNLINK FONT (MODERN 12 STANDARD))
    (LABEL "HELDFN" TYPE STATE LEFT -1 BOTTOM 49 ID HELDFN FONT (MODERN 12 BOLD)
        SELECTEDFN FMC-EDIT.FN LINKS (DISPLAY HELDFNLINK)
        INITSTATE "(FUNCTION NILL)")
    (LABEL "(FUNCTION NILL)" TYPE DISPLAY LEFT 49 BOTTOM 48 ID HELDFNLINK FONT (MODERN 12 STANDARD))
    (LABEL "MOVEDFN" TYPE STATE LEFT -1 BOTTOM 33 ID MOVEDFN FONT (MODERN 12 BOLD)
        SELECTEDFN FMC-EDIT.FN LINKS (DISPLAY MOVEDFNLINK)
        INITSTATE "(FUNCTION NILL)")
    (LABEL "(FUNCTION NILL)" TYPE DISPLAY LEFT 62 BOTTOM 32 ID MOVEDFNLINK FONT (MODERN 12 STANDARD))
    (LABEL LINKS TYPE STATE LEFT -1 BOTTOM 16 FONT (MODERN 12 BOLD)
        SELECTEDFN FMC-LINKS LINKS (DISPLAY LINKSLINK)
        INITSTATE "(NIL)")
    (LABEL "(NIL)" TYPE DISPLAY LEFT 36 BOTTOM 15 ID LINKSLINK FONT (MODERN 12 STANDARD))
    (LABEL "INFINITEWIDTH" TYPE TOGGLE LEFT 239 BOTTOM 16 ID INFINITEWIDTH FONT (MODERN 12 BOLD))
    (LABEL MESSAGE TYPE EDITSTART LEFT -1 BOTTOM 0 FONT (MODERN 12 BOLD)
        LINKS
        (EDIT MESSAGELINK))
    (LABEL "" TYPE EDIT LEFT 61 BOTTOM -1 ID MESSAGELINK FONT (MODERN 12 STANDARD)
        INITSTATE "MMMMMMMMMMMMMMMMMMMMMMMM")))

(RPAQQ FMC-GP-DESC
```

```
       ((PROPS FORMAT EXPLICIT)
        (LABEL APPLY TYPE MOMENTARY LEFT 0 BOTTOM 106 BOX 1 BOXSHADE 65535 BOXSPACE 1 FONT (MODERN 14 BOLDITALIC
                                                                                                          )
               SELECTEDFN FMC-APPLY)
        (LABEL SHOW TYPE MOMENTARY LEFT 49 BOTTOM 106 BOX 1 BOXSHADE 65535 BOXSPACE 1 FONT (MODERN 14 BOLDITALIC
                                                                                                        )
               SELECTEDFN FMC-SHOW.GROUP)
        (LABEL "ID" TYPE EDITSTART LEFT 0 BOTTOM 83 FONT (MODERN 12 BOLD)
               LINKS
               (EDIT IDLINK))
        (LABEL "" TYPE EDIT LEFT 17 BOTTOM 82 ID IDLINK FONT (MODERN 12 STANDARD))
        (LABEL "COLLECTION" TYPE EDITSTART LEFT 0 BOTTOM 67 FONT (MODERN 12 BOLD)
               LINKS
               (EDIT COLLECTIONLINK))
        (LABEL "" TYPE EDIT LEFT 75 BOTTOM 66 ID COLLECTIONLINK FONT (MODERN 12 STANDARD))
        (LABEL "DESELECT" ID DESELECT TYPE TOGGLE LEFT 0 BOTTOM 50 FONT (MODERN 12 BOLD))
        (LABEL BOX TYPE STATE LEFT 0 BOTTOM 33 FONT (MODERN 12 BOLD)
               MENUITEMS
               (1 2 3 4 5 6 7 8 9 10)
               LINKS
               (DISPLAY BOXLINK)
               INITSTATE 1)
        (LABEL 1 TYPE DISPLAY LEFT 27 BOTTOM 32 ID BOXLINK FONT (MODERN 12 STANDARD))
        (LABEL BOXSHADE TYPE STATE LEFT 0 BOTTOM 16 FONT (MODERN 12 BOLD)
               SELECTEDFN FMC-CHOOSE.ITEM.BOXSHADE LINKS (DISPLAY BOXSHADELINK))
        (LABEL "" TYPE DISPLAY LEFT 86 BOTTOM 16 ID BOXSHADELINK FONT (MODERN 12 STANDARD)
               MAXWIDTH 60)
        (LABEL BACKGROUND TYPE STATE LEFT 0 BOTTOM 0 ID BACKGROUND FONT (MODERN 12 BOLD)
               SELECTEDFN FMC-CHOOSE.ITEM.BG LINKS (DISPLAY BACKGROUNDLINK))
        (LABEL "" TYPE DISPLAY LEFT 86 BOTTOM 0 ID BACKGROUNDLINK FONT (MODERN 12 STANDARD)
               MAXWIDTH 60)))
```

(* * Creating bitmaps)

```
(DEFINEQ
```

(FMC-MAKEBITMAP

```
  [LAMBDA (LABEL FONT)                                    ; Edited 17-Aug-88 13:50 by A.BLAVIER
```

;; Create the bitmap representing LABEL in font FONT.

```
    (PROG (Bitmap Stream)
          (if (BITMAPP LABEL)
              then (RETURN LABEL)
            else [SETQ Bitmap (BITMAPCREATE (STRINGWIDTH LABEL FONT)
                                            (FONTPROP FONT 'HEIGHT]
                 (SETQ Stream (DSPCREATE Bitmap))
                 (DSPFONT FONT Stream)
                 (MOVETO 0 (FONTPROP FONT 'DESCENT)
                         Stream)
                 (PRIN1 LABEL Stream)
                 (RETURN Bitmap])
```

(FMC-COMPOUND.BITMAP

```
  [LAMBDA (COMPOUND.REGION ITEM.LIST)                     ; Edited 17-Aug-88 13:48 by A.BLAVIER
```

;; Create a bitmap composed of the bitmaps from ITEM.LIST

```
    (PROG ((CompoundBitmap (BITMAPCREATE (GET.REGION.WIDTH COMPOUND.REGION)
                                         (GET.REGION.HEIGHT COMPOUND.REGION)))
           Stream)
          (SETQ Stream (DSPCREATE CompoundBitmap))
          [for item in ITEM.LIST do (BITBLT (FMC-GET.ITEM item BITMAP)
                                            0 0 Stream (IPLUS (FMC-GET.ITEM item XBMOFFSET)
                                                              (IDIFFERENCE (GET.REGION.LEFT (FMC-GET.ITEM item
                                                                                                          REGION))
                                                                           (GET.REGION.LEFT COMPOUND.REGION)))
                                            (IPLUS (FMC-GET.ITEM item YBMOFFSET)
                                                   (IDIFFERENCE (GET.REGION.BOTTOM (FMC-GET.ITEM item REGION))
                                                                (GET.REGION.BOTTOM COMPOUND.REGION]
          (RETURN CompoundBitmap])
```

(FMC-SNAPBM

```
  [LAMBDA NIL                                             ; Edited 17-Aug-88 13:51 by A.BLAVIER
```

;; Extract a bitmap from the SCREENBITMAP

```
    (PROG ((Region (GETREGION))
           SnapBM)
          (SETQ SnapBM (BITMAPCREATE (GET.REGION.WIDTH Region)
                                     (GET.REGION.HEIGHT Region)))
          (BITBLT (SCREENBITMAP)
                  (GET.REGION.LEFT Region)
                  (GET.REGION.BOTTOM Region)
                  SnapBM)
          (RETURN SnapBM])
```

```
)

          (* * Moving items)

(DEFINEQ

(FMC-MOVE.SELECTION
  [LAMBDA (WINDOW UNIQUE.OR.MULTIPLE)                    ; Edited 17-Aug-88 14:38 by A.BLAVIER
    ;; Move the selection, either unique item or selection list.
    (LET (ItemList CompoundRegion CompoundBitmap Stream NewPos CorrX CorrY Left Bottom Width Height Right Top)
         (FMC-MARK.AS.CHANGED WINDOW)
         [if (EQ UNIQUE.OR.MULTIPLE 'UNIQUE)
             then (WINDOWPROP WINDOW 'SELECTION.LIST (LIST (WINDOWPROP WINDOW 'SELECTED.ITEM]
         (SETQ ItemList (WINDOWPROP WINDOW 'SELECTION.LIST))
         (FMC-DESELECT.LIST WINDOW)
         ;; compute the enclosing region
         [SETQ CompoundRegion (APPLY 'UNIONREGIONS (for item in ItemList collect (FMC-GET.ITEM item REGION]
         ;; compute the associated bitmap
         (SETQ CompoundBitmap (FMC-COMPOUND.BITMAP CompoundRegion ItemList))
         (SETQ Stream (DSPCREATE CompoundBitmap))
         ;; surround the moving region
         (SETQ Left (GET.REGION.LEFT CompoundRegion))
         (SETQ Bottom (GET.REGION.BOTTOM CompoundRegion))
         (SETQ Width (IDIFFERENCE (GET.REGION.WIDTH CompoundRegion)
                                  1))
         (SETQ Height (IDIFFERENCE (GET.REGION.HEIGHT CompoundRegion)
                                   1))
         (SETQ Right (fetch (REGION RIGHT) of CompoundRegion))
         (SETQ Top (fetch (REGION TOP) of CompoundRegion))
         (DRAWLINE Left Bottom Right Bottom 1 'REPLACE WINDOW NIL '(2 2))
         (DRAWLINE Right Bottom Right Top 1 'REPLACE WINDOW NIL '(2 2))
         (DRAWLINE Right Top Left Top 1 'REPLACE WINDOW NIL '(2 2))
         (DRAWLINE Left Top Left Bottom 1 'REPLACE WINDOW NIL '(2 2))
         (DRAWLINE 0 0 Width 0 1 'REPLACE Stream NIL '(2 2))
         (DRAWLINE Width 0 Width Height 1 'REPLACE Stream NIL '(2 2))
         (DRAWLINE Width Height 0 Height 1 'REPLACE Stream NIL '(2 2))
         (DRAWLINE 0 Height 0 0 1 'REPLACE Stream NIL '(2 2))
         ;;
         ;; let the user move the region
         (SETQ NewPos (FMC-MOVE.BITMAP CompoundBitmap (GET.REGION.LEFT CompoundRegion)
                                       (GET.REGION.BOTTOM CompoundRegion)
                                       WINDOW))
         ;;
         (SETQ CorrX (IDIFFERENCE (fetch (POSITION XCOORD) of NewPos)
                                  (GET.REGION.LEFT CompoundRegion)))
         (SETQ CorrY (IDIFFERENCE (fetch (POSITION YCOORD) of NewPos)
                                  (GET.REGION.BOTTOM CompoundRegion)))
         (SETQ Left (fetch (POSITION XCOORD) of NewPos))
         (SETQ Bottom (fetch (POSITION YCOORD) of NewPos))
         (add Right CorrX)
         (add Top CorrY)
         ;; remove the surrounding rectangle
         (DRAWLINE Left Bottom Right Bottom 1 'ERASE WINDOW)
         (DRAWLINE Right Bottom Right Top 1 'ERASE WINDOW)
         (DRAWLINE Right Top Left Top 1 'ERASE WINDOW)
         (DRAWLINE Left Top Left Bottom 1 'ERASE WINDOW)
         ;; update items' regions
         (for item in ItemList do (FMC-UPDATE.REGION item CorrX CorrY WINDOW)
                                  (if (FMC-GROUP? item)
                                      then (FMC-REDRAW.ITEM item WINDOW)))
         (if (EQ UNIQUE.OR.MULTIPLE 'UNIQUE)
             then (WINDOWPROP WINDOW 'SELECTED.ITEM (CAR ItemList))
                  (WINDOWPROP WINDOW 'SELECTION.LIST NIL)
                  (FMC-SELECT.ITEM WINDOW)
           else (WINDOWPROP WINDOW 'SELECTION.LIST ItemList)
                (FMC-SELECT.LIST WINDOW])

(FMC-MOVE.BITMAP
  [LAMBDA (BITMAP BM.X BM.Y WINDOW)                      ; Edited 17-Aug-88 14:06 by A.BLAVIER
    (PROG ((CorrX (IQUOTIENT (BITMAPWIDTH BITMAP)
                             2))
           (CorrY (IQUOTIENT (BITMAPHEIGHT BITMAP)
                             2))
           (Grid (WINDOWPROP WINDOW 'GRIDSIZE))
           GridOldX GridOldY (GridNewX BM.X)
           (GridNewY BM.Y)
           OldX OldY (NewX BM.X)
```

```
          (NewY BM.Y)
          NewPos)
    ;; Performs bitmap moving under the cursor, starting at (BM.X BM.Y). Returns the new position of the bitmap.
    ;;
    ;; Set the cursor position in the center of the item's region
        (CURSORPOSITION (create POSITION
                                XCOORD _ (IPLUS BM.X CorrX)
                                YCOORD _ (IPLUS BM.Y CorrY))
              WINDOW)
    ;; Redraw the item without the selection box
        (BITBLT BITMAP 0 0 WINDOW BM.X BM.Y NIL NIL 'INPUT 'REPLACE)
    ;;
        (SETCURSOR MOVINGCURSOR)
        (SETQ OldX (IPLUS BM.X CorrX))
        (SETQ OldY (IPLUS BM.Y CorrY))
        (SETQ GridOldX BM.X)
        (SETQ GridOldY BM.Y)
    ;; Track the mouse
        [if Grid
            then ;; constrain movement along the grid
                 (while (MOUSESTATE LEFT)
                    do (SETQ NewX (LASTMOUSEX WINDOW))
                       (SETQ NewY (LASTMOUSEY WINDOW))
                       (if (OR (NOT (EQP OldX NewX))
                               (NOT (EQP OldY NewY)))
                           then (SETQ GridNewX (ITIMES (IQUOTIENT (IDIFFERENCE NewX CorrX)
                                                                  Grid)
                                                       Grid))
                                (SETQ GridNewY (ITIMES (IQUOTIENT (IDIFFERENCE NewY CorrY)
                                                                  Grid)
                                                       Grid))
                                (BITBLT BITMAP 0 0 WINDOW GridOldX GridOldY NIL NIL 'INPUT 'INVERT)
                                (BITBLT BITMAP 0 0 WINDOW GridNewX GridNewY NIL NIL 'INPUT 'INVERT)
                                (SETQ GridOldX GridNewX)
                                (SETQ GridOldY GridNewY)
                                (SETQ OldX NewX)
                                (SETQ OldY NewY)))
                 (SETQ NewPos (create POSITION
                                      XCOORD _ GridNewX
                                      YCOORD _ GridNewY))
             else ;; move freely
                  (while (MOUSESTATE LEFT) do (SETQ NewX (LASTMOUSEX WINDOW))
                                              (SETQ NewY (LASTMOUSEY WINDOW))
                                              (if (OR (NOT (EQP OldX NewX))
                                                      (NOT (EQP OldY NewY)))
                                                  then (BITBLT BITMAP 0 0 WINDOW (IDIFFERENCE OldX CorrX)
                                                               (IDIFFERENCE OldY CorrY)
                                                               NIL NIL 'INPUT 'INVERT)
                                                       (BITBLT BITMAP 0 0 WINDOW (IDIFFERENCE NewX CorrX)
                                                               (IDIFFERENCE NewY CorrY)
                                                               NIL NIL 'INPUT 'INVERT))
                                              (SETQ OldX NewX)
                                              (SETQ OldY NewY))
                  (SETQ NewPos (create POSITION
                                       XCOORD _ (IDIFFERENCE NewX CorrX)
                                       YCOORD _ (IDIFFERENCE NewY CorrY]
        (SETCURSOR DEFAULTCURSOR)
        (RETURN NewPos])
```

(FMC-TRACK.NEW.ITEM

```
  [LAMBDA (ITEM WINDOW)                                    ; Edited 17-Aug-88 14:45 by A.BLAVIER
    ;; Move a newly created item
    (LET ((ItemBitmap (FMC-GET.ITEM ITEM BITMAP))
          (ItemRegion (FMC-GET.ITEM ITEM REGION))
          (OldX (LASTMOUSEX WINDOW))
          (OldY (LASTMOUSEY WINDOW))
         NewX NewY)
         (BITBLT ItemBitmap 0 0 WINDOW (LASTMOUSEX WINDOW)
                 (LASTMOUSEY WINDOW))
         (until (MOUSESTATE LEFT) do (SETQ NewX (LASTMOUSEX WINDOW))
                                     (SETQ NewY (LASTMOUSEY WINDOW))
                                     (if (OR (NOT (EQP OldX NewX))
                                             (NOT (EQP OldY NewY)))
                                         then (BITBLT ItemBitmap 0 0 WINDOW OldX OldY NIL NIL 'INPUT 'INVERT)
                                              (BITBLT ItemBitmap 0 0 WINDOW NewX NewY NIL NIL 'INPUT 'INVERT))
                                     (SETQ OldX NewX)
                                     (SETQ OldY NewY))
         (PUT.REGION.LEFT ItemRegion NewX)
```

```
        (PUT.REGION.BOTTOM ItemRegion NewY])
```

(FMC-UPDATE.BM.POSITION

```
  [LAMBDA (ITEM WINDOW)                                    ; Edited  9-Aug-88 17:17 by A.BLAVIER
```

;; Update X and Y offset of an item's bitmap within it's box region.

```
    (LET ((Bitmap (FMC-GET.ITEM ITEM BITMAP))
          (Region (FMC-GET.ITEM ITEM REGION))
          X)
         (if (FMC-GET.ITEM ITEM BOX)
             then (SETQ X (IDIFFERENCE (IQUOTIENT (GET.REGION.WIDTH Region)
                                                  2)
                                       (IQUOTIENT (BITMAPWIDTH Bitmap)
                                                  2)))
                  (FMC-PUT.ITEM ITEM XBMOFFSET X)
                  (FMC-PUT.ITEM ITEM YBMOFFSET X])
```

(FMC-UPDATE.REGION

```
  [LAMBDA (OBJECT DELTAX DELTAY WINDOW)                    ; Edited 17-Aug-88 14:50 by A.BLAVIER
```

;; This function is called by FMC-MOVE.SELECTION and by itself to update all items'regions moved by the user.

```
    (LET ((Reg (FMC-GET.ITEM OBJECT REGION))
          (Box (FMC-GET.ITEM OBJECT BOX))
          (ObjectList (FMC-GET.ITEM OBJECT OBJECTLIST)))
         (PUT.REGION.LEFT Reg (IPLUS (GET.REGION.LEFT Reg)
                                     DELTAX))
         (PUT.REGION.BOTTOM Reg (IPLUS (GET.REGION.BOTTOM Reg)
                                       DELTAY))
         (DSPFILL Reg (FMC-GET.ITEM OBJECT BACKGROUND)
                'PAINT WINDOW)
         (if Box
             then (FMC-DRAW.BOX OBJECT Box (FMC-GET.ITEM OBJECT BOXSHADE)
                                WINDOW))
         (if (FMC-GROUP? OBJECT)
             then (for obj in ObjectList do (FMC-UPDATE.REGION obj DELTAX DELTAY WINDOW])

)


          (* * Shaping items)

(DEFINEQ
```

(FMC-COMPUTE.SHAPE.REGS

```
  [LAMBDA (WINDOW)                                         ; Edited 17-Aug-88 16:11 by A.BLAVIER
```

;; Compute the shaping regions of the selected item.

```
    (LET ((Item (WINDOWPROP WINDOW 'SELECTED.ITEM))
          Region Rleft Rbottom Rwidth Rheight HalfHeight HalfWidth Reg1 Reg2 Reg3 Reg4)
         (SETQ Region (FMC-GET.ITEM Item REGION))
         (SETQ Rleft (GET.REGION.LEFT Region))
         (SETQ Rbottom (GET.REGION.BOTTOM Region))
         (SETQ Rwidth (GET.REGION.WIDTH Region))
         (SETQ Rheight (GET.REGION.HEIGHT Region))
         (SETQ HalfHeight (IQUOTIENT Rheight 2))
         (SETQ HalfWidth (IQUOTIENT Rwidth 2))
        ;;
         (if (FMC-GET.ITEM Item BOX)
             then (SETQ Reg1 (LIST (CREATEREGION Rleft (IPLUS Rbottom HalfHeight)
                                                 HalfWidth HalfHeight)
                                   (FUNCTION FMC-SHAPE)
                                   'TOPLEFT UpperLeftCursor))
                  (SETQ Reg2 (LIST (CREATEREGION (IPLUS Rleft HalfWidth)
                                                 (IPLUS Rbottom HalfHeight)
                                                 HalfWidth HalfHeight)
                                   (FUNCTION FMC-SHAPE)
                                   'TOPRIGHT UpperRightCursor))
                  (SETQ Reg3 (LIST (CREATEREGION (IPLUS Rleft HalfWidth)
                                                 Rbottom HalfWidth HalfHeight)
                                   (FUNCTION FMC-SHAPE)
                                   'BOTTOMRIGHT LowerRightCursor))
                  (SETQ Reg4 (LIST (CREATEREGION Rleft Rbottom HalfWidth HalfHeight)
                                   (FUNCTION FMC-SHAPE)
                                   'BOTTOMLEFT LowerLeftCursor))
                  (WINDOWPROP WINDOW 'SHAPE.REGS (LIST Reg1 Reg2 Reg3 Reg4))
           else (SETQ Reg1 (LIST (CREATEREGION Rleft Rbottom Rwidth Rheight)
                                 (FUNCTION FMC-SHAPE)
                                 NIL LowerRightCursor))
                (WINDOWPROP WINDOW 'SHAPE.REGS (LIST Reg1])
```

(FMC-SHAPE

```
  [LAMBDA (WHERE WINDOW)                                   ; Edited 17-Aug-88 16:40 by A.BLAVIER
```

;; Reshape an item (simple item or group, boxed or not).

```
(LET ((Item (WINDOWPROP WINDOW 'SELECTED.ITEM))
      Box Region RLeft RBottom RRight RTop WLeft WBottom SC.Region SC.RLeft SC.RBottom SC.RRight SC.RTop
      NewRegion MinWidth MinHeight CorrX CorrY)
     (FMC-MARK.AS.CHANGED WINDOW)
     (SETQ Box (FMC-GET.ITEM Item BOX))
     (SETQ Region (FMC-GET.ITEM Item REGION))
     (SETQ RLeft (GET.REGION.LEFT Region))
     (SETQ RBottom (GET.REGION.BOTTOM Region))
     (SETQ RRight (fetch (REGION PRIGHT) of Region))
     (SETQ RTop (fetch (REGION PTOP) of Region))
     (SETQ WLeft (IPLUS (GET.REGION.LEFT (WINDOWPROP WINDOW 'REGION))
                        3))
     (SETQ WBottom (IPLUS (GET.REGION.BOTTOM (WINDOWPROP WINDOW 'REGION))
                          3))
     ;; SC stands for Screen Coordinates
     (SETQ SC.RLeft (IPLUS WLeft RLeft))
     (SETQ SC.RBottom (IPLUS WBottom RBottom))
     (SETQ SC.RRight (IPLUS WLeft RRight))
     (SETQ SC.RTop (IPLUS WBottom RTop))
     (SETQ SC.Region (CREATEREGION SC.RLeft SC.RBottom (GET.REGION.WIDTH Region)
                                   (GET.REGION.HEIGHT Region)))
     ;; compute the minimum size the item should be
     (SETQ MinWidth (IPLUS (BITMAPWIDTH (FMC-GET.ITEM Item BITMAP))
                           (if Box
                               then (IPLUS Box 3)
                             else 0)))
     (SETQ MinHeight (IPLUS (BITMAPHEIGHT (FMC-GET.ITEM Item BITMAP))
                            (if Box
                                then (IPLUS Box 3)
                              else 0)))
     (FMC-DESELECT.ITEM WINDOW)
     ;; shape the item
     (SETQ NewRegion (if (NOT Box)
                         then (GETREGION MinWidth MinHeight SC.Region (FUNCTION FMC-NOBOX.NEWREGIONFN)
                                         SC.RBottom
                                         (LIST SC.RLeft SC.RTop SC.RRight SC.RBottom))
                       else (SELECTQ WHERE
                                     (TOPLEFT (GETREGION MinWidth MinHeight SC.Region (FUNCTION
                                                                                       FMC-BOX.NEWREGIONFN)
                                                         (LIST SC.RLeft SC.RTop WHERE)
                                                         (LIST SC.RRight SC.RBottom SC.RLeft SC.RTop)))
                                     (TOPRIGHT (GETREGION MinWidth MinHeight SC.Region (FUNCTION
                                                                                        FMC-BOX.NEWREGIONFN)
                                                          (LIST SC.RRight SC.RTop WHERE)
                                                          (LIST SC.RLeft SC.RBottom SC.RRight SC.RTop)))
                                     (BOTTOMRIGHT (GETREGION MinWidth MinHeight SC.Region (FUNCTION
                                                                                           FMC-BOX.NEWREGIONFN)
                                                             (LIST SC.RRight SC.RBottom WHERE)
                                                             (LIST SC.RLeft SC.RTop SC.RRight SC.RBottom)))
                                     (BOTTOMLEFT (GETREGION MinWidth MinHeight SC.Region (FUNCTION
                                                                                          FMC-BOX.NEWREGIONFN)
                                                            (LIST SC.RLeft SC.RBottom WHERE)
                                                            (LIST SC.RRight SC.RTop SC.RLeft SC.RBottom)))
                                     NIL)))
     (FMC-CLEAR.REGION Region WINDOW)
     ;; update it's region and bitmap position
     (FMC-PUT.ITEM Item REGION (CREATEREGION (IDIFFERENCE (GET.REGION.LEFT NewRegion)
                                                          WLeft)
                                             (IDIFFERENCE (GET.REGION.BOTTOM NewRegion)
                                                          WBottom)
                                             (GET.REGION.WIDTH NewRegion)
                                             (GET.REGION.HEIGHT NewRegion)))
     (FMC-UPDATE.BM.POSITION Item WINDOW)
     ;; if item is a group we'll have to move its items according to the new shape
     (if (FMC-GROUP? Item)
         then ;; CorrX and CorrY are correction factors needed to move the items appropriately
              (SELECTQ WHERE
                   (TOPLEFT (SETQ CorrX (IQUOTIENT (IDIFFERENCE (IDIFFERENCE (GET.REGION.LEFT NewRegion)
                                                                             WLeft)
                                                                RLeft)
                                                   2))
                            (SETQ CorrY (IQUOTIENT (IDIFFERENCE (IDIFFERENCE (fetch (REGION PTOP) of NewRegion
                                                                                    )
                                                                             WBottom)
                                                                RTop)
                                                   2)))
                   (TOPRIGHT (SETQ CorrX (IQUOTIENT (IDIFFERENCE (IDIFFERENCE (fetch (REGION PRIGHT)
                                                                                 of NewRegion)
                                                                               WLeft)
                                                                  RRight)
```

```
                                        2))
                                (SETQ CorrY (IQUOTIENT (IDIFFERENCE (IDIFFERENCE (fetch (REGION PTOP)
                                                                                    of NewRegion)
                                                                            WBottom)
                                                                    RTop)
                                                            2)))
                            (BOTTOMRIGHT (SETQ CorrX (IQUOTIENT (IDIFFERENCE (IDIFFERENCE (fetch (REGION PRIGHT)
                                                                                             of NewRegion)
                                                                                    WLeft)
                                                                            RRight)
                                                                    2))
                                         (SETQ CorrY (IQUOTIENT (IDIFFERENCE (IDIFFERENCE (GET.REGION.BOTTOM
                                                                                            NewRegion)
                                                                                    WBottom)
                                                                            RBottom)
                                                                    2)))
                            (BOTTOMLEFT (SETQ CorrX (IQUOTIENT (IDIFFERENCE (IDIFFERENCE (GET.REGION.LEFT NewRegion)
                                                                                    WLeft)
                                                                            RLeft)
                                                                    2))
                                        (SETQ CorrY (IQUOTIENT (IDIFFERENCE (IDIFFERENCE (GET.REGION.BOTTOM NewRegion
                                                                                                            )
                                                                                    WBottom)
                                                                            RBottom)
                                                                    2)))
                            T)
                    (for object in (FMC-GET.ITEM Item OBJECTLIST) do (FMC-UPDATE.REGION object CorrX CorrY WINDOW))
        )
    ;;
    (FMC-REDRAW.ITEM Item WINDOW)
    (WINDOWPROP WINDOW 'SELECTED.ITEM Item)
    (FMC-SELECT.ITEM WINDOW])
```

(FMC-BOX.NEWREGIONFN

```
[LAMBDA (FIXEDPOINT MOVINGPOINT X.Y.WHERE)                    ; Edited 17-Aug-88 16:44 by A.BLAVIER
```

;; Constrain box shaping so that BOXSPACE is the same horizontally and vertically.

;; This is applied to simple boxed items or to groups

```
(PROG ((AnchorX (CAR X.Y.WHERE))
       (AnchorY (CADR X.Y.WHERE))
       (WHERE (CADDR X.Y.WHERE))
       (MovingX (fetch (POSITION XCOORD) of MOVINGPOINT))
       (MovingY (fetch (POSITION YCOORD) of MOVINGPOINT))
       DeltaX DeltaY)
      (if (NULL MOVINGPOINT)
          then (RETURN FIXEDPOINT)
        else (SELECTQ WHERE
                  ((BOTTOMLEFT TOPRIGHT)
                       (SETQ DeltaX (IDIFFERENCE MovingX AnchorX))
                       (SETQ DeltaY (IDIFFERENCE MovingY AnchorY))
                       [RETURN (CONS (IPLUS AnchorX (IMIN DeltaX DeltaY))
                                     (IPLUS AnchorY (IMIN DeltaX DeltaY])
                  ((TOPLEFT BOTTOMRIGHT)
                       (SETQ DeltaX (IDIFFERENCE MovingX AnchorX))
                       (SETQ DeltaY (IDIFFERENCE AnchorY MovingY))
                       [RETURN (CONS (IPLUS AnchorX (IMIN DeltaX DeltaY))
                                     (IDIFFERENCE AnchorY (IMIN DeltaX DeltaY])
                  NIL])
```

(FMC-NOBOX.NEWREGIONFN

```
[LAMBDA (FIXEDPOINT MOVINGPOINT Y)                            ; Edited  9-Aug-88 17:15 by A.BLAVIER
```

;; Constrain item shaping so that only its width can get changed.

;; This is applied only to non-boxed items.

```
(if (NULL MOVINGPOINT)
    then FIXEDPOINT
  else (CONS (CAR MOVINGPOINT)
             Y])

)


        (* * Redrawing items)

(DEFINEQ
```

(FMC-REDRAW

```
[LAMBDA (WINDOW REGION)                                       ; Edited 22-Jul-88 17:41 by A.BLAVIER
```

;; Redraw the entire FMC window

```
(DSPFILL NIL (WINDOWPROP WINDOW 'FMC.BACKGROUND)
         'REPLACE WINDOW)
(for ITEM in (WINDOWPROP WINDOW 'ITEMLIST) do (FMC-REDRAW.ITEM ITEM WINDOW))
```

```
    (if (WINDOWPROP WINDOW 'SELECTED.ITEM)
        then (FMC-SELECT.ITEM WINDOW))
    (if (WINDOWPROP WINDOW 'SELECTION.LIST)
        then (FMC-SELECT.LIST WINDOW))
    (if (WINDOWPROP WINDOW 'DISPLAYGRID)
        then (FMC-DISPLAY.GRID WINDOW])
```

(FMC-REDRAW.ITEM

```
  [LAMBDA (ITEM WINDOW)                                    ; Edited 17-Aug-88 16:46 by A.BLAVIER
    ;; Redraw one item (simple item or group).
    (LET ((Bitmap (FMC-GET.ITEM ITEM BITMAP))
          (Region (FMC-GET.ITEM ITEM REGION))
          (ObjectList (FMC-GET.ITEM ITEM OBJECTLIST))
          (Background (FMC-GET.ITEM ITEM BACKGROUND))
          (Box (FMC-GET.ITEM ITEM BOX))
          (BoxShade (FMC-GET.ITEM ITEM BOXSHADE))
          RLeft RBottom)
         (DSPFILL Region Background 'REPLACE WINDOW)
         (if Box
             then (FMC-DRAW.BOX ITEM Box BoxShade WINDOW))
        ;; if ITEM is a group recursively call this function for each of its subitems
         (if (FMC-GROUP? ITEM)
             then (for OBJECT in ObjectList do (FMC-REDRAW.ITEM OBJECT WINDOW))
           else (SETQ RLeft (GET.REGION.LEFT Region))
                (SETQ RBottom (GET.REGION.BOTTOM Region))
                (BITBLT Bitmap 0 0 WINDOW (IPLUS RLeft (FMC-GET.ITEM ITEM XBMOFFSET))
                       (IPLUS RBottom (FMC-GET.ITEM ITEM YBMOFFSET))
                       NIL NIL 'INPUT 'PAINT])

)


          (* * Deleting and undeleting items)

(DEFINEQ

(FMC-DELETE
  [LAMBDA (WINDOW)                                         ; Edited 17-Aug-88 16:47 by A.BLAVIER
    ;; Delete selected item(s). Save them in a list so they can get undeleted.
    (LET ((Unique? (WINDOWPROP WINDOW 'SELECTED.ITEM))
          ItemList)
         [if Unique?
             then (WINDOWPROP WINDOW 'SELECTION.LIST (LIST (WINDOWPROP WINDOW 'SELECTED.ITEM]
         (SETQ ItemList (WINDOWPROP WINDOW 'SELECTION.LIST))
         (if ItemList
             then (FMC-DESELECT.LIST WINDOW)
                  (for ITEM in ItemList do (DSPFILL (FMC-GET.ITEM ITEM REGION)
                                                    WHITESHADE
                                                    'REPLACE WINDOW)
                                           (WINDOWDELPROP WINDOW 'ITEMLIST ITEM)
                                           (WINDOWADDPROP WINDOW 'DELETED.ITEMS ITEM T))
                  (FMC-PROMPTPRINT (CONCAT (LENGTH ItemList)
                                           " item(s) deleted")
                         WINDOW)
                  (WINDOWPROP WINDOW 'SELECTED.ITEM NIL)
                  (WINDOWPROP WINDOW 'SELECTION.LIST NIL)
                  (FMC-MARK.AS.CHANGED WINDOW)
           else (FMC-PROMPTPRINT "Nothing to Delete" WINDOW])
```

(FMC-UNDELETE

```
  [LAMBDA (WHAT WINDOW)                                    ; Edited 28-Jul-88 17:56 by A.BLAVIER
    ;; Undelete deleted items : the saved list works as a LIFO structure.
    (LET ((DeletedItems (WINDOWPROP WINDOW 'DELETED.ITEMS))
          UndeletedItems)
         (if DeletedItems
             then (SELECTQ WHAT
                       (LAST (SETQ UndeletedItems (LIST (CAR DeletedItems))))
                       (ALL (SETQ UndeletedItems DeletedItems))
                       (LIST [SETQ UndeletedItems
                                  (LIST (MENU (create MENU
                                                      ITEMS _ (for ITEM in DeletedItems
                                                                 collect (LIST (CONCAT (FMC-GET.ITEM ITEM TYPE)
                                                                                       " _ "
                                                                                       (FMC-GET.ITEM ITEM LABEL))
                                                                               (KWOTE ITEM])
                                  NIL)
                  (if (CAR UndeletedItems)
                      then (for ITEM in UndeletedItems do (FMC-REDRAW.ITEM ITEM WINDOW)
                                                          (WINDOWDELPROP WINDOW 'DELETED.ITEMS ITEM)
                                                          (WINDOWADDPROP WINDOW 'ITEMLIST ITEM))
                           (FMC-PROMPTPRINT (CONCAT (LENGTH UndeletedItems)
```

```
                                                    " items(s) undeleted")
                                    WINDOW)
                                (FMC-MARK.AS.CHANGED WINDOW)
                            else (FMC-PROMPTPRINT "Nothing Undeleted" WINDOW))
                    else (FMC-PROMPTPRINT "Nothing to Undelete" WINDOW])

)

          (* * GROUPing and UNGROUPing)

(DEFINEQ
```

(FMC-GROUP

```
  [LAMBDA (WINDOW)                                          ; Edited 17-Aug-88 16:50 by A.BLAVIER
    ;; Group a multiple selection.
    (LET ((ObjectList (WINDOWPROP WINDOW 'SELECTION.LIST))
          Group GroupRegion GroupBitmap)
         (if (IGREATERP (LENGTH ObjectList)
                        1)
             then (FMC-MARK.AS.CHANGED WINDOW)
                  (FMC-DESELECT WINDOW)
                  ;; remove items from the ITEMLIST
                  (for object in ObjectList do (WINDOWDELPROP WINDOW 'ITEMLIST object))
                  ;; a group is an FMC-ITEM whose TYPE is GROUP and whose OBJECTLIST is a list of items (or groups)
                  (SETQ Group (create FMC-ITEM))
                  (FMC-PUT.ITEM Group TYPE 'GROUP)
                  (FMC-PUT.ITEM Group OBJECTLIST ObjectList)
                  ;; the group's region is the union region of its items
                  [SETQ GroupRegion (APPLY 'UNIONREGIONS (for OBJECT in ObjectList collect (FMC-GET.ITEM OBJECT
                                                                                                REGION]
                  (FMC-PUT.ITEM Group REGION (CREATEREGION (IDIFFERENCE (GET.REGION.LEFT GroupRegion)
                                                                        2)
                                                           (IDIFFERENCE (GET.REGION.BOTTOM GroupRegion)
                                                                  2)
                                                           (IPLUS (GET.REGION.WIDTH GroupRegion)
                                                                  4)
                                                           (IPLUS (GET.REGION.HEIGHT GroupRegion)
                                                                  4)))
                  ;; the group's bitmap is the compound of its items' bitmaps
                  (SETQ GroupBitmap (FMC-COMPOUND.BITMAP GroupRegion ObjectList))
                  (FMC-PUT.ITEM Group BITMAP GroupBitmap)
                  (FMC-PUT.ITEM Group LABEL GroupBitmap)
                  ;; default boxing is black 1 pixel thick
                  (FMC-PUT.ITEM Group BOX 1)
                  (FMC-PUT.ITEM Group BOXSHADE 65535)
                  (FMC-PUT.ITEM Group XBMOFFSET 2)
                  (FMC-PUT.ITEM Group YBMOFFSET 2)
                  ;;
                  (FMC-REDRAW.ITEM Group WINDOW)
                  (WINDOWADDPROP WINDOW 'ITEMLIST Group)
                  (WINDOWPROP WINDOW 'SELECTED.ITEM Group)
                  (FMC-SELECT.ITEM WINDOW])
```

(FMC-UNGROUP

```
  [LAMBDA (WINDOW)                                          ; Edited 17-Aug-88 16:50 by A.BLAVIER
    ;; Unpack a group.
    ;; For safety, this function works only on a unique selection
    (LET ((Group (WINDOWPROP WINDOW 'SELECTED.ITEM))
          ObjectList)
         (SETQ ObjectList (FMC-GET.ITEM Group OBJECTLIST))
         (if (AND Group (FMC-GROUP? Group))
             then (FMC-MARK.AS.CHANGED WINDOW)
                  (WINDOWDELPROP WINDOW 'ITEMLIST Group)
                  (FMC-DESELECT.ITEM WINDOW)
                  (FMC-CLEAR.REGION (FMC-GET.ITEM Group REGION)
                         WINDOW)
                  (for OBJECT in ObjectList do (WINDOWADDPROP WINDOW 'ITEMLIST OBJECT)
                                               (WINDOWADDPROP WINDOW 'SELECTION.LIST OBJECT)
                                               (FMC-REDRAW.ITEM OBJECT WINDOW))
                  (FMC-SELECT.LIST WINDOW])

)

          (* * Align and Center functions)

(DEFINEQ
```

(FMC-ALIGN

```
[LAMBDA (ALIGN.TYPE WINDOW)                                    ; Edited 17-Aug-88 16:51 by A.BLAVIER
    ;; Align items of a multiple selection.
    ;; Reference is the first selected item, that is : the first item won't get moved.
    (LET ((ItemList (WINDOWPROP WINDOW 'SELECTION.LIST))
          RefEdge RegionOfRefItem)
         (FMC-MARK.AS.CHANGED WINDOW)
         (if (IGREATERP (LENGTH ItemList)
                        1)
             then (FMC-DESELECT.LIST WINDOW)
                  (SETQ RegionOfRefItem (FMC-GET.ITEM (CAR ItemList)
                                                      REGION))
                  (SETQ RefEdge (SELECTQ ALIGN.TYPE
                                         (ALIGNLEFT (GET.REGION.LEFT RegionOfRefItem))
                                         (ALIGNBOTTOM (GET.REGION.BOTTOM RegionOfRefItem))
                                         (ALIGNTOP (IPLUS (GET.REGION.BOTTOM RegionOfRefItem)
                                                          (GET.REGION.HEIGHT RegionOfRefItem)))
                                         (ALIGNRIGHT (IPLUS (GET.REGION.LEFT RegionOfRefItem)
                                                            (GET.REGION.WIDTH RegionOfRefItem)))
                                        T))
                  (for item in (CDR ItemList) bind region deltaX deltaY
                     do (SETQ region (FMC-GET.ITEM item REGION))
                        (SETQ deltaX 0)
                        (SETQ deltaY 0)
                        (FMC-CLEAR.REGION region WINDOW)
                        (SELECTQ ALIGN.TYPE
                             (ALIGNLEFT (SETQ deltaX (IDIFFERENCE RefEdge (GET.REGION.LEFT region)))
                                        (PUT.REGION.LEFT region RefEdge))
                             (ALIGNBOTTOM (SETQ deltaY (IDIFFERENCE RefEdge (GET.REGION.BOTTOM region)))
                                          (PUT.REGION.BOTTOM region RefEdge))
                             (ALIGNTOP (SETQ deltaY (IDIFFERENCE RefEdge (fetch (REGION PTOP) of region)))
                                       (PUT.REGION.BOTTOM region (IDIFFERENCE RefEdge (GET.REGION.HEIGHT region)
                                                                              )))
                             (ALIGNRIGHT (SETQ deltaX (IDIFFERENCE RefEdge (fetch (REGION PRIGHT) of region)))
                                         (PUT.REGION.LEFT region (IDIFFERENCE RefEdge (GET.REGION.WIDTH region)))
)
                             T)
                        (if (FMC-GROUP? item)
                            then ;; if item is a group its subitems have to moved proportionnally
                                 (FMC-REL.MOVE item deltaX deltaY))
                        (FMC-REDRAW.ITEM item WINDOW))
                  (WINDOWPROP WINDOW 'SELECTION.LIST ItemList)
                  (FMC-SELECT.LIST WINDOW])
```

(FMC-HCENTER

```
[LAMBDA (WINDOW)                                               ; Edited 17-Aug-88 16:52 by A.BLAVIER
    ;; Center items of a multiple selection horizontally.
    ;;  Reference is the first selected item.
    (LET ((ItemList (WINDOWPROP WINDOW 'SELECTION.LIST))
          HPos RegionOfRefItem)
         (FMC-MARK.AS.CHANGED WINDOW)
         (if (IGREATERP (LENGTH ItemList)
                        1)
             then (FMC-DESELECT.LIST WINDOW)
                  (SETQ RegionOfRefItem (FMC-GET.ITEM (CAR ItemList)
                                                      REGION))
                  (SETQ HPos (IPLUS (GET.REGION.LEFT RegionOfRefItem)
                                    (IQUOTIENT (GET.REGION.WIDTH RegionOfRefItem)
                                               2)))
                  (for item in (CDR ItemList) bind region deltaX
                     do (SETQ region (FMC-GET.ITEM item REGION))
                        (FMC-CLEAR.REGION region WINDOW)
                        (SETQ deltaX (GET.REGION.LEFT region))
                        (PUT.REGION.LEFT region (IDIFFERENCE (GET.REGION.LEFT region)
                                                             (IDIFFERENCE (IPLUS (GET.REGION.LEFT region)
                                                                                 (IQUOTIENT (GET.REGION.WIDTH region)
                                                                                            2))
                                                                          HPos)))
                        (SETQ deltaX (IDIFFERENCE (GET.REGION.LEFT region)
                                                  deltaX))
                        (if (FMC-GROUP? item)
                            then ;; if item is a group its subitems have to be moved proportionnally
                                 (FMC-REL.MOVE item deltaX 0))
                        (FMC-REDRAW.ITEM item WINDOW))
                  (WINDOWPROP WINDOW 'SELECTION.LIST ItemList)
                  (FMC-SELECT.LIST WINDOW])
```

(FMC-VCENTER

```
[LAMBDA (WINDOW)                                               ; Edited 17-Aug-88 16:53 by A.BLAVIER
```

```
    ;; Center items vertically.
    ;; Reference is the first selected item.
    (LET ((ItemList (WINDOWPROP WINDOW 'SELECTION.LIST))
          VPos RegionOfRefItem)
         (FMC-MARK.AS.CHANGED WINDOW)
         (if (IGREATERP (LENGTH ItemList)
                        1)
             then (FMC-DESELECT.LIST WINDOW)
                  (SETQ RegionOfRefItem (FMC-GET.ITEM (CAR ItemList)
                                                      REGION))
                  (SETQ VPos (IPLUS (GET.REGION.BOTTOM RegionOfRefItem)
                                    (IQUOTIENT (GET.REGION.HEIGHT RegionOfRefItem)
                                               2)))
                  (for item in (CDR ItemList) bind region deltaY
                     do (SETQ region (FMC-GET.ITEM item REGION))
                        (FMC-CLEAR.REGION region WINDOW)
                        (SETQ deltaY (GET.REGION.BOTTOM region))
                        (PUT.REGION.BOTTOM region (IDIFFERENCE (GET.REGION.BOTTOM region)
                                                               (IDIFFERENCE (IPLUS (GET.REGION.BOTTOM region)
                                                                                   (IQUOTIENT (GET.REGION.HEIGHT
                                                                                                region)
                                                                                              2))
                                                                            VPos)))
                        (SETQ deltaY (IDIFFERENCE (GET.REGION.BOTTOM region)
                                                  deltaY))
                        (if (FMC-GROUP? item)
                            then ;; if item is a group its subitems have to be moved proportionnally
                                 (FMC-REL.MOVE item 0 deltaY))
                        (FMC-REDRAW.ITEM item WINDOW))
                  (WINDOWPROP WINDOW 'SELECTION.LIST ItemList)
                  (FMC-SELECT.LIST WINDOW])
```

(FMC-REL.MOVE

```
  [LAMBDA (GROUP DELTAX DELTAY)                                    ; Edited 3-Aug-88 16:27 by A.BLAVIER
    ;; Move the items of a group when the group itself has been moved by a Center or Align operation.
    (LET ((ObjectList (FMC-GET.ITEM GROUP OBJECTLIST)))
         (for item in ObjectList bind Region do (SETQ Region (FMC-GET.ITEM item REGION))
                                                (PUT.REGION.LEFT Region (IPLUS (GET.REGION.LEFT Region)
                                                                               DELTAX))
                                                (PUT.REGION.BOTTOM Region (IPLUS (GET.REGION.BOTTOM Region)
                                                                                 DELTAY))
                                                (if (FMC-GROUP? item)
                                                    then (FMC-REL.MOVE item DELTAX DELTAY])

)


          (* * File saving and loading)

(DEFINEQ

(FMC-GET
  [LAMBDA (WINDOW)                                                 ; Edited 18-Aug-88 11:48 by A.BLAVIER
    ;; Read item(s) from a FMC file and add them to the window.
    (LET ((PromptWindow (WINDOWPROP WINDOW 'FMC.PROMPTWINDOW))
          FileName Stream Expr Item)
         (FMC-PROMPTPRINT "" WINDOW)
        ;; input the file name
        [SETQ FileName (PROMPTFORWORD "File to GET :" FileName NIL PromptWindow NIL NIL
                                      (CHARCODE (EOL ESCAPE LF TAB]
        (if FileName
            then ;; check that FileName exists
                 (if (FINDFILE FileName)
                     then (RESETLST
                              (SETQ Stream (OPENSTREAM FileName 'INPUT))
                              (RESETSAVE (CURSOR WAITINGCURSOR))
                              (RESETSAVE NIL (LIST 'CLOSEF Stream))
                              ;; check that the file is actually a FMC file
                              (if (NEQ (READ Stream)
                                       'FreeMenuCreator-ItemList)
                                  then (FMC-PROMPTPRINT "Not a Free Menu Creator file" WINDOW)
                                  else (FMC-DESELECT WINDOW)
                                       (FMC-PROMPTPRINT (CONCAT "GETting file " (FULLNAME Stream)
                                                                "...")
                                              WINDOW)
                                       ;; read the items in
                                       (until (EQ (SETQ Item (FMC-GET.ONE.OBJECT Stream))
                                                  'STOP)
```

```
                                    do (WINDOWADDPROP WINDOW 'ITEMLIST Item)
                                       (WINDOWADDPROP WINDOW 'SELECTION.LIST Item))
                                 (FMC-REDRAW WINDOW)
                                 (FMC-PROMPTPRINT (CONCAT "GETting file " (FULLNAME Stream)
                                                          "...done")
                                       WINDOW)))
                          ;; if the window was not (file)named yet, change it's title
                          (if (NOT (WINDOWPROP WINDOW 'FILENAME))
                              then (WINDOWPROP WINDOW 'FILENAME FileName)
                                   (WINDOWPROP WINDOW 'TITLE FileName))
                          (FMC-MARK.AS.CHANGED WINDOW)
                    else (FMC-PROMPTPRINT (CONCAT "Couldn't find file " FileName)
                                WINDOW))
          else (FMC-PROMPTPRINT "GET aborted" WINDOW])
```

(FMC-GET.ONE.OBJECT

```
[LAMBDA (STREAM)                                        ; Edited 12-Aug-88 11:33 by A.BLAVIER
    ;; Read in one item from STREAM
    (PROG (Expr Item Bitmap CompoundRegion ObjectListSize)
          (SETQ Expr (READ STREAM))
          (SETQ ObjectListSize (CAR (NTH Expr 13)))
          (if (NEQ Expr 'STOP)
              then (SETQ Item (FMC-CREATE.ITEM.FROM.LIST Expr))
                   (FMC-PUT.ITEM Item OBJECTLIST NIL)
                   [COND
                      ((NEQ (CAR Expr)
                            'GROUP)
                       ;; if it's not a group read its bitmap if any. If none compute it from its label and font
                       [if (EQ (CADR Expr)
                               '**BITMAP**)
                           then (SETQ Bitmap (HREAD STREAM))
                                (FMC-PUT.ITEM Item LABEL Bitmap)
                         else (SETQ Bitmap (FMC-MAKEBITMAP (CADR Expr)
                                                           (CADDR Expr]
                       (FMC-PUT.ITEM Item BITMAP Bitmap))
                      (T ;; it's a group : read its items and build the OBJECTLIST
                         (FMC-PUT.ITEM Item OBJECTLIST (for i from 1 to ObjectListSize collect (FMC-GET.ONE.OBJECT
                                                                                                STREAM)))
                         ;; compute its bitmap
                         [SETQ CompoundRegion (APPLY 'UNIONREGIONS (for i in (FMC-GET.ITEM Item OBJECTLIST)
                                                                      collect (FMC-GET.ITEM i REGION]
                         (FMC-PUT.ITEM Item BITMAP (FMC-COMPOUND.BITMAP CompoundRegion (FMC-GET.ITEM Item
                                                                                              OBJECTLIST)))
                         (FMC-PUT.ITEM Item LABEL (FMC-GET.ITEM Item BITMAP]
                   (RETURN Item)
            else (RETURN 'STOP])
```

(FMC-PUT

```
[LAMBDA (WINDOW)                                        ; Edited 12-Aug-88 13:35 by A.BLAVIER
    ;; Save the contents of WINDOW onto a file.
    (LET ((PromptWindow (WINDOWPROP WINDOW 'FMC.PROMPTWINDOW))
          (ItemList (WINDOWPROP WINDOW 'ITEMLIST))
          (FileName (WINDOWPROP WINDOW 'FILENAME))
          Stream)
         (if ItemList
             then (FMC-PROMPTPRINT "" WINDOW)
                  ;; input the file name
                  [SETQ FileName (PROMPTFORWORD "File to PUT to :" FileName NIL PromptWindow NIL NIL
                                       (CHARCODE (EOL ESCAPE LF TAB]
                  (if FileName
                      then (RESETLST
                                (SETQ Stream (OPENSTREAM FileName 'OUTPUT))
                                (RESETSAVE (CURSOR WAITINGCURSOR))
                                (RESETSAVE NIL (LIST 'CLOSEF Stream))
                                (FMC-PROMPTPRINT (CONCAT "PUTting file " (FULLNAME Stream)
                                                         "...")
                                      WINDOW)
                                ;; write the FMC header
                                (PRINTOUT Stream 'FreeMenuCreator-ItemList T)
                                ;; write the items
                                (for item in ItemList do (FMC-PUT.OBJECT item Stream))
                                (PRINTOUT Stream 'STOP)
                                (FMC-PROMPTPRINT (CONCAT "PUTting file " (FULLNAME Stream)
                                                         "...done")
                                      WINDOW))
                            ;; change the window title
```

```
                        (WINDOWPROP WINDOW 'FILENAME FileName)
                        (WINDOWPROP WINDOW 'TITLE FileName)
                        (WINDOWPROP WINDOW 'FMC.CHANGED NIL)
                    else (FMC-PROMPTPRINT "PUT aborted" WINDOW])
```

(FMC-PUT.OBJECT

```
  [LAMBDA (ITEM STREAM)                                    ; Edited 12-Aug-88 10:33 by A.BLAVIER
    ;; write one item on STREAM
    (LET ((ObjectList (FMC-GET.ITEM ITEM OBJECTLIST))
          (Label (FMC-GET.ITEM ITEM LABEL))
          (Message (FMC-GET.ITEM ITEM MESSAGE)))
         (PRINTOUT STREAM (LIST (FMC-GET.ITEM ITEM TYPE)
                                (if (BITMAPP Label)
                                    then "**BITMAP**"
                                  else (CONCAT '%" Label '%"))
                                (FMC-GET.ITEM ITEM FONT)
                                (FMC-GET.ITEM ITEM ID)
                                (FMC-GET.ITEM ITEM COLLECTION)
                                (FMC-GET.ITEM ITEM DESELECT)
                                (if Message
                                    then (CONCAT '%" Message '%")
                                  else (CONCAT '%" '%"))
                                (FMC-GET.ITEM ITEM INITSTATE)
                                (FMC-GET.ITEM ITEM BOX)
                                (FMC-GET.ITEM ITEM BOXSHADE)
                                (FMC-GET.ITEM ITEM BACKGROUND)
                                (FMC-GET.ITEM ITEM LINKS)
                                (LENGTH ObjectList)
                                (FMC-GET.ITEM ITEM MENU)
                                (FMC-GET.ITEM ITEM CHANGESTATE)
                                (FMC-GET.ITEM ITEM SELECTEDFN)
                                (FMC-GET.ITEM ITEM DOWNFN)
                                (FMC-GET.ITEM ITEM HELDFN)
                                (FMC-GET.ITEM ITEM MOVEDFN)
                                (FMC-GET.ITEM ITEM REGION)
                                NIL
                                (FMC-GET.ITEM ITEM XBMOFFSET)
                                (FMC-GET.ITEM ITEM YBMOFFSET)
                                (FMC-GET.ITEM ITEM INFINITEWIDTH)
                                (FMC-GET.ITEM ITEM USERDATA))
                T)
         (if (AND (BITMAPP Label)
                  (NOT (FMC-GROUP? ITEM)))
             then (HPRINT Label STREAM T T))
         (if (FMC-GROUP? ITEM)
             then ;; ITEM is a group : write its subitems
                  (for object in ObjectList do (FMC-PUT.OBJECT object STREAM])

)


          (* * Creating a summary)

(DEFINEQ
```

(FMC-EDIT.INFO

```
  [LAMBDA (WINDOW)                                         ; Edited  1-Feb-2022 17:08 by rmk
                                                           ; Edited 17-Aug-88 16:57 by A.BLAVIER
    ;; Create a "dead" TEdit window, listing a summary of the items.
    (LET ((ItemList (WINDOWPROP WINDOW 'ITEMLIST))
          Stream TEdWindow)
         (RESETLST
             (RESETSAVE (CURSOR WAITINGCURSOR))
             (SETQ Stream (OPENTEXTSTREAM NIL))
             (RESETSAVE NIL (LIST 'CLOSEF Stream))
             (FMC-PROMPTPRINT "Creating summary ..." WINDOW)
             (SETCURSOR WAITINGCURSOR)
             (FMC-SORT.ITEM.LIST ItemList)
             ;;
             (PRINTOUT Stream .FONT '(MODERN 14 BOLD)
                    "- Free Menu Creator Summary -" T T)
             (PRINTOUT Stream .FONT '(MODERN 10 REGULAR)
                    (DATE)
                    T T)
             (for item in ItemList do (FMC-EDIT.INFO.ITEM item Stream 0))
             (TEDIT.PARALOOKS Stream '(QUAD CENTERED)
                    1 2)
             (SETCURSOR DEFAULTCURSOR)
             (FMC-PROMPTPRINT "Creating summary ... done" WINDOW)
             (SETQ TEdWindow (CREATEW NIL "FMC Items Summary"))
             (OPENTEXTSTREAM Stream TEdWindow))])
```

(FMC-EDIT.INFO.ITEM

```
  [LAMBDA (ITEM STREAM SPACES)                                    ; Edited  8-Aug-88 17:00 by A.BLAVIER
    ;; This function is called by FMC-EDIT.INFO (and by itself). Writes on STREAM interesting item properties.
    (LET ((ID (FMC-GET.ITEM ITEM ID))
          (LABEL (FMC-GET.ITEM ITEM LABEL))
          (TYPE (FMC-GET.ITEM ITEM TYPE))
          (FONT (FMC-GET.ITEM ITEM FONT))
          (CHANGESTATE (FMC-GET.ITEM ITEM CHANGESTATE))
          (SELECTEDFN (FMC-GET.ITEM ITEM SELECTEDFN))
          (DOWNFN (FMC-GET.ITEM ITEM DOWNFN))
          (HELDFN (FMC-GET.ITEM ITEM HELDFN))
          (MOVEDFN (FMC-GET.ITEM ITEM MOVEDFN))
          (MENU (FMC-GET.ITEM ITEM MENU))
          (INITSTATE (FMC-GET.ITEM ITEM INITSTATE))
          (LINKS (FMC-GET.ITEM ITEM LINKS))
          (INFINITEWIDTH (FMC-GET.ITEM ITEM INFINITEWIDTH))
          (MESSAGE (FMC-GET.ITEM ITEM MESSAGE))
          (OBJECTLIST (FMC-GET.ITEM ITEM OBJECTLIST)))
         (if (FMC-GROUP? ITEM)
             then (PRINTOUT STREAM .SP SPACES .FONT '(MODERN 12 BOLD)
                         "GROUP - " .FONT '(MODERN 10 BOLDITALIC)
                         " ID : " .FONT '(MODERN 10 REGULAR)
                         ID T)
                  (for OBJECT in (FMC-SORT.ITEM.LIST OBJECTLIST) do (FMC-EDIT.INFO.ITEM OBJECT STREAM
                                                                           (IPLUS SPACES 5)))
           else (PRINTOUT STREAM .SP SPACES .FONT '(MODERN 10 BOLD)
                       "ITEM - " .FONT '(MODERN 10 BOLDITALIC)
                       " ID : " .FONT '(MODERN 10 REGULAR)
                       ID .FONT '(MODERN 10 BOLDITALIC)
                       " - LABEL : " .FONT '(MODERN 10 REGULAR)
                       LABEL T .SP SPACES .SP 5 .FONT '(MODERN 10 BOLDITALIC)
                       "TYPE : " .FONT '(MODERN 10 REGULAR)
                       TYPE T)
                (if [NOT (FMEMB TYPE '(DISPLAY EDIT NUMBER]
                    then (if (EQ TYPE 'STATE)
                             then (if [NOT (EQUAL MENU '(NIL]
                                      then (PRINTOUT STREAM .SP SPACES .SP 5 .FONT '(MODERN 10 BOLDITALIC)
                                                  "MENU : " .FONT '(MODERN 10 REGULAR)
                                                  MENU T))
                                  (if [NOT (EQUAL CHANGESTATE '(FUNCTION NILL]
                                      then (PRINTOUT STREAM .SP SPACES .SP 5 .FONT '(MODERN 10 BOLDITALIC)
                                                  "CHANGESTATE : " .FONT '(MODERN 10 REGULAR)
                                                  CHANGESTATE T)))
                         (for prop in '(SELECTEDFN DOWNFN HELDFN MOVEDFN) bind Def
                            do (SETQ Def (EVAL prop))
                               (if [NOT (EQUAL Def '(FUNCTION NILL]
                                   then (PRINTOUT STREAM .SP SPACES .SP 5 .FONT '(MODERN 10 BOLDITALIC)
                                               (CONCAT prop " : ")
                                               .FONT
                                               '(MODERN 10 REGULAR)
                                               Def T)))
                         (if INITSTATE
                             then (PRINTOUT STREAM .SP SPACES .SP 5 .FONT '(MODERN 10 BOLDITALIC)
                                         "INITSTATE : " .FONT '(MODERN 10 REGULAR)
                                         INITSTATE T))
                         (if LINKS
                             then (PRINTOUT STREAM .SP SPACES .SP 5 .FONT '(MODERN 10 BOLDITALIC)
                                         "LINKS : " .FONT '(MODERN 10 REGULAR)
                                         LINKS T)))
                (if (FMEMB TYPE '(EDIT NUMBER))
                    then (PRINTOUT STREAM .SP SPACES .SP 5 .FONT '(MODERN 10 BOLDITALIC)
                              "INFINITEWIDTH : " .FONT '(MODERN 10 REGULAR)
                              INFINITEWIDTH T])

)


          (* * Hardcopy functions)

(DEFINEQ

(FMC-HARDCOPY
  [LAMBDA (WINDOW STREAM)                                         ; Edited 17-Aug-88 17:01 by A.BLAVIER
    ;; Make a centered hardcopy of the contents of the FMC window.
    (LET ((ItemList (WINDOWPROP WINDOW 'ITEMLIST))
          (PageScale (DSPSCALE NIL STREAM))
          (Xmin 5000)
          (Xmax 0)
          (Ymin 5000)
          (Ymax 0)
         PageWidth PageHeight FMCWidth FMCHeight Xoffset Yoffset 2points 5points 10points)
        (FMC-PROMPTPRINT "Formatting for print ..." WINDOW)    ; compute X and Y offsets
        (for item in ItemList bind (region left bottom right top) do (SETQ region (FMC-GET.ITEM item REGION))
```

```
                                                    (SETQ left (GET.REGION.LEFT region))
                                                    (SETQ bottom (GET.REGION.BOTTOM region))
                                                    (SETQ right (fetch (REGION RIGHT)
                                                                   of region))
                                                    (SETQ top (fetch (REGION TOP) of region))
                                                    (if (ILESSP left Xmin)
                                                        then (SETQ Xmin left))
                                                    (if (ILESSP bottom Ymin)
                                                        then (SETQ Ymin bottom))
                                                    (if (IGREATERP right Xmax)
                                                        then (SETQ Xmax right))
                                                    (if (IGREATERP top Ymax)
                                                        then (SETQ Ymax top)))
        (SETQ PageWidth (FIX (TIMES 8.27 72 PageScale)))
        (SETQ PageHeight (FIX (TIMES 11.69 72 PageScale)))
        (SETQ FMCWidth (ITIMES PageScale (IDIFFERENCE Xmax Xmin)))
        (SETQ FMCHeight (ITIMES PageScale (IDIFFERENCE Ymax Ymin)))
        (SETQ Xoffset (IQUOTIENT (IDIFFERENCE PageWidth FMCWidth)
                                 2))
        (if (ILESSP Xoffset 0)
            then (SETQ Xoffset 0))
        (SETQ Yoffset (IQUOTIENT (IDIFFERENCE PageHeight FMCHeight)
                                 2))
        (SETQ 2points (ITIMES PageScale 2))
        (SETQ 5points (ITIMES PageScale 5))
        (SETQ 10points (ITIMES PageScale 10))
        (if (ILESSP Yoffset 0)
            then (SETQ Yoffset 0))
```

;; draw a rectangle around the items

```
        (DRAWLINE (IDIFFERENCE Xoffset 5points)
                  (IDIFFERENCE Yoffset 5points)
                  (IPLUS Xoffset FMCWidth 5points)
                  (IDIFFERENCE Yoffset 5points)
                  2points
                  'REPLACE STREAM)
        (RELDRAWTO 0 (IPLUS FMCHeight 10points)
                  2points
                  'REPLACE STREAM)
        (RELDRAWTO (MINUS (IPLUS FMCWidth 10points))
                  0 2points 'REPLACE STREAM)
        (RELDRAWTO 0 (MINUS (IPLUS FMCHeight 10points))
                  2points
                  'REPLACE STREAM)
```

;; adjust X and Y offsets so that the lower left item gets printed in the lower left corner of the area

```
        (SETQ Xoffset (IDIFFERENCE Xoffset (ITIMES PageScale Xmin)))
        (SETQ Yoffset (IDIFFERENCE Yoffset (ITIMES PageScale Ymin)))
```

;; draw the items

```
        (for item in ItemList do (FMC-HARDCOPY.ITEM item STREAM Xoffset Yoffset))
        (FMC-PROMPTPRINT (CONCAT "Formatting for print ... " " done")
                  WINDOW])
```

(FMC-HARDCOPY.ITEM

```
  [LAMBDA (ITEM STREAM XOFFSET YOFFSET)                    ; Edited 17-Aug-88 17:12 by A.BLAVIER
```

;; Recursive function initially called by FMC-HARDCOPY.

```
    (LET ((Label (FMC-GET.ITEM ITEM LABEL))
          (Font (FMC-GET.ITEM ITEM FONT))
          (Bitmap (FMC-GET.ITEM ITEM BITMAP))
          (Region (FMC-GET.ITEM ITEM REGION))
          (Box (FMC-GET.ITEM ITEM BOX))
          (BoxShade (FMC-GET.ITEM ITEM BOXSHADE))
          (ObjectList (FMC-GET.ITEM ITEM OBJECTLIST))
          (PageScale (DSPSCALE NIL STREAM))
          RLeft RBottom RWidth RHeight RRight RTop)
         [SETQ RLeft (IPLUS XOFFSET (ITIMES PageScale (GET.REGION.LEFT Region]
         [SETQ RBottom (IPLUS YOFFSET (ITIMES PageScale (GET.REGION.BOTTOM Region]
         (SETQ RWidth (ITIMES PageScale (GET.REGION.WIDTH Region)))
         (SETQ RHeight (ITIMES PageScale (GET.REGION.HEIGHT Region)))
         (SETQ RRight (IPLUS RLeft RWidth))
         (SETQ RTop (IPLUS RBottom RHeight))
         (SETQ Region (CREATEREGION RLeft RBottom RWidth RHeight))
         (DSPFILL Region (FMC-GET.ITEM ITEM BACKGROUND)
                  'REPLACE STREAM)
         (if Box
             then (SETQ Box (ITIMES PageScale Box))
                  (DSPFILL (CREATEREGION RLeft RBottom Box RHeight)
                           BoxShade
                           'REPLACE STREAM)
                  (DSPFILL (CREATEREGION RLeft (IDIFFERENCE RTop Box)
                                         RWidth Box)
                           BoxShade
                           'REPLACE STREAM)
                  (DSPFILL (CREATEREGION (IDIFFERENCE RRight Box)
```

```
                                RBottom Box RHeight)
                        BoxShade
                        'REPLACE STREAM)
                (DSPFILL (CREATEREGION RLeft RBottom RWidth Box)
                        BoxShade
                        'REPLACE STREAM))
        (if (FMC-GROUP? ITEM)
            then (for object in ObjectList do (FMC-HARDCOPY.ITEM object STREAM XOFFSET YOFFSET))
          else (if (BITMAPP Label)
                   then [BITBLT Bitmap 0 0 STREAM (IPLUS RLeft (ITIMES PageScale (FMC-GET.ITEM ITEM XBMOFFSET)))
                                (IPLUS RBottom (ITIMES PageScale (FMC-GET.ITEM ITEM YBMOFFSET]
                 else (MOVETO (IPLUS RLeft (ITIMES PageScale (FMC-GET.ITEM ITEM XBMOFFSET)))
                              [IPLUS RBottom (ITIMES PageScale (FMC-GET.ITEM ITEM YBMOFFSET))
                                     (ITIMES PageScale (FONTPROP Font 'DESCENT]
                              STREAM)
                      (PRINTOUT STREAM .FONT Font Label])

)


          (* * Creating the desription list)

(DEFINEQ

(FMC-COMPUTE
  [LAMBDA (WINDOW)                                          ; Edited 12-Aug-88 13:49 by A.BLAVIER
    ;; Compute a description list suitable for the FREEMENU function.
    ;; Store the list in the global variable FM-DESCRIPTION.
    ;; Create and open a real Free Menu window built out of the description.
    (LET ((Background (WINDOWPROP WINDOW 'FMC.BACKGROUND))
          Description FM-WINDOW)
         (RESETLST
             (RESETSAVE (CURSOR WAITINGCURSOR))
             (FMC-PROMPTPRINT "COMPUTING ... " WINDOW)
             ;; build the description list
             (SETQ Description '(PROPS FORMAT EXPLICIT))
             [if (AND Background (NOT (EQP Background 0)))
                 then (SETQ Description (APPEND Description `(BACKGROUND ,Background]
             [SETQ Description (CONS Description (for object in (FMC-SORT.ITEM.LIST (WINDOWPROP WINDOW
                                                                                        'ITEMLIST))
                                                                collect (FMC-COMPUTE.OBJECT object]
             (SETQ FM-DESCRIPTION Description)
             ;; create the Free Menu
             (SETQ FM-WINDOW (FREEMENU Description))
             (FMC-PROMPTPRINT "COMPUTING ... DONE" WINDOW))
         (MOVEW FM-WINDOW (GETBOXPOSITION (WINDOWPROP FM-WINDOW 'WIDTH)
                                  (WINDOWPROP FM-WINDOW 'HEIGHT)
                                  (IDIFFERENCE (CAR (WINDOWPROP WINDOW 'REGION))
                                         0)
                                  (IDIFFERENCE (CADR (WINDOWPROP WINDOW 'REGION))
                                         0)))
         (OPENW FM-WINDOW)
         (TOTOPW FM-WINDOW])

(FMC-COMPUTE.OBJECT
  [LAMBDA (OBJECT COLLECTIONID DESELECT)                    ; Edited 11-Aug-88 16:55 by A.BLAVIER
    ;; This recursive function computes the description sublist of OBJECT [Group or Item]
    (PROG (Region Type Label Id Box Boxshade Font CHANGESTATE SELECTEDFN DOWNFN HELDFN MOVEDFN Menu Links
                 Initstate Background InfiniteWidth ObjectList TmpLst)
          (SETQ Region (FMC-GET.ITEM OBJECT REGION))
          (SETQ Type (FMC-GET.ITEM OBJECT TYPE))
          (SETQ Label (if (STREQUAL (FMC-GET.ITEM OBJECT LABEL)
                                    "*NOLABEL*")
                          then ""
                        else (FMC-GET.ITEM OBJECT LABEL)))
          (SETQ Id (FMC-GET.ITEM OBJECT ID))
          (if (FMC-GROUP? OBJECT)
              then (SETQ Collection (FMC-GET.ITEM OBJECT COLLECTION)))
          (SETQ Box (FMC-GET.ITEM OBJECT BOX))
          (SETQ Boxshade (FMC-GET.ITEM OBJECT BOXSHADE))
          (SETQ Font (FMC-GET.ITEM OBJECT FONT))
          (SETQ CHANGESTATE (FMC-GET.ITEM OBJECT CHANGESTATE))
          (SETQ SELECTEDFN (FMC-GET.ITEM OBJECT SELECTEDFN))
          (SETQ DOWNFN (FMC-GET.ITEM OBJECT DOWNFN))
          (SETQ HELDFN (FMC-GET.ITEM OBJECT HELDFN))
          (SETQ MOVEDFN (FMC-GET.ITEM OBJECT MOVEDFN))
          (SETQ Menu (FMC-GET.ITEM OBJECT MENU))
          (SETQ Links (FMC-GET.ITEM OBJECT LINKS))
          (SETQ Initstate (FMC-GET.ITEM OBJECT INITSTATE))
          (SETQ Background (FMC-GET.ITEM OBJECT BACKGROUND))
          (SETQ InfiniteWidth (FMC-GET.ITEM OBJECT INFINITEWIDTH))
```

```
          (SETQ ObjectList (FMC-GET.ITEM OBJECT OBJECTLIST))
          (if (FMC-GROUP? OBJECT)
              then ;; it's a group : compute the header
                   [SETQ TmpLst `(PROPS ID ,Id LEFT ,(GET.REGION.LEFT Region)
                                        BOTTOM
                                        ,(GET.REGION.BOTTOM Region)
                                        BOX
                                        ,Box BOXSHADE ,Boxshade BOXSPACE ,(IDIFFERENCE (FMC-GET.ITEM OBJECT
                                                                                              YBMOFFSET)
                                                                                       Box]
                   [if (AND Background (NOT (EQP Background 0)))
                       then (SETQ TmpLst (APPEND TmpLst `(BACKGROUND ,Background]
                   (SETQ TmpLst (CONS 'GROUP (LIST TmpLst)))
                   ;; and recursively build it's description
                   (FMC-SORT.ITEM.LIST ObjectList)
                   (NCONC1 TmpLst (FMC-COMPUTE.OBJECT (CAR ObjectList)
                                         Collection
                                         (FMC-GET.ITEM OBJECT DESELECT)))
                   (for item in (CDR ObjectList) do (NCONC1 TmpLst (FMC-COMPUTE.OBJECT item Collection)))
            else ;; it's a simple item : do the exhausting task of building a property list-format description
                 (SETQ TmpLst (LIST 'LABEL Label 'TYPE Type 'LEFT (GET.REGION.LEFT Region)
                                    'BOTTOM
                                    (GET.REGION.BOTTOM Region)))
                 (if Id
                     then (LISTPUT TmpLst 'ID Id))
                 [if (EQ Type 'NWAY)
                     then (LISTPUT TmpLst 'COLLECTION COLLECTIONID)
                          (if DESELECT
                              then (LISTPUT TmpLst 'NWAYPROPS '(DESELECT T]
                 (if Box
                     then (LISTPUT TmpLst 'BOX Box)
                          (LISTPUT TmpLst 'BOXSHADE Boxshade)
                          (LISTPUT TmpLst 'BOXSPACE (IDIFFERENCE (FMC-GET.ITEM OBJECT YBMOFFSET)
                                                                 Box)))
                 (if Font
                     then (LISTPUT TmpLst 'FONT Font))
                 [if (AND [NOT (EQUAL CHANGESTATE '(FUNCTION NILL]
                          (LISTP CHANGESTATE))
                     then (LISTPUT TmpLst 'CHANGESTATE (if (EQUAL (CAR CHANGESTATE)
                                                                  'LAMBDA)
                                                           then CHANGESTATE
                                                          else (CADR CHANGESTATE]
                 [for prop in '(CHANGESTATE SELECTEDFN DOWNFN HELDFN MOVEDFN) bind Def
                    do (SETQ Def (EVAL prop))
                       (if [NOT (EQUAL Def '(FUNCTION NILL]
                           then (LISTPUT TmpLst prop (if (EQUAL (CAR Def)
                                                                'LAMBDA)
                                                         then Def
                                                        else (CADR Def]
                 [if (CAR Menu)
                     then (LISTPUT TmpLst 'MENUITEMS (CAR Menu))
                          (if (CADR Menu)
                              then (LISTPUT TmpLst 'MENUFONT (CADR Menu)))
                          (if (CADDR Menu)
                              then (LISTPUT TmpLst 'MENUTITLE (CADDR Menu]
                 (if (CAR Links)
                     then (LISTPUT TmpLst 'LINKS Links))
                 (if Initstate
                     then (LISTPUT TmpLst 'INITSTATE Initstate))
                 (if (OR [AND [NOT (FMEMB Type '(EDIT NUMBER]
                              (NOT Box)
                              (NOT (EQUAL (GET.REGION.WIDTH Region)
                                          (BITMAPWIDTH (FMC-GET.ITEM OBJECT BITMAP]
                         (AND (FMEMB Type '(EDIT NUMBER))
                              (NOT InfiniteWidth)))
                     then (LISTPUT TmpLst 'MAXWIDTH (GET.REGION.WIDTH Region)))
                 (if (AND Background (NOT (EQP Background 0)))
                     then (LISTPUT TmpLst 'BACKGROUND Background)))
          (RETURN TmpLst])

)


          (* * Miscellaneous)

(DEFINEQ

(FMC-CREATE.ITEM.FROM.LIST
  [LAMBDA (L)                                          ; Edited 12-Aug-88 11:21 by A.BLAVIER
    ;; Given a list as those saved in a FMC file, create a new item.
    (create FMC-ITEM
            TYPE _ (CAR (NTH L 1))
            LABEL _ (CAR (NTH L 2))
```

```
          FONT _ (CAR (NTH L 3))
          ID _ (CAR (NTH L 4))
          COLLECTION _ (CAR (NTH L 5))
          DESELECT _ (CAR (NTH L 6))
          MESSAGE _ (CAR (NTH L 7))
          INITSTATE _ (CAR (NTH L 8))
          BOX _ (CAR (NTH L 9))
          BOXSHADE _ (CAR (NTH L 10))
          BACKGROUND _ (CAR (NTH L 11))
          LINKS _ (CAR (NTH L 12))
          OBJECTLIST _ (CAR (NTH L 13))
          MENU _ (CAR (NTH L 14))
          CHANGESTATE _ (CAR (NTH L 15))
          SELECTEDFN _ (CAR (NTH L 16))
          DOWNFN _ (CAR (NTH L 17))
          HELDFN _ (CAR (NTH L 18))
          MOVEDFN _ (CAR (NTH L 19))
          REGION _ (CAR (NTH L 20))
          BITMAP _ (CAR (NTH L 21))
          XBMOFFSET _ (CAR (NTH L 22))
          YBMOFFSET _ (CAR (NTH L 23))
          INFINITEWIDTH _ (CAR (NTH L 24))
          USERDATA _ (CAR (NTH L 25])
```

(FMC-DRAW.BOX

```
[LAMBDA (ITEM BOX BOXSHADE WINDOW)                    ; Edited 17-Aug-88 17:38 by A.BLAVIER
```

;; Draw a BOX wide box around the item with BOXSHADE.

```
(LET ((Region (FMC-GET.ITEM ITEM REGION))
      RLeft RBottom RWidth RHeight RRight RTop)
     (SETQ RLeft (GET.REGION.LEFT Region))
     (SETQ RBottom (GET.REGION.BOTTOM Region))
     (SETQ RWidth (GET.REGION.WIDTH Region))
     (SETQ RHeight (GET.REGION.HEIGHT Region))
     (SETQ RRight (fetch (REGION PRIGHT) of Region))
     (SETQ RTop (fetch (REGION PTOP) of Region))
     (if BOX
         then (DSPFILL (CREATEREGION RLeft RBottom BOX RHeight)
                       BOXSHADE
                       'REPLACE WINDOW)
              (DSPFILL (CREATEREGION RLeft (IDIFFERENCE RTop BOX)
                                     RWidth BOX)
                       BOXSHADE
                       'REPLACE WINDOW)
              (DSPFILL (CREATEREGION (IDIFFERENCE RRight BOX)
                                     RBottom BOX RHeight)
                       BOXSHADE
                       'REPLACE WINDOW)
              (DSPFILL (CREATEREGION RLeft RBottom RWidth BOX)
                       BOXSHADE
                       'REPLACE WINDOW])
```

(FMC-CHOOSE.WINDOW.BG

```
[LAMBDA (WINDOW)                                      ; Edited 17-Aug-88 17:39 by A.BLAVIER
```

;; Choose a background shade for the FMC window.

```
(LET ((ShadeMenu (WINDOWPROP (WINDOWPROP WINDOW 'FMC.IP.WINDOW)
                             'SHADE.MENU))
      Shade)
     (SETQ Shade (MENU ShadeMenu))
     (if Shade
         then (WINDOWPROP WINDOW 'FMC.BACKGROUND (CDR Shade))
              (FMC-REDRAW WINDOW])
```

(FMC-DISPLAY.GRID

```
[LAMBDA (WINDOW)                                      ; Edited 17-Aug-88 17:40 by A.BLAVIER
  (LET [(Size (WINDOWPROP WINDOW 'GRIDSIZE]
       [if (AND Size (IGREATERP Size 2))
           then (for y from 0 to (WINDOWPROP WINDOW 'HEIGHT) by Size do (DRAWLINE 0 y (WINDOWPROP WINDOW
                                                                                        'WIDTH)
                                                                                y 1 'REPLACE WINDOW NIL
                                                                                (LIST 1 (SUB1 Size]
       (WINDOWPROP WINDOW 'DISPLAYGRID T])
```

(FMC-SET.GRIDSIZE

```
[LAMBDA (SIZE WINDOW)                                 ; Edited 22-Jun-88 13:47 by A.BLAVIER
  (if (EQ SIZE 'NOGRID)
      then (WINDOWPROP WINDOW 'GRIDSIZE NIL)
           (FMC-PROMPTPRINT "No grid" WINDOW)
           (if (WINDOWPROP WINDOW 'DISPLAYGRID)
               then (FMC-REDRAW WINDOW))
           (WINDOWPROP WINDOW 'DISPLAYGRID NIL)
    else (FMC-PROMPTPRINT (CONCAT "Grid size is " SIZE)
```

```
            WINDOW)
        (if (NOT (EQP (WINDOWPROP WINDOW 'GRIDSIZE)
                      SIZE))
            then (WINDOWPROP WINDOW 'GRIDSIZE SIZE)
                 (if (WINDOWPROP WINDOW 'DISPLAYGRID)
                     then (FMC-REDRAW WINDOW])
```

(FMC-FONT->LIST

```
[LAMBDA (FONT)                                          ; Edited 17-Aug-88 17:42 by A.BLAVIER
```

;; Compute a Font description list based on a Font Descriptor.

```
(LET [(Face (FONTPROP FONT 'FACE]
     (LIST (FONTPROP FONT 'FAMILY)
           (FONTPROP FONT 'SIZE)
           (COND
              ((EQUAL Face '(MEDIUM REGULAR REGULAR))
               'STANDARD)
              ((EQUAL Face '(MEDIUM ITALIC REGULAR))
               'ITALIC)
              ((EQUAL Face '(BOLD REGULAR REGULAR))
               'BOLD)
              ((EQUAL Face '(BOLD ITALIC REGULAR))
               'BOLDITALIC])
```

(FMC-LIST->FONT

```
[LAMBDA (FONT.LIST)                                     ; Edited 17-Aug-88 17:42 by A.BLAVIER
```

;; Compute a Font Descriptor based on a Font description list.

```
(FONTCREATE (CAR FONT.LIST)
       (CADR FONT.LIST)
       (CADDR FONT.LIST)
       0
       'DISPLAY])
```

(FMC-SORT.ITEM.LIST

```
[LAMBDA (LIST)                                          ; Edited 17-Aug-88 17:43 by A.BLAVIER
```

;; Sort items by order of appearance in the window from top to bottom and from left to right.

```
(SORT LIST (FUNCTION (LAMBDA (ITEMA ITEMB)
                       (LET [(LEFTA (GET.REGION.LEFT (FMC-GET.ITEM ITEMA REGION)))
                             (LEFTB (GET.REGION.LEFT (FMC-GET.ITEM ITEMB REGION)))
                             (TOPA (fetch (REGION TOP) of (FMC-GET.ITEM ITEMA REGION)))
                             (TOPB (fetch (REGION TOP) of (FMC-GET.ITEM ITEMB REGION)))
                             (BOTTOMA (GET.REGION.BOTTOM (FMC-GET.ITEM ITEMA REGION)))
                             (BOTTOMB (GET.REGION.BOTTOM (FMC-GET.ITEM ITEMB REGION]
                            (OR (> BOTTOMA TOPB)
                                (AND (< LEFTA LEFTB)
                                     (< BOTTOMB TOPA])
```

(FMC-IMPORT

```
[LAMBDA (WINDOW)                                        ; Edited 12-Aug-88 10:42 by A.BLAVIER
```

;; Import items from a Free Menu.

```
(LET
 (FM.WINDOW FM.ITEMS)
 (FMC-PROMPTPRINT "Click on the Free Menu you want to import from" WINDOW)
 (while (MOUSESTATE (NOT LEFT)))
 (SETQ FM.WINDOW (WHICHW))
 (SETQ FM.ITEMS (WINDOWPROP FM.WINDOW 'FM.ITEMS))
 (if (NOT FM.ITEMS)
     then (FMC-PROMPTPRINT "This is not a Free Menu" WINDOW)
   else (INVERTW FM.WINDOW)
        (FMC-PROMPTPRINT "Importing items ..." WINDOW)
        (FMC-DESELECT WINDOW)
        (for ITEM in FM.ITEMS bind NEW.ITEM label message links userData menuProps changeState selectedFn
                                   downFn heldFn movedFn
           do (SETQ label (fetch FM.LABEL of ITEM))
              (SETQ message (fetch FM.MESSAGE of ITEM))
              (SETQ links (fetch FM.LINKS of ITEM))
              (SETQ userData (fetch FM.USERDATA of ITEM))
              (SETQ changeState (LISTGET userData 'CHANGESTATE))
              (SETQ selectedFn (fetch FM.SELECTEDFN of ITEM))
              (SETQ downFn (fetch FM.DOWNFN of ITEM))
              (SETQ heldFn (fetch FM.HELDFN of ITEM))
              (SETQ movedFn (fetch FM.MOVEDFN of ITEM))
              [SETQ NEW.ITEM (COPYALL
                               (FMC-CREATE.ITEM.FROM.LIST
                                (LIST (fetch FM.TYPE of ITEM)
                                      (if (OR (STREQUAL label "")
                                              (NULL label))
                                          then "*NOLABEL*"
                                        else label)
                                      (FMC-FONT->LIST (fetch FM.FONT of ITEM))
```

```
                                        (fetch FM.ID of ITEM)
                                        NIL NIL (if (OR (STREQUAL message "Will select this item when you release
                                                               the button.")
                                                        (STREQUAL message "Will let you select a value from a pop
                                                               up menu.")
                                                        (STREQUAL message "Will toggle this item when you release
                                                               the button."))
                                                    then ""
                                                  else message)
                                        (fetch FM.INITSTATE of ITEM)
                                        NIL 0 0 (if (CAR links)
                                                    then (LIST (CAR links)
                                                               (fetch FM.ID of (CADR links)))
                                                  else '(NIL))
                                        NIL
                                        (if (LISTGET userData 'MENUITEMS)
                                            then [SETQ menuProps (LIST (LISTGET userData 'MENUITEMS]
                                                 [if (LISTGET userData 'MENUFONT)
                                                     then (NCONC1 menuProps (LISTGET userData 'MENUFONT]
                                                 [if (LISTGET userData 'MENUTITLE)
                                                     then (if (NOT (CDR menuProps))
                                                              then (NCONC1 menuProps NIL))
                                                          (NCONC1 menuProps (LISTGET userData 'MENUTITLE]
                                                 menuProps
                                          else '(NIL))
                                        (if changeState
                                            then (if (ATOM changeState)
                                                     then (LIST 'FUNCTION changeState)
                                                   else changeState)
                                          else '(FUNCTION NILL))
                                        (if (ATOM selectedFn)
                                            then (LIST 'FUNCTION selectedFn)
                                          else selectedFn)
                                        (if (ATOM downFn)
                                            then (LIST 'FUNCTION downFn)
                                          else downFn)
                                        (if (ATOM heldFn)
                                            then (LIST 'FUNCTION heldFn)
                                          else heldFn)
                                        (if (ATOM movedFn)
                                            then (LIST 'FUNCTION movedFn)
                                          else movedFn)
                                        (fetch FM.REGION of ITEM)
                                        (FMC-MAKEBITMAP (if (OR (STREQUAL label "")
                                                                (NULL label))
                                                            then "*NOLABEL*"
                                                          else label)
                                                        (fetch FM.FONT of ITEM))
                                        0 0 (LISTGET userData 'INFINITEWIDTH)
                                        NIL]
                                [if (OR (STREQUAL label "")
                                        (NULL label))
                                    then (PUT.REGION.WIDTH (FMC-GET.ITEM NEW.ITEM REGION)
                                                           (BITMAPWIDTH (FMC-GET.ITEM NEW.ITEM BITMAP)))
                                         (PUT.REGION.HEIGHT (FMC-GET.ITEM NEW.ITEM REGION)
                                                            (BITMAPHEIGHT (FMC-GET.ITEM NEW.ITEM BITMAP]
                                [if (LISTGET userData 'BOX)
                                    then (FMC-PUT.ITEM NEW.ITEM BOX (LISTGET userData 'BOX))
                                         (FMC-PUT.ITEM NEW.ITEM BOXSHADE (LISTGET userData 'BOXSHADE))
                                         (FMC-PUT.ITEM NEW.ITEM XBMOFFSET (LISTGET userData 'BOXOFFSET))
                                         (FMC-PUT.ITEM NEW.ITEM YBMOFFSET (LISTGET userData 'BOXOFFSET]
                                (WINDOWADDPROP WINDOW 'ITEMLIST NEW.ITEM)
                                (WINDOWADDPROP WINDOW 'SELECTION.LIST NEW.ITEM)
                                (FMC-REDRAW.ITEM NEW.ITEM WINDOW))
                          (INVERTW FM.WINDOW)
                          (FMC-PROMPTPRINT "Importing items ... DONE" WINDOW)
                          (FMC-SELECT.LIST WINDOW)
                          (FMC-MARK.AS.CHANGED WINDOW])
```

(FMC-PROMPTPRINT

```
  [LAMBDA (STR WINDOW)                                    ; Edited 17-Aug-88 17:44 by A.BLAVIER
    ;; Print a message in the FMC prompt window.
    (LET [(PromptWindow (WINDOWPROP WINDOW 'FMC.PROMPTWINDOW]
         (DSPRESET PromptWindow)
         (PRINTOUT PromptWindow T STR)
         (FLASHWINDOW PromptWindow 1])

)


          (* * Icon stuff)

(RPAQQ FMC-ICON
```

)

(RPAQQ **FMC-ICON.MASK**

)

```
(RPAQ? FMC-ICON.TEMPLATE (create TITLEDICON ICON _ FMC-ICON MASK _ FMC-ICON.MASK TITLEREG _
                                 (CREATEREGION 2 2 70 28)))


          (* *)

[OR (SASSOC 'FMCreator BackgroundMenuCommands)
    (NCONC1 BackgroundMenuCommands '(FMCreator '(FMC-CREATE)
                                          "Opens a Free Menu Creator window for use"]

(SETQ BackgroundMenu NIL)

(RPAQ MOVINGCURSOR (CURSORCREATE '
                                 'NIL 7 7))

(PUTPROPS FM-CREATOR COPYRIGHT ("Rank Xerox France. Author Andre BLAVIER" 1988))
```

FUNCTION INDEX

MACRO INDEX

VARIABLE INDEX

RECORD INDEX

PROPERTY INDEX